Complexity

Machine Learning Applications in Complex Economics and Financial Networks

Lead Guest Editor: Benjamin Miranda Tabak
Guest Editors: Thiago Christiano Silva, Liang Zhao, and Ahmet Sensoy

Machine Learning Applications in Complex Economics and Financial Networks

Complexity

Machine Learning Applications in Complex Economics and Financial Networks

Lead Guest Editor: Benjamin Miranda Tabak
Guest Editors: Thiago Christiano Silva, Liang Zhao, and Ahmet Sensoy

Contents

Hindawi
Complexity
Volume 2021, Article ID 6653410, 19 pages
https://doi.org/10.1155/2021/6653410

Research Article

Understanding Service Providers' Competency in Knowledge-Intensive Crowdsourcing Platforms: An LDA Approach

Biyu Yang[1], **Xu Wang**[1,2] **and Zhuofei Ding**[3]

[1]*College of Mechanical Engineering, Chongqing University, Chongqing 400030, China*
[2]*Chongqing Laboratory of Logistics, Chongqing University, Chongqing 400030, China*
[3]*School of Informatics, University of Edinburgh, Edinburgh EH8 9JU, UK*

Correspondence should be addressed to Xu Wang; wx921@163.com

Received 12 November 2020; Revised 5 February 2021; Accepted 8 July 2021; Published 17 July 2021

Academic Editor: Liang Zhao

Knowledge-intensive crowdsourcing (KIC) is becoming one of the most promising domains of crowdsourcing by leveraging human intelligence and building a large labor-intensive service network. In this network, the service providers (SPs) constitute the backbone of the KIC platform and play an important role in connecting the platform and service requesters. The SPs are a group of distributed crowds with a complex composition and high level of uncertainty, resulting in great challenges in service quality and platform management. Understanding the SPs' competency is an effective way for the platform to manage them. Therefore, we attempt to connect the competency analysis to the environment of KIC to investigate and identify the criteria of SPs' competency (i.e., the competency factors and dimensions required for being competent for the SPs' business). To this end, we leverage the Latent Dirichlet Allocation (LDA) model to explore and extract hidden competency dimensions from online interview records. We then introduce the competency theory to identify and label the competency factors and dimensions and construct the three-level KSAT competency model, which presents a comprehensive vision of the SPs' performance standards in the context of KIC. Given the competency criteria in the KSAT competency model, we use the Best-Worst Method (BWM) to determine their weights, which reflect their importance when evaluating the SPs' competency from the platforms' perspective. The results show that skill and knowledge are the two most important competency factors, and customer relationship management and communication ability are the two most valuable competency dimensions when evaluating the SPs' competency. Furthermore, the KSAT competency model can be applied to analyze the competency of individuals or organizations in many other industries as well.

1. Introduction

As facilitated by highly developed information technologies, knowledge-intensive crowdsourcing (KIC) taps into the creative and innovation fields (e.g., designers), changing the traditional way of how business is conducted [1]. For example, in the design industry, penetration of technology, especially Internet and smartphone apps, has changed how design is practiced, produced, and traded. In terms of the critical role, it plays in today's knowledge-based economy, KIC is considered to be one of the most promising domains of crowdsourcing in the future [2, 3].

The KIC platform operates as a two- or multisided market, meaning that each side of the market derives externalities from the participation of the respective other side, which is called the network effects [4, 5]. Due to this, the KIC platform experiences tremendous growth in the user base and becomes a large complex network. In this network, a large number of open and crowdsourcing service providers (SPs) constitute the backbone of the KIC platform. Also, based on the resource-based view, the network effects allow the SPs to be converted into critical resources, creating a continuous competitive edge for the platform [6], and bring certain benefits. As a vital connection between the platform and service requesters, they assist the KIC platform in running a wide range of business and services catering to numerous end customers of different types and attain business success in their areas of professionals. In this process, they can meet the service requesters' demand. Especially for those high-quality and competent SPs, they

can perform knowledge-based tasks well and generate satisfying outcomes. For the SPs, delivering services with committed quality and performance to achieve sustainable business growth is a concrete demonstration of competency, which enhances service requesters' satisfaction and trust toward the platform [7]. In turn, service requesters who previously had a good shopping or interactive experience are more likely to engage with the platform on an ongoing basis. By delivering anticipated services to their customers on the platform, the SPs enable the platform to attract more customers, generate more traffic, and capture more market share, so that the platform can remain competitive in the market [8, 9].

Despite the certain benefits the SPs have brought to the platform and service requesters, some potential risks in terms of service quality and platform health may arise from the crowdsourcing activities. For instance, some SPs may prove precarious, due to the fact that the supplier database consists of complex and previously unknown crowds [10]. There may also be potential hazards such as cheating, manipulating task outputs, or extracting sensitive information from crowdsourcing systems [10, 11]. Even so, the KIC platform still weakens the input controls and simplifies the registration procedures to attract SPs. Additionally, the relationship between the platform and the SPs is not the classic principal-agent relationship (i.e., the platform does not hire these SPs), which offers SPs the flexibility of their work time and schedules [5]. Accordingly, concerns about the uncertainties of the operation process, including SPs' availability, service awareness, financial and intellectual property, and privacy risks, are growing, which may, in turn, jeopardize the platform's reputation and affect its healthy operation [10, 11]. Under this circumstance, it is imperative for the platform to take some necessary actions to identify those who are not competent for the knowledge-intensive tasks.

Regarding the above benefits and risks, it is the responsibility of the platform, as an indispensable manager and regulator of the KIC activities, to manage the SPs on the platform. To this end, conducting competency analysis and understanding the SPs' competency is an effective way for the platform to fulfill the responsibilities and to have a clear vision of the SPs' performance standards and expectations [12]. On the one hand, the competency analysis offers a solution for the platform to construct a competency evaluation framework for the SPs [12, 13] and differentiate high performance from middle and low performance [14]. On the other hand, the competency analysis helps the SPs to achieve certain goals under the standardized framework developed by the platform. Therefore, we attempt to associate the competency analysis with the context of KIC.

Practically, we have noted that some KIC platforms have taken measures to facilitate the SPs' performance. According to our survey, some platforms in China will release a list of leaders online in terms of different criteria, such as bidding price and daily or total sales. Also, the reputation system of the SPs about their service quality, speed, and attitude is visible on their homepage. However, it is biased to consider only reputation, total sales volume, or platform score. Some other factors that may determine the SPs' competency are not taken into consideration, such as entrepreneurial experience [15], communication ability [16], and innovativeness [17, 18]. It shows that not enough attention has been paid by practitioners to those aspects, and, as yet, a comprehensive understanding of the SPs' competency is still lacking.

Therefore, both the theoretical backgrounds and the practical applications motivate us to investigate the SPs' competency (i.e., the detailed components of competency that are required for being competent for the SPs' business in KIC). To this end, this paper aims to address the following question:

What is about the SPs' competency in the context of KIC and to which competency criteria should the practitioners pay more attention when assessing the SPs' competency and performance?

To understand the SPs' competency, we aim to explore and identify competency criteria (i.e., the competency factors and dimensions). We analyze online interview records posted by several KIC platforms in China, which includes successful SPs' opinions about the qualifications to obtain sustainable business growth and conduct their career well. We first crawl these online interview records and then extract and identify the competency dimensions leveraging the Latent Dirichlet Allocation (LDA) model. We then construct the KSAT competency model for the KIC platform to evaluate and manage the SPs. Further, questionnaires are developed to collect experts' opinions and the Best-Worst Method (BWM) is applied to determine the weights and priorities of the competency criteria in the proposed KSAT competency model. Therefore, the main contributions of this research are summarized as follows:

(1) This study expands the outreach of the competency theory by introducing it to the KIC environment. To the best of our knowledge, it is the first time that the competency theory is applied in the field of KIC.

(2) This study presents a novel research framework enabling the KIC platform to transform the online textual information into useful knowledge about SPs' competency. The incorporation of text mining and decision-making methods offers insights for researchers and practitioners to understand the hidden content behind the large collection of unstructured textual information on the KIC platform.

(3) The proposed KSAT competency model provides a rich set of indicators and variables that allow both researchers and practitioners to flexibly design, build, and formulate specific evaluation framework in terms of different requirements and goals.

In the following sections, the related works are presented in Section 2. The methods used in this research are detailed in Section 3, followed by the data analysis and results in Section 4. We then discuss the KSAT competency model and

criteria importance in Section 5. Finally, the conclusion is drawn in Section 6.

2. Related Works

2.1. Knowledge-Intensive Crowdsourcing. The collaboration landscape has changed remarkably over recent decades where users can shape the Web and availability of information via highly developed information technology. Traditionally, collaborations were mostly concentrated within organizations' internal function departments [19] and also limited to messaging tools such as e-mail. However, it is nowadays possible to leverage the knowledge and intelligence of an immense number of people across geographic and organizational boundaries through crowdsourcing [20, 21]. Since its appearance, crowdsourcing has attained much success and has become a widely commercial phenomenon [22]. Meanwhile, the KIC is considered to be one of the most promising domains of crowdsourcing in the future, and in terms of the critical role, it plays in today's knowledge-based economy [3]. It refers to crowdsource human intelligence- and expertise-related tasks, such as question answering, image annotation, product development, website design, logo design, and software development [23], which cannot be performed by computers, to a crowd of people in an open call [24, 25]. In recent two decades, many KIC companies like Threadless, InnoCentive, Amazon Mechanical Turk, and some Chinese crowdsourcing platforms, such as ZBJ.COM, epwk.com, and 680.com, are established to support posting and performing various KIC tasks [23].

The KIC intermediaries are typical two- or multisided markets, where the network effects will attract an increasing number of participants from both supply and demand sides, to join the platform for valuable advantages [26]. Oosterman et al. [27] suggested that the KIC has the advantage of low cost for service requesters; therefore, it allows cost-saving and efficient use of resources. Additionally, by inviting a crowd of customers to new product development through a crowdsourcing practice, taking Dell IdeaStorm [28] for example, innovative product or service ideas can be generated and then applied to the production process, thus making the products more attractive to markets and adding value to companies' business [19]. Further, crowdsourcing eliminates geographical limitations and offers SPs the chance to develop their careers and pursue valuable and interesting jobs [29]. Consequently, an increasing number of service requests turn to the KIC platform for business ideas from the SPs and many SPs adopt online KIC crowdsourcing to gain knowledge and monetary benefits from transactions with service requests [23, 30, 31].

Based on the resource-based view, the network effects can turn SPs into critical resources that bring sustained competitive advantages to the platform [6]. As an intermediary, the KIC platform has to better manage SPs for customer satisfaction, competitive advantages, and sustainable growth. The most direct measure is to control the output quality and enhance customer satisfaction. Existing research proposed different quality control approaches to estimate the SPs' quality for a specific task. Vakharia and Lease [32] and Li et al. [33] reviewed different task-oriented quality control methods implemented by crowdsourcing platforms practically and researchers academically. The qualification test refers to a set of golden questions with known true answers that the SPs have to answer [34]. They are allowed to perform the real tasks until they pass the test and achieve a threshold score. For example, Clickworker, Crowdsource, and MTurk provide prequalification systems to assess the skill level (e.g., language level) of potential SPs. The gold-injected method mixes golden tasks with real tasks and workers do not know which tasks are gold ones during the task completion process [35]. For example, CrowdComputing Systems and CrowdFlower enable service requesters to inject gold standard data, i.e., a collection of tasks with known correct answers, into their tasks to measure the SPs' performance automatically. Iterative computation methods iteratively compute and update the SPs' quality by leveraging all other SPs' answers for all tasks. The underlying concept is that the SPs who frequently submit reliable answers will be assigned with high quality scores and answers provided by the SPs with high scores will be selected as true answers [36]. Furthermore, many other iterative computation approaches have been applied and developed to compute and measure the SPs' quality, such as EM-based (expectation estimation) methods, and graph-based methods [37]. However, these SPs' quality assessment methods are task-specific and have limited applications [38]. In addition, these approaches focused on the determination of a single label [39], e.g., the quality score of the SPs, or whether the SPs are allowed to perform a task.

There also exists another steam of research using multicriteria decision-making (MCDM) methods considering varied attributes to select the appropriate SPs and ensure the output quality for a task in the KIC context. Gong [3] proposed an integrated AHP (Analytic Hierarchy Process) and TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) approach and took attributes such as credit, skill test score, and the number of completed tasks into consideration, to evaluate and select SPs in a network of crowdsourcing marketplaces. Zhang and Su [40] considered several criteria of the SPs, namely, interests, competence, reputation, and availability to participate, and offered a combined fuzzy DEMATEL (Decision Making Trial and Evaluation Laboratory) and TOPSIS approach to select the candidate SPs for a KIC task. The relationships among these criteria and their indicators were further explored in [41]. However, the MCDM methods mainly aggregate the SPs' attributes from the perspective of the platform operators or service requesters and only consider a single or a small part of the criteria on behalf of their research goals [40].

Existing studies indicated that the management of SPs is of great significance for the KIC platform. Much attention has been paid to task-specific test design, quality computation methods improvement, and the combination of MCDM approaches to determine the weights of criteria. However, the competency analysis about the SPs and investigation of the competency components are not included, even though it is vital for the SPs to be competent in their crowdsourcing business. As a result, we attempt to connect

the competency analysis with KIC and thus the competency analysis is introduced.

2.2. Competency Analysis. The concept of the competency analysis was coined by McClelland and the McBer and Company in the 1970s, and it is defined as components of performance associated with "clusters of life outcomes" [42]. This definition views competency very broadly as any psychological or behavioral attributes associated with long-term success [43]. Later on, the competency movement started in the 1970s. Gibbons [44] argued that the movement was mainly caused by the disconnection between education and the labor market. Professional organizations had to articulate performance standards and requirements and develop competency profiles with which candidates have to comply to be professional [8]. The concept of competency is multidimensional, and various conceptions emerged. Now competency is generally conceptualized as "knowledge, skills, abilities, or other characteristics (KSAOs) that differentiate high from low performance" [14]. To date, the competency analysis is widely applied in many facets of human resource management. Kurz and Bartram [45] and Bartram [46] introduced the Great Eight competency model as a generic competency model that can be applied across a variety of jobs and organizations. By including the research of McClelland, Mirabile [14] proposed the KSAOs competency model that consists of a set of attributes possessed by the workers, typically indicated as knowledge, skills, attitudes, and personal traits required for effective work performance [47]. Based on the Great Eight, Krumm et al. [48] developed the KSAOs model for virtual teamwork which contains 60 potentially relevant items and compared the differences of KSAOs requirements between virtual and traditional teams. According to the empirical study in Hertel et al. [49], the authors found that a set of KSAOs (e.g., persistence, willingness to learn, creativity, independence, interpersonal trust, and intercultural skills) were related to tele-cooperation performance and indicated that creativity and independence significantly contributed to the team performance. Cogliser et al. [50] indicated that computer self-efficacy was the main performance predictor of a virtual management organization. Maurer and Lippstreu [51] conducted a survey to rate a varied set of KSAOs in terms of improvability, importance, and "needed at entry" facets. Prahalad and Hamel [52] pointed out that focusing on a collection of core competencies, i.e., the company's collective knowledge about how to coordinate diverse production skills and technologies, will help the corporation gain competitive advantages. Boyatzis [53] offered a "total" system approach that determines which characteristics of the managers enable them to be effective in various management jobs. Wu and Lee [54] developed the competencies of the global managers by using eight different IQs and proposed the fuzzy DEMATEL method to segment the required competencies into meaningful portions. Li et al. [7] proposed a multicriteria competency analysis framework for the crowdsourcing delivery personnel.

In general, these studies present a fact that individual or organizational performance is influenced by various competency items. Across different domains of surveys, however, models are rather heterogeneous in terms of which specific competencies are significant to performance. Although these studies offer valuable contributions and insights that help us to understand the competency of different roles, the differences in their findings highlight that their model structures are by no means universal and strongly depend on the characteristics of the specific context.

3. Methodology

To investigate the competency dimensions and understand the SPs' competency in KIC, this paper takes the advantages of the SPs' experience sharing information and explores the SPs' competency based on their understanding and perception. We first apply the topic modelling techniques to the crawled raw data and then construct the three-level KSAT competency model based on the extracted and identified competency dimensions. Later, the BWM is leveraged to explore the importance and weights of competency criteria in the KSAT competency model. Our research framework is depicted in Figure 1. We detail our methods in this section.

3.1. Topic Modelling. The rising development and accessibility of large electronic archives, along with increased computational facilities, have led to an interest in the textual content analysis [55]. The topic modelling is an effective modelling method for extracting implied themes in large-scale text based on word cooccurrence for each document in the corpus [56–58]. Moreover, the topic modelling is a useful way to "let the text talk" due to the independence from the evaluator's personal opinions or experiences [59], and it has been studied and applied in various fields, such as recommendation systems [60], online health communities [61], and customer-generated content analysis [62].

The LDA is a well-known unsupervised machine learning technique for natural language processing [63] and is the simplest and most popular topic modelling algorithm [64, 65], which has the advantage of recognizing the hidden topics and mining deep semantics of huge amounts of textual documents through an effective way. The basic idea of the LDA is that each document exhibits a mixture of latent topics wherein each topic is characterized by a distribution over the words, i.e., per-document topic distributions and per-word topic distributions [66, 67]. The generative probabilistic model of the LDA is represented in Figure 2.

The LDA defines the following terms:

(1) A word is the basic unit of discrete data, defined to be an item from a vocabulary indexed by $\{1, \ldots, V\}$

(2) A document is a sequence of N words denoted by $d = \{w_1, w_2, \ldots, w_N\}$, where w_n is the nth word in the sequence

(3) A corpus is a collection of M documents denoted by $D = \{d_1, d_2, \ldots, d_M\}$

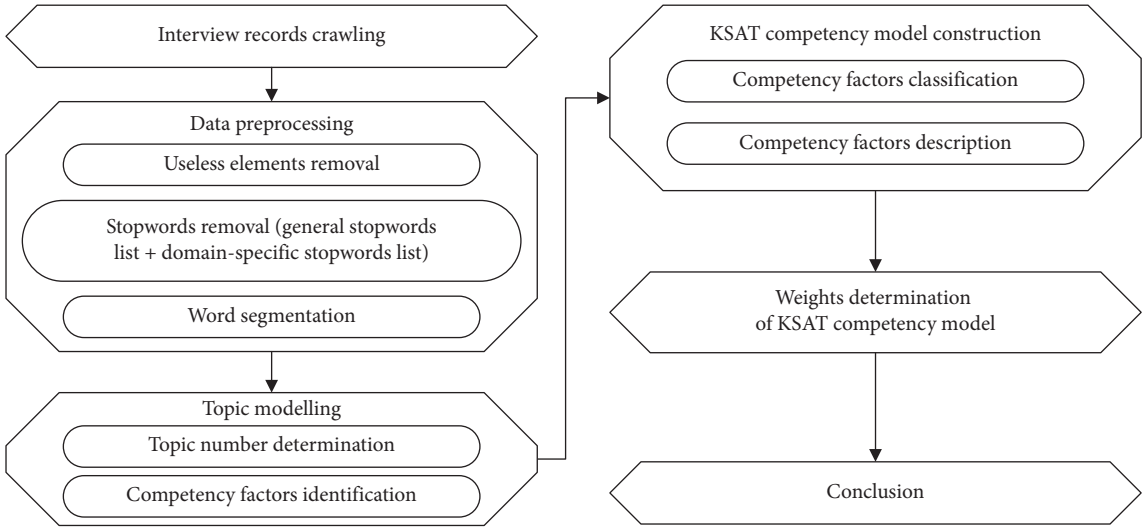


Figure 1: Research framework.

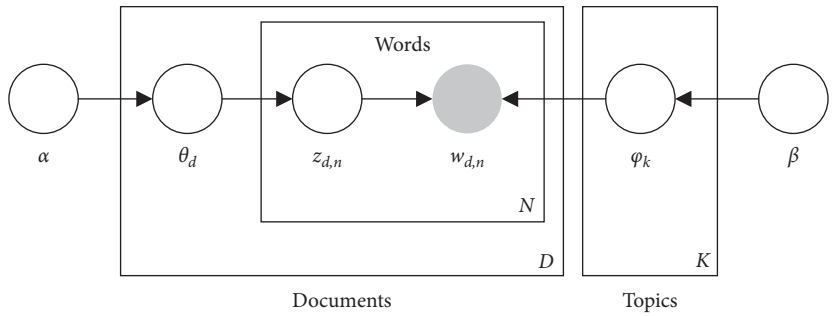


Figure 2: The generative probabilistic graph model of LDA.

As shown in Figure 2, the words within documents $w_{d,n}$ are observable variables, while the other variables, including the topics φ_k, $k = \{1, 2, \ldots, K\}$ (the distributions over words), the topic distribution per document θ_d, and the per-word topic distribution $z_{d,n}$, are not known. These latter items represent unobservable variables (white circles in Figure 2) that should be estimated from the analysis of observable variables (shaded circles in Figure 2). Parameters α and β are the hyperparameters for prior distributions of θ_d and φ_k, respectively. The plate notations at the bottom of each rectangle denote their usage to illustrate the replications; i.e., the K plate represents the number of topics, the N plate represents the total number of unique words within documents, and the M plate represents the number of documents. The arrows represent conditional dependencies among components in the following way: per-word topic distribution $z_{d,n}$ is dependent on the topic distribution per document θ_d, and the observed word in each document $w_{d,n}$ is dependent on $z_{d,n}$ and all the topics φ_k. The conditional dependencies enable the definition of the joint distribution of observed and unobserved variables. As a result, the LDA has remarkable advantages in employing Bayesian learning to infer latent variables by calculating their posterior distribution from the joint distribution. Learning the unobservable components allows us to capture the hidden semantic structure in the documents. More specifically, the main outputs yielded by the LDA are, namely, topics φ_k and their weights in each document θ_d, and per-word weight within each topic $z_{d,n}$. That is to say, the outputs of LDA consist of K topics, wherein each topic is denoted as a combination of words with different probability of occurrence. However, the combination of words cannot deliver the precise meaning of each topic; it would be better to label them. In other words, human judgments and intervention are a usual way to label the topics on the basis of the semantic similarities of included words [68]. For more details about the LDA, one can refer to [66, 67].

3.2. Best-Worst Method. The BWM is a recently developed MCDM method [69]. It uses a structured way to conduct the pairwise comparisons, which has several major advantages [7]. (i) The decision-makers are required to identify the best and worst criteria (or alternatives) prior to conducting a pairwise comparison, which enables them to have a clear understanding of the range of evaluation, consequently, resulting in more consistent and reliable pairwise comparisons [70]. (ii) By using two opposite references (best and worst) in a single optimization model, which is called consider-the-opposite-strategy, the BWM has been proven

to be effective in mitigating possible anchoring bias arising during the pairwise comparisons process [71, 72]. (iii) The BWM better balances the data and time efficiency and, at the same time, enables the decision-makers to check the consistency of the provided pairwise comparisons [71]. On the one hand, compared to other pairwise comparison-based methods using a single vector such as the Swing and SMART family, the BWM overcomes the main weakness that it is unavailable to check pairwise comparison consistency, while maintaining the high data (and time) efficiency of such single vector input-only methods [71]. On the other hand, the number of pairwise comparisons needed to be conducted in the BWM is less than that of full-matrix-based methods, such as AHP, which effectively enhances time and data efficiency. Although the number of pairwise comparisons under the full-matrix-based method is sufficient to check the consistency, decision-makers have to answer too many questions, which can result in confusion and inconsistency [71].

The method has been widely applied in many real-world problems, such as the supply chain, manufacturing, logistics, airline industry, supplier selection, and service quality evaluation. For a review of the applications, one could refer to Mi et al. [73].

In this study, we use the BWM due to its advantages and wide applications. Specifically, the steps to determine the weights of criteria using the BWM are as follows [69, 70]:

(1) Determine a set of decision criteria $\{c_1, c_2, \ldots, c_n\}$ by experts or decision-makers.

(2) Identify the best (B) and the worst (W) criteria by experts or decision-makers.

(3) Determine the preference of the best over all the other criteria by experts or decision-makers using a number between 1 and 9 (where 1 is "equally important" and 9 is "extremely more important"). The result of best-to-others comparisons is vector $V_B = (a_{B1}, a_{B2}, \ldots, a_{Bj}, \ldots, a_{Bn})$, where a_{Bj} shows the preference of criterion B over criterion j.

(4) Determine the preference of all the criteria over the worst by experts or decision-makers using the same scale (1 to 9). The result of others-to-worst comparisons is vector $V_W = (a_{1W}, a_{2W}, \ldots, a_{jW}, \ldots, a_{nW})^T$, where a_{jW} indicates the preference of criterion j over criterion W.

(5) Compute the optimal weights $(w_1^*, w_2^*, \ldots, w_n^*)$.

The optimal weights are computed by minimizing the maximum absolute differences of $\{|w_B - a_{Bj}w_j|, |w_j - a_{jW}w_W|\}$ for all j, which can be expressed by the following optimization problem:

$$\min \max_j \left\{ \left| w_B - a_{Bj} w_j \right|, \left| w_j - a_{jW} w_W \right| \right\}, \tag{1}$$

subject to

$$\sum_{j=1}^{n} w_j = 1, \quad w_j \geq 0, \text{ for all } j. \tag{2}$$

Equation (2) is converted into

$$\min \xi, \tag{3}$$

subject to

$$\begin{aligned} &\left| w_B - a_{Bj} w_j \right| \leq \xi, \quad \text{for all } j, \\ &\left| w_j - a_{jW} w_W \right| \leq \xi, \quad \text{for all } j, \\ &\sum_{j=1}^{n} w_j = 1, \\ &w_j \geq 0, \quad \text{for all } j. \end{aligned} \tag{4}$$

$W^* = (w_1^*, w_2^*, \ldots, w_n^*)$ is the result of equation (4), indicating the optimal weight of criteria. ξ^* is the result of the objective function in equation (4), indicating the consistency of the provided pairwise comparisons. If ξ^* is closer to zero, a higher level of consistency is in the pairwise comparisons by experts.

When the MCDM problem is a hierarchical criteria tree, then the results of equation (4) are called local weights. To determine the global weights of subcriteria in the last level of the tree, their local weights are multiplied by the weights of the category to which they belong. When we have a number of experts, we follow all the five steps for each expert individually. To aggregate the final results (global weights), we use the geometric mean.

4. Data Analysis and Results

In this section, we give details of the data collection and preprocessing, the data analysis, and the results of our model. Related procedures are conducted on an Intel Core i7 CPU, 16 GB RAM machine. The raw data are crawled and analyzed based on scikit-learn under Python version 3.6.5.

4.1. Data Collection and Preprocessing. According to the report [74], the transaction size of the KIC market in 2020 increased by 306.3 billion RMB compared to 2019 in China. As facilitated by the Chinese policy "mass entrepreneurship and innovation" [75], crowdsourcing platforms in China, such as ZBJ.COM, epwk.com, and 680.com that are specialized in supporting KIC activities, also experienced booming growth. Given that SPs participate in these platforms at different time and with various business capabilities, in order to facilitate their business growth and career development, these platforms provide them the opportunities for knowledge and experience sharing and learning. Particularly, these platforms will regularly organize interviews with successful SPs (i.e., top order winners, top income earners, or long-established SPs) to share experience regarding a series of questions, such as "how did you manage to get so many orders?"; "what did you do to get an order at such a high price?"; "what efforts did you make to be a long-established SP?"; and "What is your plan to further facilitate your business in the future?". Then, the interview records will be posted on their platforms in order that more SPs will access and learn from these successful experiences.

According to our survey with SPs of ZBJ.COM, they considered the experiences of the successful SPs to be highly informative, providing effective guidance on the issues they were facing at that time. We think that these online interview records are of great significance for the KIC platform to understand SPs competency and derive competency dimensions that can improve SPs management. To undertake this study, we apply the following steps to collect the raw data, i.e., the interview records, as the corpora for this research.

(1) Selecting the data sources: to ensure the quality and quantity of the raw data, we select the most three popular KIC platforms, namely, ZBJ.COM, epwk.com, and 680.com, as the data source.

(2) Writing the clawer programming: due to the large amount of content available on these platforms, accessing the data online through the programming is an effective and efficient way. Consequently, we code the programming in Python.

(3) Crawling the raw data: running the programming in Step 2, we crawl the online interview records on the platforms. Key information including the title, content, poster, post time, view times, comments, and responses is collected and finally the corpora contain 1760 records in total within a time frame from January 2014 to July 2019.

The raw data collected from the KIC platform are all unstructured and in Chinese. To effectively extract implied topics from these large-scale texts, we apply the general text preprocessing steps to clean the unstructured text for topic modelling as follows:

(1) Cutting each article into sentences and eliminating all numbers and alphabets, just leaving the Chinese characters: note that Chinese text is processed in UTF-8 encoding format.

(2) Defining the user dictionary and tokenizing each sentence into multiple space-separated words: in this step, we refer to a predefined user dictionary to know where to pause in a sentence. Unlike in English, there is no space between two Chinese characters or phrases. Given the different segmentation, the meaning of the sentence may be completely different. With a predefined dictionary, a complete Chinese sentence will be cut into several tokens.

(3) Removing the stopwords from each sentence: to better clean the data, except for using the popular Chinese stopwords list, we construct our own stopwords list with words that are domain-specific and highly appear in our corpora but useless for analysis (e.g., crowdsourcing, the platform, innovation design).

(4) Constructing the term-document matrix: after removing the stopwords, the term-document matrix (i.e., the distribution/frequency of terms (rows) within documents (columns)) used as the main input to the LDA is constructed.

4.2. Competency Dimensions Identification. Prior to building a topic model from the experience sharing articles, we need to exogenously give the number of topics K. To obtain the proper topic number, Aletras and Stevenson [76] and Ramage et al. [77] introduced cosine measures to capture the similarities of generated topics, where the lowest average cosine similarity denotes the best model and thus determines the appropriate topic number. In this sense, we determine the optimal number of topics from a discrete range rather than a continuous range. We calculate the average cosine similarity setting the number of topic K over a number set $\{4, 6, 8, 10, 12, 14, 16, 18, 20, 22\}$. Figure 3 presents a comparison over discrete topic numbers in terms of the average cosine similarity of topics. As shown in Figure 3, we find that the average cosine similarity score becomes the lowest (0.0375) when the number of topics is set to 18. Therefore, the appropriate number of topics is 18. We then set the topic number K to 18 and we obtain 18 clusters of words from the corpora (see the second column in Table 1). As discussed earlier, the topics are distributions over words; the top five keywords with the highest probability (most frequency) derived from the posterior distribution (i.e., $z_{d,n}$) are provided for each topic in Table 1. Given the corpora are in Chinese, we translate the words in English for readers to better understand the topics and attach the original Chinese words.

However, presenting as the combinations of words cannot characterize the underlying content of the topics. Unfortunately, automatic labeling of the topics is infeasible because the topic extraction by the LDA is an unsupervised learning process. Instead, it requires human judgment and intervention to check the coherence and meaningfulness of these topics and then label them through their judgment [66, 68]. As a result, the authors interviewed the managers and the SPs in ZBJ.COM and the researchers in related fields and then validated and labeled the extracted topics, by referring to the Spencer's competency dictionary [13]. As in Kurz and Bartram [45], we refer to the label of each topic as a competency dimension, as indicated in the last column of Table 1.

4.3. The KSAT Competency Model. Mirabile [14] proposed the KSAOs competency model that consists of a set of attributes possessed by workers, typically indicated as knowledge, skills, attitudes, and personal traits required for effective work performance. In order to frame our findings in theory and also offer more significant insights the LDA results, the eighteen competency dimensions are classified into four competency factors, namely, knowledge, skill, ability, and trait, according to the aforementioned interviews with managers, the SPs and related researchers, on the basis of the meaning of topics and competency dimensions. As in Kurz and Bartram [45], we define the main criteria as competency factors and subcriteria as competency dimensions. Particularly, we summarize the two competency dimensions, demand understanding and reasonable suggestion, into one single dimension, task analysis, and

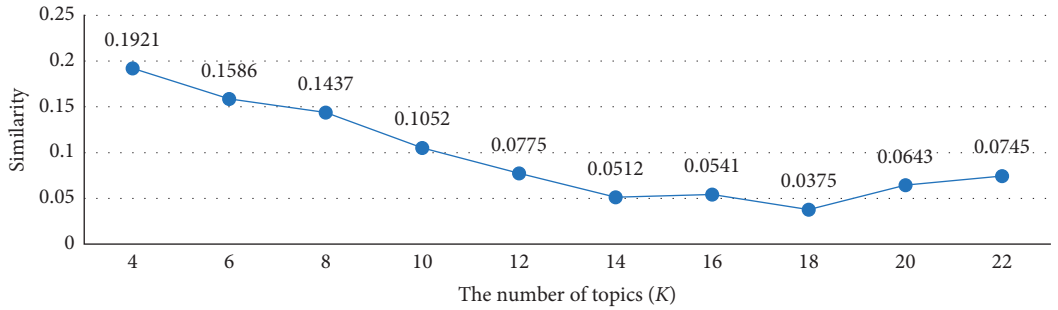


Figure 3: The average cosine similarity over different topic numbers.

Table 1: Competency factors identified by LDA.

Topic	Clusters of keywords	Competency dimensions
1	Pursuit (追求)*0.111 Online (线上)*0.090 Process (流程)*0.057 Offline (线下)*0.041 Keep improving (精益求精)*0.038	Online and offline coordination
2	Competition (竞争)*0.127 Advantages (优势)*0.058 Development (发展)*0.039 Mindset (心态)*0.037 Confront (面对)*0.031	Competitive spirit
3	Demand (需求)*0.078 Understanding (理解)*0.077 Master (把握)∗0.040 Sincerity (真诚)*0.036 Accurate (准确)*0.035	Demand understanding
4	Promotion (提升)*0.119 Improvement (完善)*0.057 Brand image (品牌形象)*0.055 Management (管理)*0.046 Competitiveness (竞争力)*0.042	Branding
5	Honesty (诚信)*0.170 Principle (原则)*0.093 Reflection (体现)*0.072 Promise (承诺)*0.034 Guarantee (保障)*0.034	Trustworthiness
6	Team (团队)*0.113 Enthusiasm (热情)*0.063 Be filled (充满)*0.050 Passion (激情)*0.039 Devotion (热爱)*0.037	Team environment
7	Production (作品)*0.220 View (意见)*0.044 Exchange (交流)*0.031 Suggestion (建议)*0.030 Idea (想法)*0.029	Reasonable suggestion
8	Modification (修改)*0.238 Patience (耐心)*0.157 After-sales service (售后服务)*0.046 Revision (修改意见)*0.030 For free (免费)*0.029	Modification and after-sales service
9	Cooperation (合作)*0.161 Trust (信任)*0.064 Relationship (关系)*0.027 Long term (长期)*0.026 Establish (建立)*0.019	Customer relationship management

Table 1: Continued.

Topic	Clusters of keywords	Competency dimensions
10	Innovation (创意)*0.266 Responsible (负责)*0.073 Promise (保证)*0.048 Problem solving (解决问题)*0.037 Originality (原创)*0.031	Innovation ability
11	Market (市场)*0.134 Perspective (角度)*0.058 Market positioning (定位)*0.056 User (用户)*0.033 Analysis (分析)*0.023	Customers' industry background
12	Service (服务)*0.195 Experience (经验)*0.081 Provide (提供)*0.060 Concept (理念)*0.049 Accumulation (积累)*0.019	Profession experience
13	Concentration (用心)*0.273 Diligence (努力)*0.162 Manage (经营)*0.054 Contribution (付出)*0.034 Win-win (共赢)*0.033	Professional dedication
14	Team (团队)*0.330 Goal (目标)*0.057 Member (成员)*0.043 Excellent (优秀)*0.036 Power (力量)*0.025	Team composition
15	Professional (专业)*0.277 Service (服务)*0.109 Quality (品质)*0.041 Acceptance (认可)*0.033 Word-of-mouth (口碑)*0.030	Customer acceptance
16	Communication (沟通)*0.268 Demand (需求)*0.127 Timing (时间)*0.053 Active (主动)*0.035 Plan (方案)*0.032	Communication ability
17	Achievement (成功)*0.269 Grow (成长)*0.166 Powerful (强大)*0.086 Specifics (细节)*0.033 Vision (发展)*0.023	Achievement orientation
18	Ability (能力)*0.367 Entrepreneur (创业)*0.081 Experience (经历)*0.058 Background (背景)*0.052 Possess (具备)*0.049	Entrepreneurial experience

The English words represent the translations of the original Chinese words in parentheses, and the decimals following "*" indicate the probability that each word belongs to that topic.

summarize team composition and team environment as team management, likewise. Subsequently, the three-level KSAT competency model is constructed and presented in Figure 4, and the descriptions of the competency criteria in the model are presented in Table 2. The model embraces a comprehensive and hierarchical set of competency criteria that offers the researchers and practitioners in KIC the flexibility to systematically build, verify, and change the SPs' selection and evaluation mechanisms to suit their requirements.

4.4. Weights of Competency Criteria. Due to the KIC platform's goals on healthy and sustainable operation process, the proposed KSAT competency model can be further employed to evaluate the SPs' competency and select and keep the competent SPs. Identifying the importance of each competency criterion plays a significant role in better guiding both the SPs and the KIC platform to plan, design, and implement mechanisms and strategies for sustainable development and management. We develop an online

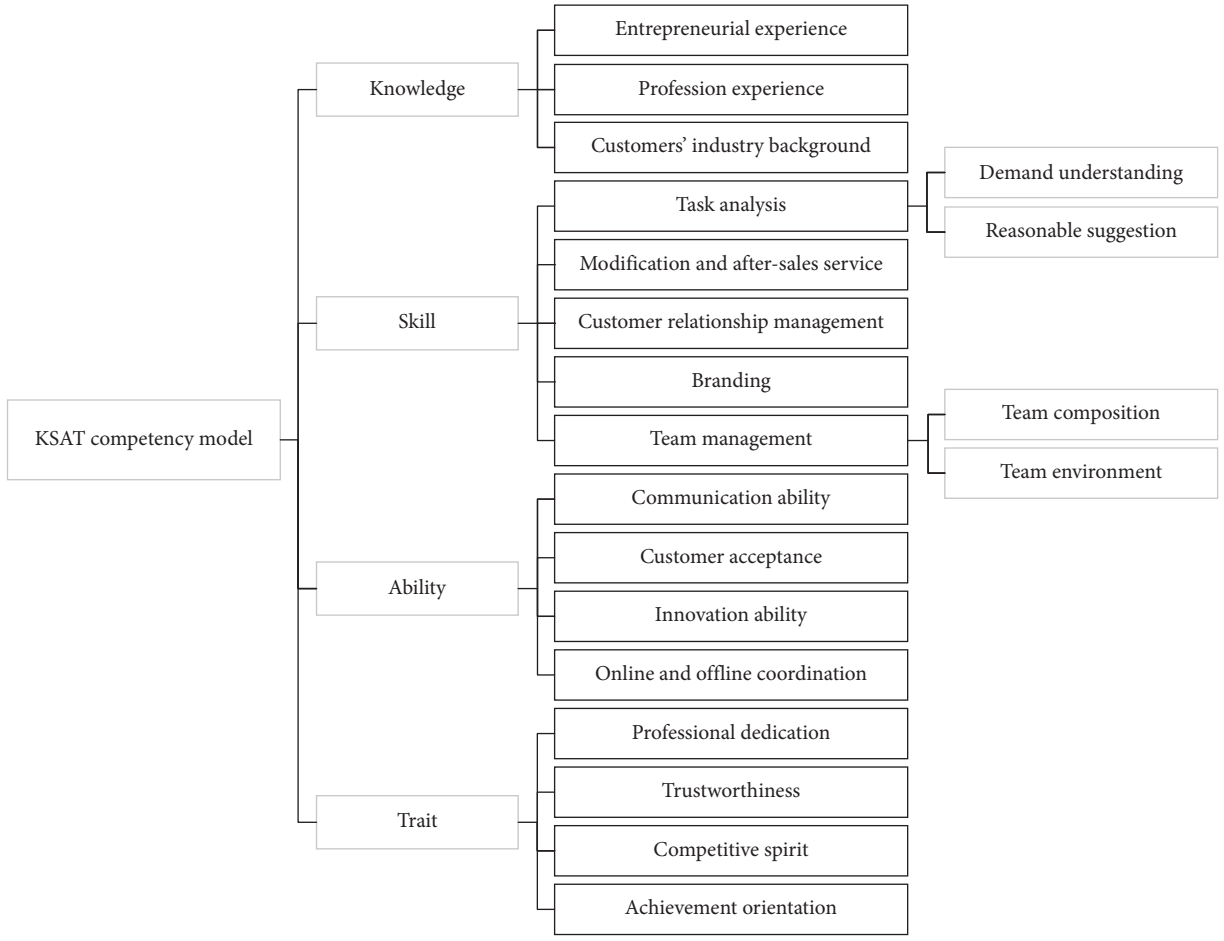


Figure 4: KSAT competency model for platforms estimating SPs' performance in KIC context.

questionnaire involving the aforementioned KSAT competency criteria shown in Table 2 and employ the BWM to determine the weights of competency criteria. In this step, we invite 15 experts and ask for their opinion. Of the 15 respondents involved in this research, 9 with over 5 years' working experience are employed as managers in crowdsourcing companies, like ZBJ.COM. The remaining 6 are researchers in the crowdsourcing field with an average of 7 years' research experience. To ensure that all the respondents have adequate information to conduct the comparisons, the description of the BWM and the competency factors and dimensions are also provided.

In this paper, the importance of the competency criteria for the KIC platform to evaluate the SPs' competency is examined and evaluated based on the four competency factors (i.e., knowledge, skill, ability, and trait) and twenty competency dimensions (eighteen in level 2 and two in level 3). Tables 3–5 present the local and global weights of the competency factors and dimensions in levels 1–3, respectively.

5. Discussion

In this study, our concentration is to investigate and understand the SPs' competency by recognizing detailed competency criteria (i.e., competency factors and competency dimensions) in KIC. We first explore the successful SPs' experience sharing information using the LDA approach to identify natural and hidden competency dimensions. Then, we construct the KSAT competency model and determine the weights of competency criteria for the KIC platform evaluating the SPs' competency. Such information acts as a source on how to evaluate, manage, and encourage SPs.

5.1. Results Analysis Related to KSAT Competency Model. The highly competent SPs are necessary for the KIC platform to meet fluctuating demands for numerical and functional flexibility [80]. In KIC markets, the SPs' competency acts as decision factors that have a great influence on the quality of task outcomes, customer satisfaction and loyalty, and the platform's reputation and sustainable growth. This study aims to understand the SPs' competency in KIC and identifies a list of competency criteria by extracting from the competent SPs' experience. The KSAT competency model is subsequently constructed based on competency theory that Mirabile [14] conceptualized competency as "knowledge, skills, abilities, or other characteristics" that differentiate high from low performance. Specifically, the KSAT

Table 2: Description of identified competency factors.

Competency factors	Description	Competency dimensions	Description
Knowledge	The prior trial and error experience related to successful crowdsourcing business launches, development, and resolution of emerging problems [15, 78], that can help SPs to be innovative, trigger new ideas, and seize opportunities [13, 79]	Entrepreneurial experience	The experience that increases with every firm launched and helps to acquire detailed knowledge of administrative procedures for registration, corporate tax declarations, and social security [15]
		Profession experience	The knowledge and experience in the same industry domains where firms launch [15], that help to perform tasks, manage, and run a business [79]
		Customers' industry background	The knowledge and experience related to customers' industry domains which can help better understand customer requirements, and output satisfying service products [13]
Skill	The domain-specific occupational expertise required to perform knowledge-intensive tasks [13, 80]	Task analysis	The expertise to deal effectively with customers' requirements and problems of service demands [81, 82]
		Demand understanding	The ability seeking to understand customers' expressed needs and requirements [83–85]
		Reasonable suggestion	The ability seeking to develop superior suggestions and solutions to meet customers' demands [83–85]
		Modification and after-sales service	The service supports such as free modification of designed service products and problem-solving provided by service providers once the transaction takes place [86]
		Customer relationship management	The ability to build and maintain friendly, warm relationships with customers by managing their buying behaviors and feedback for continual sales and sustainable collaboration relationship [13, 83, 87]
		Branding	The ability to promote their own service brand image [86, 88, 89] and generate word-of-mouth among customers [16]
		Team management	The ability to have a detailed and accurate understanding of how the organization operates functionally via team member selection, team commitment building, and promotion [54, 78, 83] to improve teamwork and cooperation [13]
Skill	The domain-specific occupational expertise required to perform knowledge-intensive tasks [13, 80]	Team composition	The members of the team are with functional heterogeneity and have different backgrounds and knowledge [83], which can complement the way for firms to obtain information benefits [78, 90]
		Team environment	The ability to create a harmonious and positive team atmosphere [83]
Ability	Task performance or the effective outcomes achieved by crowdsourcing business process [7]	Communication ability	The ability to keep customers and cooperators informed with useful information [13]
		Customer acceptance	The ability to influence customers and gain their approvals and trust [13]
		Innovation ability	The ability to be innovative in thinking and create novel ideas and solutions to problems [54]
		Online and offline coordination	The ability to coordinately operate online and offline business [91]

Table 2: Continued.

Competency factors	Description	Competency dimensions	Description
Trait	The physical characteristics and consistent responses to situations or information [13]	Professional dedication	The willingness to put the team's needs before personal needs, and align own behavior with the needs, priorities, and goals of the business to meet business goals [13]
		Trustworthiness	The confident expectations of customers on SPs' performing a particular transaction, reflected by customer reviews, store images, and transaction performance [86, 92]
		Competitive spirit	The willingness to confront challenges and make efforts to stand out from the mass peers [13]
		Achievement orientation	The concerns for working well or for achieving business goals and ambition [13]

Table 3: Weights of the competency criteria by BWM.

Competency factors	Weight	Competency dimensions in level 2	Weight
Knowledge	0.215	Entrepreneurial experience	0.251
		Profession experience	0.445
		Customers' industry background	0.284
Skill	0.436	Task analysis	0.322
		Modification and after-sales service	0.155
		Customer relationship management	0.232
		Branding	0.166
		Team management	0.126
Ability	0.228	Communication ability	0.437
		Customer acceptance	0.278
		Innovation ability	0.135
		Online and offline coordination	0.150
Trait	0.121	Professional dedication	0.208
		Trustworthiness	0.433
		Competitive spirit	0.184
		Achievement orientation	0.175

Table 4: Weights of competency dimensions in level 3 by BWM.

Competency dimensions in level 2	Competency dimensions in level 3	Weight
Task analysis	Demand understanding	0.577
	Reasonable suggestion	0.423
Team management	Team composition	0.647
	Team environment	0.353

competency model proposed in this work includes knowledge (entrepreneurial experience, profession experience, and customers' industry background), skill (task analysis, modification and after-sales service, customer relationship management, branding, and team management), ability (communication ability, customer acceptance, innovation ability, and online and offline coordination), and trait (professional dedication, trustworthiness, competitive spirit, and achievement orientation).

Knowledge has been deemed to be a key factor for success in the empirical research of entrepreneurship and crowdsourcing [15, 79]. In KIC, the SPs are mainly regarded as self-employed workers or small- and medium-sized enterprises (SMEs) and viewed as entrepreneurs [93, 94]. Entrepreneurial experience is thought to be mandatory when starting up a new business as it can help focus on strategic issues to alleviate the liabilities of newness, such as establishing new business partnerships [15]. Also, entrepreneurs will be more sensitive to business opportunities, future technologies, and customer demand via learning by doing process [15, 95]. Furthermore, profession experience can boost experiential learning, enhance the development of operational knowledge, and ease the transfer process of prior knowledge before starting the new business [15]. Relying on entrepreneurial and professional experience, entrepreneurs can also save costs on routine development. In addition to its importance in entrepreneurship, knowledge is the determinant of the good performance in crowdsourcing.

Table 5: Global weights of competency dimensions.

Competency dimensions	Global weight	Rank
Customer relationship management	0.101	1
Communication ability	0.100	2
Profession experience	0.096	3
Demand understanding	0.081	4
Branding	0.072	5
Modification and after-sales service	0.067	6
Customer acceptance	0.063	7
Customers' industry background	0.061	8
Reasonable suggestion	0.059	9
Entrepreneurial experience	0.054	10
Trustworthiness	0.052	11
Team composition	0.036	12
Online and offline coordination	0.034	13
Innovation ability	0.020	14
Professional dedication	0.018	15
Competitive spirit	0.014	16
Achievement orientation	0.012	17
Team environment	0.011	18

According to [96], knowledge diversity of the individual crowds facilitates all types of contribution to open innovation projects as having knowledge in diverse fields allows the contributors to understand the task requirements [97] or blend disparate solution elements in novel ways. Being familiar with the unique industry background of the targeted customers will help the SPs better understand customer requirements, cut to the heart of the matter, and achieve consensus on the solution with customers [97].

Skill, defined as the domain-specific occupational expertise in this work, appears to be advantageous for both the KIC platform and the SPs. In the crowdsourcing context, skill is regarded as the basis for preselecting the proper SPs in auction systems as it demonstrates how the SPs are at doing particular tasks [40, 41]. Rather than exploring and extracting dimensions of skill, the state of the art mainly concentrated on the indictors in a specific context to measure the level of skill or directly assign a numerical value to quantify [41]. The KIC tasks are commonly complex and creative, which cannot be done by computers [3]. A complete understanding of the task requirements and sound advice can help service requesters to better visualize the desired outcomes, thus reducing the perception gap between the service requesters and the SPs [97]. We conclude the two subdimensions (i.e., demand understanding and reasonable suggestion) as task analysis, which is viewed as a critical starting point to the task success. Meanwhile, the SPs in KIC have to offer modification and after-sales service, because the task outcome is not going to be perfect and matches the customers' expectations all at once. Offering such post-sale services could reduce customers' risk perception of task failure and poor transaction experience. Accordingly, offering high-quality services is a powerful way to enhance customer satisfaction and thus retain the targeted customers [86]. Customer relationship management, a strong tool in marketing and business [87], is another competency dimension of the SPs to maintain a long-term relationship with their service requesters. In addition, the SPs commonly own one or several virtual stores on the KIC platform. Brand image of the stores plays a vitally important role in attracting customers as it can create a positive attitude towards the SPs' virtual stores that will, in turn, encourage the intention to repurchase and produce positive word-of-mouth [86]. Furthermore, team management is also mentioned by the SPs as a core competency dimension in conducting KIC business. In China, many SPs register on the KIC platform as a team so as to organically aggregate human resources and leverage each member's strengths. However, existing research rarely explores the team management competency of SPs in the environment of crowdsourcing.

Despite the domain-specific skill, ability, defined as the task performance or the effective outcomes achieved during the crowdsourcing business process, is also regarded as a main competency factor by the SPs. The KIC activities involve many intelligence-related tasks and intangible services that everyone is likely to have a different understanding of the task requirements and outcomes. To mitigate such cognitive differences, communication is essential. Concretely, the accurate and useful information about the understanding of customer demand and expected outcomes, problem-solving plans, and modification suggestions need to be efficiently and actively conveyed by the SPs. As communication helps narrow cognitive gaps, innovation ability accounts for divergent thinking and creative ideas, differentiating services, and products from others [98]. In innovation contests, the innovativeness of solutions is considered an important reference for the selection of the winner [17]. In addition, customer acceptance reflects the degree of customers' trust in the SPs and the probability that the customers expect to obtain high-quality outcomes from the SPs [40]. Gaining customers' trust and acceptance has always been considered as one of the ultimate goals of marketing [92]. Additionally, the online SPs may also offer services offline as a complementary sales channel, which is a distraction of SPs' time and effort on online business. Hence, coordinately operating online and offline crowdsourcing business at the same time is a challenge for the SPs.

Besides, personal traits play a critically important role in determining the SPs' competency in the KIC activities. Previous research has widely investigated the significance of personal traits in exploring individual work performance or entrepreneur success [81]. Batey and Furnham [99] found that the "big five" personality traits account for up to 47% of the variation in divergent thinking. A meta-analysis by Feist [100] showed that the individuals who are open to new experiences, conscientious, hostile, confident, and emotionally impulsive are more likely to generate creative outcomes. Findings by Sebora et al. [81] asserted that the achievement orientation of the founders is positively related to the success of e-commerce entrepreneurial ventures as it helps the entrepreneurs overcome obstacles and compensates for other weaknesses. In crowdsourcing delivery, personal traits were considered as a main criterion in [7] for evaluating the delivers' competence and responsibility is one

of the most important subcriteria when quantifying the competence score. These researches indicate that personal traits have been found to be a robust factor of high-quality outcomes and performance [81]. Specifically, in knowledge-intensive crowdsourcing, the SPs regard professional dedication, trustworthiness, competitive spirit, and achievement orientation as the competency dimensions.

5.2. Results Analysis Related to Competency Importance. The BWM results show that the competency criteria have different importance when the KIC platform evaluates the SPs' competency. Table 5 indicates that "customer relationship management" and "communication ability" have the highest priorities of all the competency factors, while "team environment" is the least important. "Customer relationship management" emerged in the 1970s [101] as a useful tool for managing and optimizing sales-force automation within companies, enhancing customer satisfaction and loyalty [102, 103], and consequently reaching and retaining long-term partnerships with customers [104]. In this sense, the SPs with high-level customer relationship management will encourage the loyal end customers to retain on the platform, thus helping generate sustainable value and maintaining long-term growth. This attests the business model of Upwork (one of the largest online KIC marketplaces) charging its fees on a sliding scale to encourage the longer-term relations between the SPs and service requesters [105].

As to "communication ability," the KIC activities involve intensive and dynamic interaction among the end customers, the platform operators, and the SPs. Online chatting, instant messaging, and social media are the most popular ways for the SPs to interact with their customers [16]. With more information processed, communication about the task ideas may shift the interpretation or understanding of the task at hand [106], towards a way that helps to reach consensus between the SPs and the service requesters about the way how tasks are performed and the form of outcomes, thus reducing the gap between the service requesters' perception and expectation about the service and enhancing their overall satisfaction. Foundational research emphasized the role of communication in the form of dialogue, feedback, and other contextual factors, in the way a message is received and interpreted by the viewers, as well as how they respond [106, 107], which indicates that the KIC platform may develop different types of communication systems to assist the SPs with their interaction with customers in various manners. Also, a high-level of communication ability that focuses on professional interactions, being honest and responsive, and respecting customer culture during interactions, will impress the customers with professional knowledge, and high-quality services, leading to a good customer relationship and reputation [89]. Therefore, it is suggested that the SPs need to focus on the language used, cultural awareness, and promptness of their response during the interaction process. While "team environment" is weighted as the least important competency factor by experts, it is in line with the actual situation that, as a platform does not hire the SPs, it is relatively impossible for a platform to observe their interactions within the organization. However, the KIC platform need to pay attention to this factor as it is considered as an essential dimension by the SPs that influences their competency and performance.

As shown in Table 3, "skill" is the most important of all the four competency factors. The main reason for that is domain-specific skills are regarded as the key to a successful business [108, 109]. Unlike the short and low-complexity simple tasks [110], the KIC tasks are domain-specific, high in complexity, and not easy to decompose apparently, which put high requirements on the participating SPs [111]. Gong [3] pointed out that the lack of domain-specific skill and expertise has limited the development of the KIC marketplaces, which revealed the dominating role of the SPs' skill in the KIC context. Further, the SPs' skill demonstrates how capable he or she is at doing particular tasks in the domain, and the SPs with high skill are expected to contribute to high-quality outcomes [40, 108]. The prerequisite for the SPs conducting a specific task is that they have the relevant and necessary skills to perform at the required quality [112]. This implies that the KIC platform could grade the SPs' skill levels, such that the resources can be strategically allocated and assigned. For example, the high-skill level SPs could be assigned to perform more complex KIC tasks that require a comprehensive usage of different skills. For the SPs, they need to develop and enhance different types of skills, so as to perform more complex tasks to gain more income.

The "ability" factor ranks in second place, which means that, in absolute terms, it is still more important than "knowledge" and "trait." Among all the four competency factors, it is not surprising that "knowledge" and "trait" are weighted as the two least important. This is in line with the actual situation involving KIC activities in China, due to the fact that, to attract more SPs to participate in KIC activities, the input control of the KIC platform is relatively weak. For example, the steps for applicants to be SPs at ZBJ.COM (https://help.zbj.com/fw/detail?articleId=14762) are (1) registering by a telephone number and a password; (2) filling in some required information, such as self-introduction, location, e-mail address, and task types willing to perform; and (3) uploading photos of identification card. The steps of epwk.com (https://i.epwk.com/User/Basicinfo/index.html) and 680.com (http://help.680.com/view_4.html) are similar to ZBJ.COM.

In addition, the fact that "knowledge" is not ranked the lowest infers that entrepreneurial- and business-related knowledge experience are also regarded by the experts as essential competency dimensions in demonstrating the SP's competency and performance.

6. Conclusion

In this study, by combining text mining and decision-making techniques, we conducted a comprehensive competency analysis of SPs in the environment of KIC. In this process, we leveraged online interview records posted by several KIC platforms in China, which includes the successful SPs' opinions about the qualifications to obtain sustainable business growth and conduct their career well. By applying the topic modelling approach to these materials, we identified four competency factors and twenty competency dimensions in general and thus constructed the hierarchical KSAT competency model. Further, we employed the BWM to identify the weights and priorities of the competency criteria in the KSAT competency model. The relationships between the competency criteria are also discussed, leading to the following conclusions:

(1) The proposed KSAT competency model gives practitioners of the KIC platform a comprehensive vision of SPs' performance standards in the context of KIC. Further, it also provides the practitioners flexibility to choose different criteria according to different application scenarios and objective. In KIC markets, the SPs, viewed as entrepreneurs, are varied in backgrounds, skills, and abilities and contributions, leading to great challenges to their management for the KIC platform. Also, the SPs in the start-up stages lack the benefits of continuous competition. To stabilize the market environment and advances the platform's interests, the platform managers have to develop better input control mechanisms, incentive mechanisms, and SPs' life-cycle management systems to retain and incubate the valuable SPs. Also, the KSAT competency model can be considered as a self-check or learning system rather as a mere grading tool to help the SPs realize their strengths and weaknesses and allow them to improve in aspects where they are weaker, in a structured and targeted manner.

(2) In the operation system of KIC, the platform is the rule maker and it is better for the SPs to focus on developing and enhancing those competency criteria that the decision-makers considered as important to achieve a higher ranking in the platform's evaluation system. Particularly, as "skill" and "ability" are viewed as the most important competency factors, it is necessary for the SPs to master not only excellent occupational expertise but also comprehensive business operation capabilities to get ahead in the fierce competition.

(3) The KIC platform also needs to pay attention to the competency criteria weighted as least important ones, such as team environment, when developing evaluation mechanisms and management systems. Insufficient attention to these criteria may lead to unfair or biased evaluations as the SPs view them as essential elements that affect their business success.

This research can be extended in several ways. First, although the LDA model shows popular applications in text analysis, it still has some shortcomings. As an unsupervised learning algorithm, the LDA inherently has disadvantages in fully understanding natural languages but it requires no human intervention. Future research may use supervised learning algorithms for identifying the SPs' competency components from online interview records. Second, further longitudinal exploration can be conducted to analyze the antecedent and consequential relationships among these competency criteria. Experiments are also needed to help the crowdsourcing researchers establish causality by eliminating extraneous factors and endogeneity issues. Additionally, experimental approaches will possibly enable further explorations into situations where various management mechanisms are existing, in order that more insights can be derived from their interactions and how they could jointly work to enhance performance and customer purchasing intentions. Third, this research is limited by the single source of materials, which may result in incomplete competency identification. It might be necessary to collect and use more data from different data sources in future research to obtain more generalized and significant findings. Finally, it may be worthwhile to extend our study to where it could include other industries, other text mining techniques, and other data as future research for more significant analytical results on the SPs' competency and performance in KIC.

Data Availability

The data used in this paper consist of two parts: the online interview records of SPs posted on the KIC platforms and the BWM questionnaires that were sent to the selected experts. They are available upon request to the corresponding author at wx921@163.com.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Science and Technology Support Program of China (project no. 2018YFB1403602), the Fundamental Research Funds for the Central Universities (project no. 2020CDCGJX019), the Graduate Research and Innovation Foundation of Chongqing (Project no. CYS20007), the Technological Innovation and Application Program of Chongqing (project no. cstc2019jscx-mbdxX0008), and the National Social Science Foundation of China (project no. 18BJY066). The authors would like to thank the members in the team of R&D of technology consulting service platform for their assistance and guidance.

References

[1] X. Tian, J. Shi, and X. Qi, "Talent crowdsourcing via stochastic sequential assignments," *SSRN Electronic Journal*, 2018.

[2] A. Kittur, J. V. Nickerson, M. Bernstein et al., "The future of crowd work," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pp. 1301–1317, San Antonio, TX, USA, February 2013.

[3] Y. Gong, "Estimating participants for knowledge-intensive tasks in a network of crowdsourcing marketplaces," *Information Systems Frontiers*, vol. 19, no. 2, pp. 301–319, 2017.

[4] M. Wessel, F. Thies, and A. Benlian, "Opening the floodgates: the implications of increasing platform openness in crowdfunding," *Journal of Information Technology*, vol. 32, no. 4, pp. 344–360, 2017.

[5] F. Thies, M. Wessel, and A. Benlian, "Network effects on crowd funding platforms: exploring the implications of relaxing input control," *Information Systems Journal*, vol. 28, no. 6, pp. 1239–1262, 2018.

[6] M. Sun and E. Tse, "The resource-based view of competitive advantage in two-sided markets," *Journal of Management Studies*, vol. 46, no. 1, pp. 45–64, 2009.

[7] L. Li, X. Wang, and J. Rezaei, "A bayesian best-worst method-based multicriteria competence analysis of crowdsourcing delivery personnel," *Complexity*, vol. 2020, Article ID 4250417, 17 pages, 2020.

[8] M. Mulder, "Conceptions of professional competence," in *International Handbook of Research in Professional and Practice-Based Learning*, C. H. H. G. S. Billett, Ed., Springer, Dordrecht, The Netherlands, pp. 107–137, 2014.

[9] N. Otani, Y. Baba, and H. Kashima, "Quality control of crowdsourced classification using hierarchical class structures," *Expert Systems with Applications*, vol. 58, pp. 155–163, 2016.

[10] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–40, 2018.

[11] K. Kritikos, B. Pernici, P. Plebani et al., "A survey on service quality description," *ACM Computing Surveys*, vol. 46, no. 1, pp. 1–58, 2013.

[12] P. Hager, A. Gonczi, and J. Athanasou, "General issues about assessment of competence," *Assessment & Evaluation in Higher Education*, vol. 19, no. 1, pp. 3–16, 1994.

[13] L. M. Spencer and S. M. Spencer, *Competence at Work: Models for Superior Performance*, John Wiley & Sons, London, UK, 1993.

[14] R. J. Mirabile, "Everything you wanted to know about competency modeling," *Training & Development*, vol. 51, no. 8, p. 73, 1997.

[15] A. Oe and H. Mitsuhashi, "Founders' experiences for startups' fast break-even," *Journal of Business Research*, vol. 66, no. 11, pp. 2193–2201, 2013.

[16] L. Aroean, D. Dousios, and N. Michaelidou, "Exploring interaction differences in microblogging word of mouth between entrepreneurial and conventional service providers," *Computers in Human Behavior*, vol. 95, pp. 324–336, 2019.

[17] T. Mack and C. Landau, "Winners, losers, and deniers: self-selection in crowd innovation contests and the roles of motivation, creativity, and skills," *Journal of Engineering and Technology Management*, vol. 37, pp. 52–64, 2015.

[18] L. B. Jeppesen and K. R. Lakhani, "Marginality and problem-solving effectiveness in broadcast search," *Organization Science*, vol. 21, no. 5, pp. 1016–1033, 2010.

[19] E. H. Hwang, P. V. Singh, and L. Argote, "Jack of all, master of some: information network and innovation in crowdsourcing communities," *Information Systems Research*, vol. 30, no. 2, pp. 389–410, 2019.

[20] J. N. Cummings, J. A. Espinosa, and C. K. Pickering, "Crossing spatial and temporal boundaries in globally distributed projects: a relational model of coordination delay," *Information Systems Research*, vol. 20, no. 3, pp. 420–439, 2009.

[21] F. Leal, B. M. Veloso, B. Malheiro, H. González-Vélez, and J. C. Burguillo, "Scalable modelling and recommendation using wiki-based crowdsourced repositories," *Electronic Commerce Research and Applications*, vol. 33, Article ID 100817, 2019.

[22] A. R. Kurup and G. P. Sajeev, "A task recommendation scheme for crowdsourcing based on expertise estimation," *Electronic Commerce Research and Applications*, vol. 41, Article ID 100946, 2020.

[23] X. Zhang, B. Gong, H. Ni, Z. Liang, and J. Su, "Identifying participants' characteristics influencing participant estimation in knowledge-intensive crowdsourcing," in *Proceedings of the 2019 8th International Conference on Industrial Technology and Management (ICITM)*, Cambridge, UK, March 2019.

[24] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[25] J. Howe, "Croudsourcing: why the power of the crowd is driving the future of business," *Nieman Reports*, vol. 62, no. 4, p. 47, 2008.

[26] Y. Xu, D. E. Ribeiro-Soriano, and J. Gonzalez-Garcia, "Crowdsourcing, innovation and firm performance," *Management Decision*, vol. 53, no. 6, pp. 1158–1169, 2015.

[27] J. Oosterman, A. Bozzon, G.-J. Houben et al., "Crowd vs. experts," in *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*, pp. 567-568, Seoul, South Korea, April 2014.

[28] B. L. Bayus, "Crowdsourcing new product ideas over time: an analysis of the Dell IdeaStorm community," *Management Science*, vol. 59, no. 1, pp. 226–244, 2013.

[29] M. Christoforaki and P. G. Ipeirotis, "A system for scalable and reliable technical-skill testing in online labor markets," *Computer Networks*, vol. 90, pp. 110–120, 2015.

[30] A. Hagiu and H. Hałaburda, "Information and two-sided platform profits," *International Journal of Industrial Organization*, vol. 34, no. 1, pp. 25–35, 2014.

[31] J. Brustle, Y. Cai, F. Wu, and M. Zhao, "Approximating gains from trade in two-sided markets via simple mechanisms," in *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 589-590, Cambridge, MA, USA, June 2017.

[32] D. Vakharia and M. Lease, "Beyond AMT: an analysis of crowd work platforms," 2013, http://arxiv.org/abs/1310.1672.

[33] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2296–2319, 2016.

[34] S. Zhu, S. Kane, J. Feng, and A. Sears, "A crowdsourcing quality control model for tasks distributed in parallel," in *Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts-CHI EA'12*, pp. 2501–2506, ACM Press, New York, NY, USA, 2012.

[35] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "Cdas: a crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 510, pp. 1040–1051, 10 edition, 2012.

[36] L. Shamir, D. Diamond, and J. Wallin, "Leveraging pattern recognition consistency estimation for crowdsourcing data

analysis," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 474–480, 2016.

[37] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi, "Community-based bayesian aggregation models for crowdsourcing," in *Proceedings of the 23rd International Conference on World Wide Web-WWW'14*, pp. 155–164, Seoul, South Korea, April 2014.

[38] K. Li, S. Wang, and X. Cheng, "Crowdsourcee evaluation based on persuasion game," *Computer Networks*, vol. 159, pp. 1–9, 2019.

[39] D. Dang, Y. Liu, X. Zhang, and S. Huang, "A crowdsourcing worker quality evaluation algorithm on MapReduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 1879–1888, 2016.

[40] 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.

[41] X. Zhang, B. Gong, Y. Cao, Y. Ding, and J. Su, "Investigating participants' attributes for participant estimation in knowledge-intensive crowdsourcing: a fuzzy DEMATEL based approach," *Electronic Commerce Research*, vol. 145, pp. 1–32, 2020.

[42] D. C. McClelland, "Testing for competence rather than for "intelligence"" *American Psychologist*, vol. 28, no. 1, pp. 1–14, 1973.

[43] T. R. Athey and M. S. Orth, "Emerging competency methods for the future," *Human Resource Management*, vol. 38, no. 3, pp. 215–225, 1999.

[44] R. Gibbons, "On competence: a critical analysis of competence-based reforms in higher education," *The Journal of Higher Education*, vol. 51, no. 6, pp. 695–697, 1980.

[45] R. Kurz and D. Bartram, "Competency and individual performance: modelling the world of work," in *Organizational Effectiveness: The Role of Psychology*, I. T. Robertson, M. Callinan, and D. Bartram, Eds., John Wiley & Sons, Ltd, Chichester, UK, pp. 227–255, 2002.

[46] D. Bartram, "The great eight competencies: a criterion-centric approach to validation," *Journal of Applied Psychology*, vol. 90, no. 6, pp. 1185–1203, 2005.

[47] L. Ploum, V. Blok, T. Lans, and O. Omta, "Toward a validated competence framework for sustainable entrepreneurship," *Organization & Environment*, vol. 31, no. 2, pp. 113–132, 2018.

[48] S. Krumm, J. Kanthak, K. Hartmann, and G. Hertel, "What does it take to be a virtual team player? The knowledge, skills, abilities, and other characteristics required in virtual teams," *Human Performance*, vol. 29, no. 2, pp. 123–142, 2016.

[49] G. Hertel, U. Konradt, and K. Voss, "Competencies for virtual teamwork: development and validation of a web-based selection tool for members of distributed teams," *European Journal of Work and Organizational Psychology*, vol. 15, no. 4, pp. 477–504, 2006.

[50] C. C. Cogliser, W. L. Gardner, M. B. Gavin, and J. C. Broberg, "Big five personality factors and leader emergence in virtual teams," *Group & Organization Management*, vol. 37, no. 6, pp. 752–784, 2012.

[51] T. J. Maurer and M. Lippstreu, "Expert vs. general working sample differences in KSAO improvability ratings and relationships with measures relevant to occupational and organizational psychology," *Journal of Occupational and Organizational Psychology*, vol. 81, no. 4, pp. 813–829, 2008.

[52] C. Prahalad and G. Hamel, "The core competence of the corporation," *Strategic Learning in a Knowledge Economy*, Routledge, Abingdon, UK, pp. 3–22, 2000.

[53] R. Boyatzis, *The Competent Manager: A Model for Effective Performance*, Wiley, Hoboken, NJ, USA, 1982.

[54] W.-W. Wu and Y.-T. Lee, "Developing global managers' competencies using the fuzzy DEMATEL method," *Expert Systems with Applications*, vol. 32, no. 2, pp. 499–507, 2007.

[55] C. Lucas, R. A. Nielsen, M. E. Roberts, B. M. Stewart, A. Storer, and D. Tingley, "Computer-assisted text analysis for comparative politics," *Political Analysis*, vol. 23, no. 2, pp. 254–277, 2015.

[56] D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 55–65, 2010.

[57] K. Chen, G. Kou, J. Shang, and Y. Chen, "Visualizing market structure through online product reviews: integrate topic modeling, TOPSIS, and multi-dimensional scaling approaches," *Electronic Commerce Research and Applications*, vol. 14, no. 1, pp. 58–74, 2015.

[58] R.-S. Wu and P.-H. Chou, "Customer segmentation of multiple category data in e-commerce using a soft-clustering approach," *Electronic Commerce Research and Applications*, vol. 10, no. 3, pp. 331–341, 2011.

[59] U. H. Graneheim and B. Lundman, "Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness," *Nurse Education Today*, vol. 24, no. 2, pp. 105–112, 2004.

[60] A. Ansari, Y. Li, and J. Z. Zhang, "Probabilistic topic model for hybrid recommender systems: a stochastic variational bayesian approach," *Marketing Science*, vol. 37, no. 6, pp. 987–1008, 2018.

[61] Y. Lu, P. Zhang, J. Liu, J. Li, and S. Deng, "Health-related hot topic detection in online communities using text clustering," *PLoS One*, vol. 8, no. 2, Article ID e56221, 2013.

[62] J. Mou, G. Ren, C. Qin, and K. Kurcz, "Understanding the topics of export cross-border e-commerce consumers feedback: an LDA approach," *Electronic Commerce Research*, vol. 19, no. 4, pp. 749–777, 2019.

[63] B. Cao, X. Liu, J. Liu, and M. Tang, "Domain-aware mashup service clustering based on LDA topic model from multiple data sources," *Information and Software Technology*, vol. 90, pp. 40–54, 2017.

[64] W. Wang, Y. Feng, and W. Dai, "Topic analysis of online reviews for two competitive products using latent Dirichlet allocation," *Electronic Commerce Research and Applications*, vol. 29, pp. 142–156, 2018.

[65] F. Al-Obeidat, B. Spencer, and E. Kafeza, "The opinion management framework: identifying and addressing customer concerns extracted from online product reviews," *Electronic Commerce Research and Applications*, vol. 27, pp. 52–64, 2018.

[66] K. Bastani, H. Namavari, and J. Shaffer, "Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints," *Expert Systems with Applications*, vol. 127, pp. 256–271, 2019.

[67] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[68] J. Chang, S. Gerrish, C. Wang, J. Boydgraber, and D. M. Blei, "Reading tea leaves: how humans interpret topic models," in *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pp. 288–296, Vancouver, Canada, December 2009.

[69] J. Rezaei, "Best-worst multi-criteria decision-making method," *Omega*, vol. 53, pp. 49–57, 2015.
[70] J. Rezaei, "Best-worst multi-criteria decision-making method: some properties and a linear model," *Omega*, vol. 64, pp. 126–130, 2016.
[71] J. Rezaei, "A concentration ratio for nonlinear best worst method," *International Journal of Information Technology & Decision Making*, vol. 19, no. 3, pp. 891–907, 2020.
[72] A. J. Bradley, "Training in the mitigation of anchoring bias: a test of the consider-the-opposite strategy," *Learning and Motivation*, vol. 53, pp. 36–48, 2016.
[73] X. Mi, M. Tang, H. Liao, W. Shen, and B. Lev, "The state-of-the-art survey on integrations and applications of the best worst method in decision making: why, what, what for and what's next?" *Omega*, vol. 87, pp. 205–225, 2019.
[74] Sharing Economy Research Center, "China Sharing Economy Development Report 2020," *State Information Center, Beijing, China*, https://www.ndrc.gov.cn/xxgk/jd/wsdwhfz/202003/t20200310_1222769.html., 2020.
[75] X. Tian, G. Kou, and W. Zhang, "Geographic distance, venture capital and technological performance: evidence from Chinese enterprises," *Technological Forecasting and Social Change*, vol. 158, Article ID 120155, 2020.
[76] N. Aletras and M. Stevenson, "Measuring the similarity between automatically generated topics," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 22–27, Stroudsburg, PA, USA, April 2014.
[77] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009): A Meeting of SIGDAT, a Special Interest Group of ACL*, pp. 248–256, Singapore, August 2009.
[78] C. M. Beckman, M. D. Burton, and C. O'Reilly, "Early teams: the impact of team demography on VC financing and going public," *Journal of Business Venturing*, vol. 22, no. 2, pp. 147–173, 2007.
[79] M. W. Staniewski, "The contribution of business experience and knowledge to successful entrepreneurship," *Journal of Business Research*, vol. 69, no. 11, pp. 5147–5152, 2016.
[80] C. M. V. D. Heijde and B. I. J. M. Van Der Heijden, "A competence-based and multidimensional operationalization and measurement of employability," *Human Resource Management*, vol. 45, no. 3, pp. 449–476, 2006.
[81] T. C. Sebora, S. M. Lee, and N. Sukasame, "Critical success factors for e-commerce entrepreneurship: an empirical study of Thailand," *Small Business Economics*, vol. 32, no. 3, pp. 303–316, 2009.
[82] A. Van Riel, J. Semeijn, and W. Janssen, "E-service quality expectations: a case study," *Total Quality Management & Business Excellence*, vol. 14, no. 4, pp. 437–450, 2003.
[83] S. Chorev and A. R. Anderson, "Success in Israeli high-tech start-ups; critical factors and process," *Technovation*, vol. 26, no. 2, pp. 162–174, 2006.
[84] A. K. Kohli and B. J. Jaworski, "Market orientation: the construct, research propositions, and managerial implications," *Journal of Marketing*, vol. 54, no. 2, pp. 1–18, 1990.
[85] S. F. Slater and J. C. Narver, "Market orientation and the learning organization," *Journal of Marketing*, vol. 59, no. 3, pp. 63–74, 1995.
[86] V. Kaushik, A. Khare, R. Boardman, and M. B. Cano, "Why do online retailers succeed? The identification and prioritization of success factors for Indian fashion retailers," *Electronic Commerce Research and Applications*, vol. 39, Article ID 100906, 2020.
[87] H. Gil-Gomez, V. Guerola-Navarro, R. Oltra-Badenes, and J. A. Lozano-Quilis, "Customer relationship management: digital transformation and sustainable business model innovation," *Economic Research-Ekonomska Istraživanja*, vol. 33, no. 1, pp. 2733–2750, 2020.
[88] C. Veloutsou and C. Ruiz Mafe, "Brands as relationship builders in the virtual world: a bibliometric analysis," *Electronic Commerce Research and Applications*, vol. 39, Article ID 100901, 2020.
[89] S. Wang, Y. Hong, N. Archer, and Y. Wang, "Modeling the success of small and medium sized online vendors in business to business electronic marketplaces in China," *Journal of Global Information Management*, vol. 19, no. 4, pp. 45–75, 2011.
[90] G. N. Chandler, B. Honig, and J. Wiklund, "Antecedents, moderators, and performance consequences of membership change in new venture teams," *Journal of Business Venturing*, vol. 20, no. 5, pp. 705–725, 2005.
[91] R. Wentrup, "The online–offline balance: internationalization for Swedish online service providers," *Journal of International Entrepreneurship*, vol. 14, pp. 562–594, 2016.
[92] Y. Zhao, Y. Zhao, X. Yuan, and R. Zhou, "How knowledge contributor characteristics and reputation affect user payment decision in paid Q&A? An empirical analysis from the perspective of trust theory," *Electronic Commerce Research and Applications*, vol. 31, pp. 1–11, 2018.
[93] C. Gieure and I. Buendía-Martínez, "Determinants of translation-firm survival: a fuzzy set analysis," *Journal of Business Research*, vol. 69, no. 11, pp. 5377–5382, 2016.
[94] Z. Hasnain, B. Yvonne, D. John, and E. Mahesh, "Straight from the horse's mouth: founders' perspectives on achieving "traction" in digital start-ups," *Computers in Human Behavior*, vol. 95, pp. 262–274, 2019.
[95] C. E. Helfat and M. B. Lieberman, "The birth of capabilities: market entry and the importance of pre-history," *Industrial and Corporate Change*, vol. 11, no. 4, pp. 725–760, 2002.
[96] K. Frey, C. Lüthje, and S. Haag, "Whom should firms attract to open innovation platforms? The role of knowledge diversity and motivation," *Long Range Planning*, vol. 44, no. 5-6, pp. 397–420, 2011.
[97] J. Mo, S. Sarkar, S. Sarkar, and S. Menon, "Know when to run: recommendations in crowdsourcing contests," *MIS Quarterly*, vol. 42, no. 3, pp. 919–944, 2018.
[98] X. Tian, M. Niu, W. Zhang, L. Li, and E. Herrera-Viedma, "A novel todim based on prospect theory to select green supplier with Q-rung orthopair fuzzy set," *Technological and Economic Development of Economy*, vol. 27, no. 2, pp. 284–310, 2020.
[99] M. Batey and A. Furnham, "The relationship between measures of creativity and schizotypy," *Personality and Individual Differences*, vol. 45, no. 8, pp. 816–821, 2008.
[100] G. J. Feist, "A meta-analysis of personality in scientific and artistic creativity," *Personality and Social Psychology Review*, vol. 2, no. 4, pp. 290–309, 1998.
[101] F. Buttle, *Customer Relationship Management: Concepts and Tools*, Elsevier Butterworth-Heinemann, Oxford, UK, 2004.
[102] C. Claycomb, C. Dröge, and R. Germain, "The effect of just-in-time with customers on organizational design and performance," *The International Journal of Logistics Management*, vol. 10, no. 1, pp. 37–58, 1999.

[103] S. Aggarwal, "Flexibility management: the ultimate strategy," *Industrial Management*, vol. 39, no. 1, pp. 5–14, 1997.

[104] L. Y. M. Sin, A. C. B. Tse, and F. H. K. Yim, "CRM: conceptualization and scale development," *European Journal of Marketing*, vol. 39, no. 11-12, pp. 1264–1290, 2005.

[105] S. E. S. Dijkstra, *Upwork at Work: Labor as A Service: A Transformation of Labor in the Platform Society*, Utrecht University, Utrecht, Netherlands, 2017.

[106] K. L. Guth and D. C. Brabham, "Finding the diamond in the rough: exploring communication and platform in crowdsourcing performance," *Communication Monographs*, vol. 84, no. 4, pp. 510–533, 2017.

[107] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Champaign, IL, USA, 1949.

[108] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: issues and directions," *IEEE Internet Computing*, vol. 17, no. 2, pp. 76–81, 2013.

[109] C. Stasz and D. J. Brewer, *Academic Skills at Work: Two Perspectives*, 1999.

[110] N. Luz, N. Silva, and P. Novais, "A survey of task-oriented crowdsourcing," *Artificial Intelligence Review*, vol. 44, no. 2, pp. 187–213, 2015.

[111] S. Basu Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in knowledge-intensive crowdsourcing," *The VLDB Journal*, vol. 24, no. 4, pp. 467–491, 2015.

[112] L. B. Erickson and E. M. Trauth, "Getting work done: evaluating the potential of crowdsourcing as a model for business process outsourcing service delivery," in *Proceedings of the 2013 Annual Conference on Computers and People Research*, pp. 135–140, Cincinnati, OH, USA, May 2013.

Hindawi
Complexity
Volume 2021, Article ID 9865171, 1 page
https://doi.org/10.1155/2021/9865171

Corrigendum

Corrigendum to "Research on Credit Card Default Prediction Based on *k*-Means SMOTE and BP Neural Network"

Ying Chen and Ruirui Zhang

School of Business, Sichuan Agricultural University, Chengdu 611830, China

Correspondence should be addressed to Ruirui Zhang; zhangruiruisw@163.com

Received 31 May 2021; Accepted 31 May 2021; Published 14 June 2021

In the article titled "Research on Credit Card Default Prediction Based on *k*-Means SMOTE and BP Neural Network" [1], the authors would like to clarify that they employed the python package, kmeans-smote 0.1.2, in this study [2]. The error is that a citation to the related article was not included in the original publication, and the following text in Section 3 should be replaced with the addition of the missing references, 22 and 23 [2, 3]:

"Therefore, according to the problem of imbalance of credit card sample categories, this paper uses an improved smote algorithm called *k*-means SMOTE algorithm" should be replaced with "Therefore, according to the problem of imbalance of credit card sample categories, this paper uses an improved smote algorithm called *k*-means SMOTE algorithm [22, 23]."

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Chen and R. Zhang, "Research on Credit Card Default Prediction Based on *k*-Means SMOTE and BP Neural Network," *Complexity*, vol. 2021, Article ID 6618841, 13 pages, 2021.

[2] G. Douzas, F. Bacao, and F. Last, "Oversampling for imbalanced learning based on *k*-means and SMOTE," 2018, https://arxiv.org/abs/1711.00837.

[3] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on *k*-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, 2018.

Hindawi
Complexity
Volume 2021, Article ID 6641298, 12 pages
https://doi.org/10.1155/2021/6641298

Research Article

Advantages of Combining Factorization Machine with Elman Neural Network for Volatility Forecasting of Stock Market

Fang Wang,[1,2] **Sai Tang**,[3] **and Menggang Li**[2,4,5]

[1]*School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China*
[2]*Beijing Laboratory of National Economic Security Early-Warning Engineering, Beijing Jiaotong University, Beijing 100044, China*
[3]*School of Humanities, Social Sciences & Law, Harbin Institute of Technology, Harbin, China*
[4]*National Academy of Economic Security, Beijing Jiaotong University, Beijing 100044, China*
[5]*Beijing Center for Industrial Security and Development Research, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Menggang Li; mgli1@bjtu.edu.cn

Received 29 December 2020; Accepted 11 May 2021; Published 24 May 2021

Academic Editor: Thiago Christiano Silva

With a focus in the financial market, stock market dynamics forecasting has received much attention. Predicting stock market fluctuations is usually challenging due to the nonlinear and nonstationary time series of stock prices. The Elman recurrent network is renowned for its capability of dealing with dynamic information, which has made it a successful application to predicting. We developed a hybrid approach which combined Elman recurrent network with factorization machine (FM) technique, i.e., the FM-Elman neural network, to predict stock market volatility. In this paper, the Standard & Poor's 500 Composite Stock Price (S&P 500) index, the Dow Jones industrial average (DJIA) index, the Shanghai Stock Exchange Composite (SSEC) index, and the Shenzhen Securities Component Index (SZI) were used to demonstrate the validity of our proposed FM-Elman model in time-series prediction. The results were compared with predictions obtained from the other two models which are basic BP neural network and the Elman neural network. Some experiments showed that the FM-Elman model outperforms others through different accuracy measures. Furthermore, the effects of volatility degree on prediction performance from different stock indexes were investigated. An interesting phenomenon had been found through some numerical experiments on the effects of different user-specified dimensions on the proposed FM-Elman neural network.

1. Introduction

In recent years, the fluctuation analysis of financial time series has received a lot of concerns. Stock market volatility prediction has become a significant topic in economic research. The study of stock market volatility forecasting can be helpful for policy makers to take appropriate decisions on asset allocation and risk management. Therefore, predicting the volatility of financial time series with a reasonable accuracy deserves much attention. However, stock market exhibits nonlinear and chaotic properties in nature [1, 2]. Statistical models then have some difficulties in dealing with nonlinear and nonstationary time series or deriving satisfactory forecasting performance under the statistical assumptions of normally distributed observations. The predicting becomes more challenging.

Artificial neural network has the advantages on learning from sample data and capturing the nonlinear relations among interconnected neurons through training mode [3]. It is capable of dealing with nonlinear high-dimensional data and approximating any nonlinear functions with arbitrary precision [4–7]. Particularly, the simple recurrent network, i.e., Elman neural network (Elman NN) [8] has shown its stronger ability as it has the characteristic of time-varying. And the Elman NN is a kind of feedback network where the added layer connecting to the hidden layer can be regarded as a time delay operator capable of memorizing recent events. It is a time-varying

predictive control system that has faster convergence and more accurate mapping ability.

Elman NN has been utilized to financial prediction and applied to many other different types of time series. Most studies on Elman NN obtained higher accuracy. Zheng [9] used an Elman NN to forecast opening prices of the Shanghai Stock Exchange. Wu and Duan applied the Elman NN in predicting stock [10] and gold future markets [11], respectively. In the area of electricity prediction, Rani and Victoire [12] integrated the decomposition method and group search optimization algorithm into the Elman NN. It showed that the Elman NN outperformed other approaches.

There are also other artificial neural networks like wavelet neural network and radial basis function neural network [13–16]. Some developed artificial intelligence techniques like expert systems [17, 18], support vector machines (SVMs) [19, 20], and hybrid methods [21, 22] are also applied in forecasting stock prices. Recently, some novel models have utilized random jump or random time effective function with different neural networks [23, 24] which have been proposed in forecasting financial market.

Although the models which are based on artificial intelligent have achieved remarkable results, there are still limitations. There is a few technique in most models which pay attention to the nonlinear interactions among the inputs. For example, the nonlinearities in neural network models were handled by the activation functions. These models without consideration of interactions between features with different scales have been widely used in some applications such as image processing, mechanical translation, and speech recognition [25–27].

FM is originally used for collaborative recommendations which were first introduced by Rendle [28]. FM is a supervised learning method that can model feature interactions with second-order even when the data have very high sparsity and high dimension. FMs show state-of-the-art performance as they have two main benefits. First, FMs are on a par with polynomial regression but can achieve empirical accuracy with smaller and faster evaluation results. Second, unlike the linear regression, FMs can infer the weights of feature interactions that were not observed in the training dataset. The weights of second-order feature interactions have the low-rank property which makes FMs become increasingly popular in the recommender system. Although FM is a general framework of matrix factorization, FM shows more flexibility as the matrix factorization method only models the relation between two entities [29]. FMs are general predictors like SVMs and have a lot of applications in industry. FMs are applicable to any variables with real feature and are not restricted to recommender systems. FM gives a promising direction for the prediction purpose in regression, classification, and ranking [30–33].

As far as we know, real-world time series is rarely pure nontime-varying. And the linear regression is not always capable of deriving the interactions between features which however are more common in various applications. Hence, the problem of dealing with time-varying and nonlinear interactions can be solved by combining FM with Elman NN. Moreover, it is almost universally agreed in the forecasting literature that no single model is the best in every situation because a real-world problem is often complex. Using any single model may not be able to capture different patterns equally well [34]. Therefore, we propose a forecasting model combining FM technique with Elman NN for stock market volatility prediction in the present paper.

In this paper, we apply the FM-Elman neural network to forecast the volatility degree's behavior of the Standard & Poor's 500 Composite Stock Price (S&P 500) index, the Dow Jones industrial average (DJIA) index, the Shanghai Stock Exchange Composite (SSEC) index, and the Shenzhen Securities Component index (SZI) from January 2nd, 2000, to December 31st, 2011. Different threshold values were introduced into our model, and the corresponding volatility prediction results were presented. To show the advantages of the proposed FM-Elman model, we compare the predicting results with two other neural network models including BP network and Elman recurrent network through three performance evaluation measures such as the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

The remainder of this paper is presented as following sections. In Section 2, the Elman NN and FM are reviewed where they are prepared for our proposed model. Then, we give the prediction model FM-Elman neural network in Section 3. In this section, we first give the model description and in the same time introduce some needed ingredients of it. And the algorithm of the FM-Elman model is also given. Section 4 presents the main forecasting results of the FM-Elman model. This section gives predicting comparisons among our proposed model, BP neural network, and Elman neural network. It not only presents the effects of different parameters like volatility degree and user-specified dimension on the FM-Elman model's performance but also considers other evaluation measures. And Section 5 highlights some necessary conclusions finally.

2. Elman Neural Network and Factorization Machine

2.1. Elman Neural Network (Elman NN). Elman neural network was founded by Elman [8] in 1990 which is famous for its recurrent topology structure. Unlike the BP network, an Elman NN has a set of recurrent nodes. The so-called recurrent nodes in the buffer received message from the peered output nodes in the hidden layer and then transmitted message to the hidden layer. Every hidden node is connected to only one recurrent neuron, and the message will remain the same after transmitting. Hence, the number of recurrent layer nodes is the same as the number of hidden nodes, and the recurrent layer contains the state of input data from the hidden layer.

Figure 1 gives the structure of multi-input Elman NN. The Elman NN is composed of the input layer, the hidden layer, the output layer, and the recurrent layer. There are n nodes in the input layer, and both the hidden layer and the recurrent layer have m nodes. In the output layer, there exists only one unit neuron. The mathematical computation for the nonlinear state of the Elman NN is

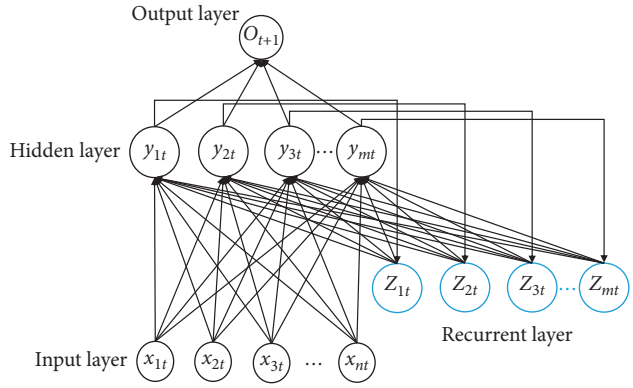


Figure 1: Topology of the Elman neural network.

$$\begin{cases} \overrightarrow{u}(t) = f\left(W^A \overrightarrow{x}(t-1) + W^B \overrightarrow{z}(t)\right), \\ \overrightarrow{z}(t) = \overrightarrow{u}(t-1), \\ y(t) = g\left(W^C \overrightarrow{u}(t)\right), \end{cases} \tag{1}$$

where $\overrightarrow{u}(t)$ is the vector of output values in the hidden layer, $y(t)$ is the final output of the network, and $\overrightarrow{x}(t-1) = (x_{1,t-1}, x_{2,t-1}, \ldots, x_{n,t-1})$ denotes the input of the network at time $t-1$. The weight matrix W^A connects the input layer node to the node in the hidden layer, W^B connects the node in the recurrent layer to the hidden layer neuron, and W^C is the matrix which connects the node in the hidden layer to the output node. Functions $f(\cdot)$ and $g(\cdot)$ are the activation functions where $f(x)$ is the sigmoid function and $g(x)$ is an identity function in this paper.

From equation (1) and through deduction, we can obtain that

$$\overrightarrow{z}(t) = f\left(W^A_{t-1} \overrightarrow{x}(t-2) + W^B_{t-1} \overrightarrow{z}(t-1)\right), \tag{2}$$

where $\overrightarrow{z}(t)$ depends on the matrix W^A_{t-1} and W^B_{t-1} which comes from different time. Elman NN has the ability to adapt to time series varying.

2.2. Factorization Machine. FM has the same prediction ability as SVMs but also has capability of estimating reliable parameters under very sparse data. The feature of modelling all variable interactions is comparable to a polynomial kernel in SVM. The equation for a FM with second-order feature is defined as follows:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j, \tag{3}$$

where the parameters $w_0 \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^n$, and $\mathbf{V} \in \mathbb{R}^{n \times k}$ have to be determined. And $\langle \cdot, \cdot \rangle$ is the inner product of two vectors with size k. Then,

$$\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}, \tag{4}$$

which models the interaction between the ith variable and the jth variable, where v_i is the ith variable with $k (\in \mathbb{N}_0^+)$ dimension factors.

Our intuition for the complexity of equation (3) is in $O(kn^2)$ because all pairwise interactions have to be computed. As there is no parameter in a model depending on two variables directly, the pairwise interactions in equation (3) are reformulated as follows:

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j = \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{if} x_i \right)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2 \right). \tag{5}$$

And the equation only needs linear runtime $O(kn)$ to be computed after the reformulation. So, FMs are applicable from a computational point of view.

3. Our Proposed Method

3.1. Modelling. We construct the Elman recurrent neural network with factorization machine, i.e., FM-Elman neural network, to predict the volatility of different stock indexes. The detailed topology of FM-Elman neural network is presented in Figure 2.

The layers of the FM-Elman neural network are analyzed as follows:

(1) *Hidden Layer*. The nodes in the hidden layer are partitioned into two parts. One part of them has normal nodes which show the linear relations among the input data. And the remaining nodes in the other part incorporate all interactions between each pair of features from the input data. The results are computed by

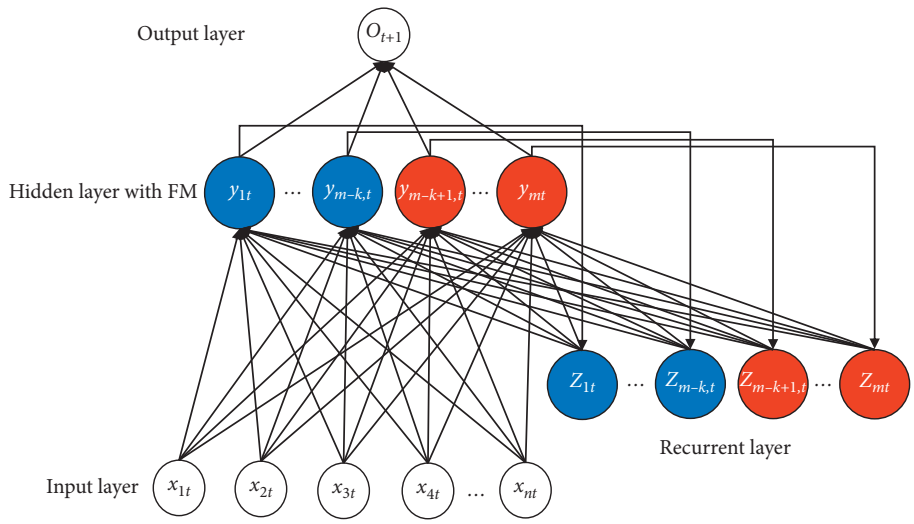


Figure 2: Structure of the Elman neural network with factorization machine.

$$y_j(t) = f\left(\sum_{i=1}^{n} v_{ij}x_i(t) + \sum_{z=1}^{m} u_{zj}x_z(t)\right), \quad i = 1,2,\ldots,n, j = 1,2,\ldots,m-k, z = 1,2,\ldots,m,$$

$$y_{m-k+j}(t) = f\left(\begin{array}{l} \frac{1}{2}\left(\left(\sum_{i=1}^{n} \hat{v}_{i,m-k+j}x_i(t)\right)^2 - \sum_{i=1}^{n}\left(\hat{v}_{i,m-k+j}\right)^2 x_i^2(t)\right)+ \\ \\ \frac{1}{2}\left(\left(\sum_{i=1}^{n} \hat{u}_{z,m-k+j}x_z(t)\right)^2 - \sum_{i=1}^{n}\left(\hat{u}_{z,m-k+j}\right)^2 x_z^2(t)\right) \end{array}\right), \quad i = 1,2,\ldots,n, j = 1,2,\ldots,k, z = 1,2,\ldots,m, \tag{6}$$

where x_i is the input value from input node i, y_j denotes the value of the jth node in the hidden layer, v_{ij} is the undetermined weight which relates the ith input node to the jth normal node in the hidden layer, $\hat{v}_{ij}$ is the weight connecting the ith input node to the remaining node in the hidden layer which is also undetermined, u_{zj} and $\hat{u}_{zj}$ have the same meaning with v_{ij} and $\hat{v}_{ij}$ except for the first two parameters which link the nodes in the recurrent layer to the hidden layer, t is the iteration number in the formulas, k is the user-specified dimension, and f is the activation function.

(2) *Recurrent Layer.* The number of nodes in the recurrent layer is the same as the number of hidden nodes. Each hidden node is connected to only one node in the recurrent layer, and the connected weight is a constant value one. So, the recurrent layer is also partitioned into two parts which are presented as follows:

$$\begin{aligned} x_z(t) &= y_j(t-1), \quad j = 1,2,\ldots,m-k, \\ &\quad z = 1,2,\ldots,m-k, \\ x_{m-k+z}(t) &= y_{m-k+j}(t-1), \quad j = 1,2,\ldots,k, \\ &\quad z = 1,2,\ldots,k. \end{aligned} \tag{7}$$

(3) *Output Layer.* The outputs are

$$O_j(t) = g\left(\sum_{i=1}^{m} w_{ij}y_i(t)\right), \quad j = 1,2,\ldots,p, \tag{8}$$

where w_{ij} is the undermined weight, O_j is the output value of jth node in the output layer, and g is the activation function.

There are different loss functions to calculate the error between the actual and the estimated values from the output layer. We consider the squared loss function which is given as follows:

$$E(t) = \frac{1}{2}(T(t) - O(t))^2, \tag{9}$$

where $T(t)$ is the actual value in the tth iteration.

So, the final output error of the FM-Elman model is computed by

$$l = \frac{1}{N}\sum_{t=1}^{N} E(t) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}(T(t) - O(t))^2. \tag{10}$$

To optimize the FM-Elman model, we often use the stochastic gradient decent method to update the weights until it achieves convergence.

3.2. Algorithm of FM-Elman Model. The training process of the FM-Elman model is detailed as follows:

(1) The gradients of the weights and the updated rule in the output layer are computed as follows:

$$\begin{aligned}\Delta w_j &= -\eta\frac{\partial l}{\partial w_j} = \eta(T(t)-O(t))y_j(t),\\ w_j &\leftarrow w_j - \eta\frac{\partial l}{\partial w_j},\end{aligned} \tag{11}$$

where $j = 1, 2, \ldots, m$, η is the learning rate.

(2) The gradients of the weights and the updated rule in the hidden layer connected by the recurrent layer are calculated as the following two cases:

$$\begin{aligned}\Delta u_{zj} &= -\eta\frac{\partial l}{\partial u_{zj}} = \eta(T(t)-O(t))g'w_j f' x_z(t), \quad j = 1, 2, \ldots, m-k,\\ \Delta\widehat{u}_{z,m-k+j} &= -\eta\frac{\partial l}{\partial \widehat{u}_{z,m-k+j}} = \eta(T(t)-O(t))g'w_{m-k+j}f'\left(\left(\sum_{z=1}^{m}\widehat{u}_{z,m-k+j}x_z(t)\right)x_z(t) - \sum_{z=1}^{m}\widehat{u}_{z,m-k+j}x_z^2(t)\right), \quad j = 1, 2, \ldots, k,\\ u_{zj} &\leftarrow u_{zj} - \eta\frac{\partial l}{\partial u_{zj}}, \quad j = 1, 2, \ldots, m-k,\\ \widehat{u}_{z,m-k+j} &\leftarrow \widehat{u}_{z,m-k+j} - \eta\frac{\partial l}{\partial \widehat{u}_{z,m-k+j}}, \quad j = 1, 2, \ldots, k,\end{aligned} \tag{12}$$

where $z = 1, 2, \ldots, m$, η is the learning rate, k is the user-specified dimension, and g' and f' are corresponded derivative functions of g and f.

(3) The gradients of the weights and the updated rule in the hidden layer connected by the input layer are computed as the following two cases:

$$\begin{aligned}\Delta v_{ij} &= -\eta\frac{\partial l}{\partial v_{ij}} = \eta(T(t)-O(t))g'w_j f' x_i(t), \quad j = 1, 2, \ldots, m-k,\\ \Delta\widehat{v}_{i,m-k+j} &= -\eta\frac{\partial l}{\partial \widehat{v}_{i,m-k+j}} = \eta(T(t)-O(t))g'w_{m-k+j}f'\left(\left(\sum_{i=1}^{m}\widehat{v}_{i,m-k+j}x_i(t)\right)x_z(t) - \sum_{i=1}^{m}\widehat{v}_{i,m-k+j}x_i^2(t)\right), \quad j = 1, 2, \ldots, k,\\ v_{ij} &\leftarrow v_{ij} - \eta\frac{\partial l}{\partial v_{ij}}, \quad j = 1, 2, \ldots, m-k,\\ \widehat{v}_{i,m-k+j} &\leftarrow \widehat{v}_{i,m-k+j} - \eta\frac{\partial l}{\partial \widehat{v}_{i,m-k+j}}, \quad j = 1, 2, \ldots, k,\end{aligned} \tag{13}$$

where $i = 1, 2, \ldots, n$, η is the learning rate, k is the user-specified dimension, and g' and f' are corresponded derivative functions of g and f.

4. Forecasting Results

4.1. Data Selecting and Processing. Stock prices' different changing behaviors and volatility predicting study have long been a focus in economic research. We use the logarithmic return to describe the statistical characteristic of a stock return volatility. The stock logarithmic return is defined as

$$r(t) = \ln S(t) - \ln S(t-1), \tag{14}$$

where $S(t)$ denotes the stock daily closing price at time t.

The data (http://www.finance.yahoo.com) chosen for our experiment are the Standard & Poor's 500 Composite Stock Price (S&P 500) index, the Dow Jones industrial average (DJIA) index, the Shanghai Stock Exchange Composite (SSEC) index, and the Shenzhen Securities Component index (SZI). All the data are collected from trading days ranging from January 2nd, 2000, to December 31st, 2011. So, the size of different time series is 3019, 3027, 2901, and 2902, respectively. We partition them into training sets (from January 2nd, 2000, to December 31st, 2007) and testing sets (from January 2nd, 2008, to December 31st, 2011). Table 1 describes the data's statistical feature, where N_{all} and N_{test} denote the sizes of the whole data and the testing samples, respectively.

In this paper, a threshold value $\theta (\geq 0)$ is introduced as the volatility degree. Let $\Re(\theta)$ denote the set in which the stock returns' absolute values are greater than the valueθ, and the definition is given by $\Re(\theta) = \{r(t) \| r(t)| \geq \theta, t = 1, 2, \ldots, T\}$. Once the value θ is set, we can obtain the dataset including the satisfied stock daily closing price. Figure 3 gives an example of stock returns with different thresholds. For a fixed threshold value θ, the corresponding stock trading dates are determined. The trading dates are in the set where time t values satisfy $|r(t)| \geq \theta$, $t = 1, 2, \ldots, T$. Newly formed series are arranged in a chronological order. Table 2 gives the numbers of data for stock indexes distributed in training data and testing data under different threshold values θ.

When the threshold value θ equals 0, the averaged values of daily absolute returns for S&P 500, DJIA, SSEC, and SZI are 0.0094, 0.0089, 0.0117, and 0.0130, respectively. We set different volatility degrees 0.003, 0.006, 0.009, and 0.012 to see the data numbers distributed in the training and testing data set, respectively. In Table 2, as the threshold value θ increases, the quantity of data in both training dataset and testing dataset that exceeds the given threshold value gradually reduces. And it can be predicted that when θ is larger than 0.012, the corresponding numbers will be fewer.

Four input variables including the daily opening prices, the daily highest prices, the daily lowest prices, and the daily closing prices are selected according to the newly formed dates. And we choose the next time daily closing price in the chronological ordered datasets as the output variable. In order to reduce the noise's impact on the stock markets, all the input data X are normalized as follows:

Table 1: The data and their descriptive statistical analysis.

Name	N_{all}	N_{test}	Mean	Std.
S&P 500	3019	1009	1190.02	184.26
DJIA	3027	1009	10612.79	1397.02
SSEC	2901	976	2226.36	973.46
SZI	2902	976	7028.88	4389.84

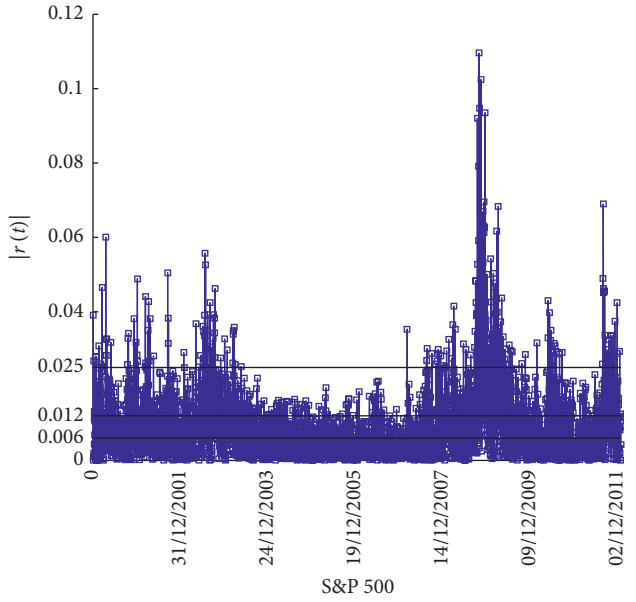


Figure 3: Absolute returns of S&P 500 with different thresholds.

$$X' = \frac{X - \min X}{\max X - \min X}. \tag{15}$$

Then, it is easily to obtain the actual prediction value through $X = X'(\max X - \min X) + \min X$.

4.2. Performances of FM-Elman Model. In the proposed FM-Elman neural network, we choose the structure with $4 \times 10 \times 1$ where the number of input nodes is 4, the number of the hidden nodes is 10, and the number of the output nodes is 1. We set the maximum iterations number as 5000, $\eta = 0.02$, and the predefined minimum training threshold is $\varepsilon = 5 \times 10^{-5}$.

To analyze and evaluate the predicting performance of the FM-Elman neural network model, we use the accuracy measures with the corresponding definitions as follows:

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{t=1}^{N}|T(t) - O(t)|, \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{t=1}^{N}(T(t) - O(t))^2}, \\ \text{MAPE} &= 100 \times \frac{1}{N}\sum_{t=1}^{N}\left|\frac{T(t) - O(t)}{T(t)}\right|, \end{aligned} \tag{16}$$

Table 2: Numbers of data distributed in training data and testing data under different threshold values θ.

Threshold θ	S&P 500		DJIA		SSEC		SZI	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
0	2010	1009	2018	1009	1925	976	1926	976
0.003	1415	777	1426	756	1485	785	1520	824
0.006	987	585	945	544	1105	643	1190	688
0.009	663	455	622	431	817	520	923	562
0.012	447	350	418	327	615	411	688	462

where $T(t)$ and $O(t)$ are the actual value and the predictive value in the tth iteration and N is the sample size. When the values of these evaluation measures are smaller, the prediction performance is better.

In this section, we derive the stock prices' different fluctuation behaviors through the proposed FM-Elman neural network. First, the new proposed model is proved to be better compared with BPNN and Elman neural model through different evaluation measures. Then, the prediction performance of FM-Elman neural network is measured when threshold value θ varies. And finally, we can see how the user-specified dimension k impacts the performance of the proposed model.

When the threshold value equals 0, the training datasets and testing datasets are the original datasets. And the prediction results of S&P500 and SSEC by the FM-Elman neural model with $k = 3$ are presented in Figures 4 and 5.

We then give the performance comparisons among different prediction models for $\theta = 0$ in Table 3 where the MAPE (100) means the latest 100 days of MAPE in the testing data. Three different prediction models include BPNN, Elman neural network, and FM-Elman neural network with the user-specified dimension $k = 3$. Table 3 shows that FM-Elman model' evaluation errors are all smaller than those in the other two models. In addition, the MAPE (100) value is smaller than the corresponding stock index' MAPE value. It shows that the short-term prediction outperforms the long-term prediction.

4.3. The Impact of θ. When θ varies from 0.003 to 0.012, different prediction analysis of indexes S&P 500, DJIA, SSEC, and SZI can be performed by the FM-Elman neural network. Figures 6 and 7 are the prediction analysis of S&P 500 and SSEC by the FM-Elman neural model. The two figures also show the effectiveness of forecasting with different volatility degree values of θ. When θ is small, the performance of volatility prediction is revealed better through the empirical results. Like $\theta = 0.03$ and $\theta = 0.06$ in Figures 6(a) and 6(b), the predictive values are closer to the actual values than those in Figures 6(c) and 6(d). Figure 7 also indicates the similar results.

We choose the often recommended criterion MAPE to measure the prediction performance for stock indexes S&P 500, DJIA, SSEC, and SZI under the FM-Elman neural model which is presented in Table 4. When the value of θ increases, the value of MAPE increases gradually. And the numerical experiment results show that using the FM-Elman neural network model, the volatility degree forecasting is feasible.

4.4. The Impact of k. In this subsection, we want to analyze how the user-specified dimension k affects the prediction performance by the FM-Elman neural model through numerical experiment when $\theta = 0$. From the descriptions in Section 3, when k increases, the amount of red nodes in both hidden layer and recurrent layer becomes larger. That means more hidden and recurrent nodes in the proposed FM-Elman neural network will contain interaction information from the connected inputs. Then, the computation becomes complicated as k increases. It is interesting to see in Table 5 that the evaluation values of MAE, MAPE, and RMSE for indexes S&P 500, DJIA, SSEC, and SZI increase first and then decrease with the increasing of k. So, the low user-specified dimension or high user-specified dimension is a better choice. No matter which outperforms the predicting results of BPNN and Elman NN from the previous Table 3.

4.5. Further Predicting Performance Evaluation. We adopt three trend-type statistical methods, i.e., directional symmetry (DS), correct up-trend (CP), and correct down-trend (CD) [35], to check the practical stock movement. When the values of these three performance evaluation results become larger, the forecasting of change direction will be more precise. The definitions of these three performance evaluation methods are given as

$$\mathrm{DS} = \frac{100}{N_1}\sum_{t=1}^{N_1} a_t, \quad a_t = \begin{cases} 1, & \text{if } (T_t - T_{t-1})(O_t - O_{t-1}) \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where N_1 is the number of testing samples.

$$\mathrm{CP} = \frac{100}{N_2}\sum_{t=1}^{N_2} a_t, \quad a_t = \begin{cases} 1, & \text{if } T_t - T_{t-1} > 0 \text{ and } O_t - O_{t-1} \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where N_2 is the number of testing samples which satisfy $T_t - T_{t-1} > 0$.

$$\mathrm{CD} = \frac{100}{N_3}\sum_{t=1}^{N_3} a_t, \quad a_t = \begin{cases} 1, & \text{if } T_t - T_{t-1} < 0 \text{ and } O_t - O_{t-1} \leq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

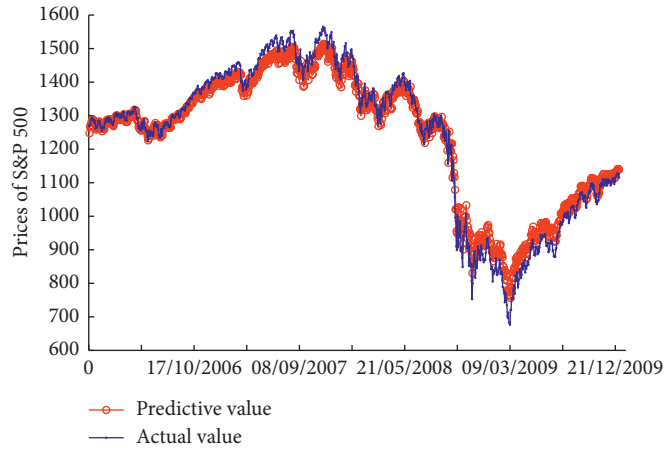


Figure 4: Comparisons of prediction results and the daily closing prices of S&P 500 for testing datasets.

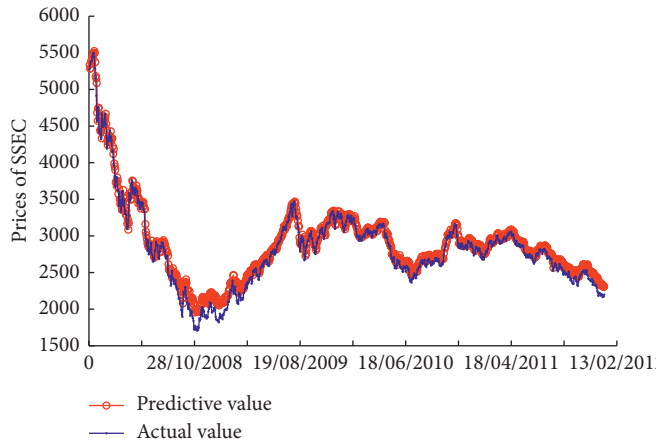


Figure 5: Comparisons of prediction results and the daily closing prices of SSEC for testing datasets.

Table 3: Comparisons of prediction performances among different models for $\theta = 0$.

Evaluation errors	S&P 500			DJIA		
	BPNN	Elman NN	FM-Elman	BPNN	Elman NN	FM-Elman
MAE	54.1822	46.8432	43.5088	321.3915	294.4472	280.9011
MAPE	5.1566	4.5231	4.1692	3.3196	3.0473	2.9379
RMSE	60.2121	53.8414	49.7034	382.7756	354.6576	348.8277
MAPE (100)	1.7813	1.5526	1.5273	1.1938	1.1733	1.1103
Evaluation errors	SSEC			SZI		
	BPNN	Elman NN	FM-Elman	BPNN	Elman NN	FM-Elman
MAE	68.5672	53.6271	50.4433	287.2003	255.4868	228.2216
MAPE	2.4598	1.9369	1.8008	2.6363	2.2933	2.0552
RMSE	85.4527	72.682	69.7871	355.0629	324.7933	298.6005
MAPE (100)	2.8151	2.5933	2.5618	3.0594	2.8707	2.7645

where N_3 is the number of testing samples which satisfy $T_t - T_{t-1} < 0$. T_t and O_t are the actual value and the predictive value in the tth iteration, respectively.

In Table 6, the trend-type measures DS, CP, and CD for stock indexes S&P 500, DJIA, SSEC, and SZI under varying volatility degrees are presented through some numerical experiments. When the value of θ changes, all the stock indexes change a little. And we can see that the direction forecasting results of SSEC and SZI show better performance than S&P 500 and DJIA since the DS, CP, and CD values in two indexes before all exceed 50. And the performance results of stock indexes S&P 500 and DJIA are more sensitive

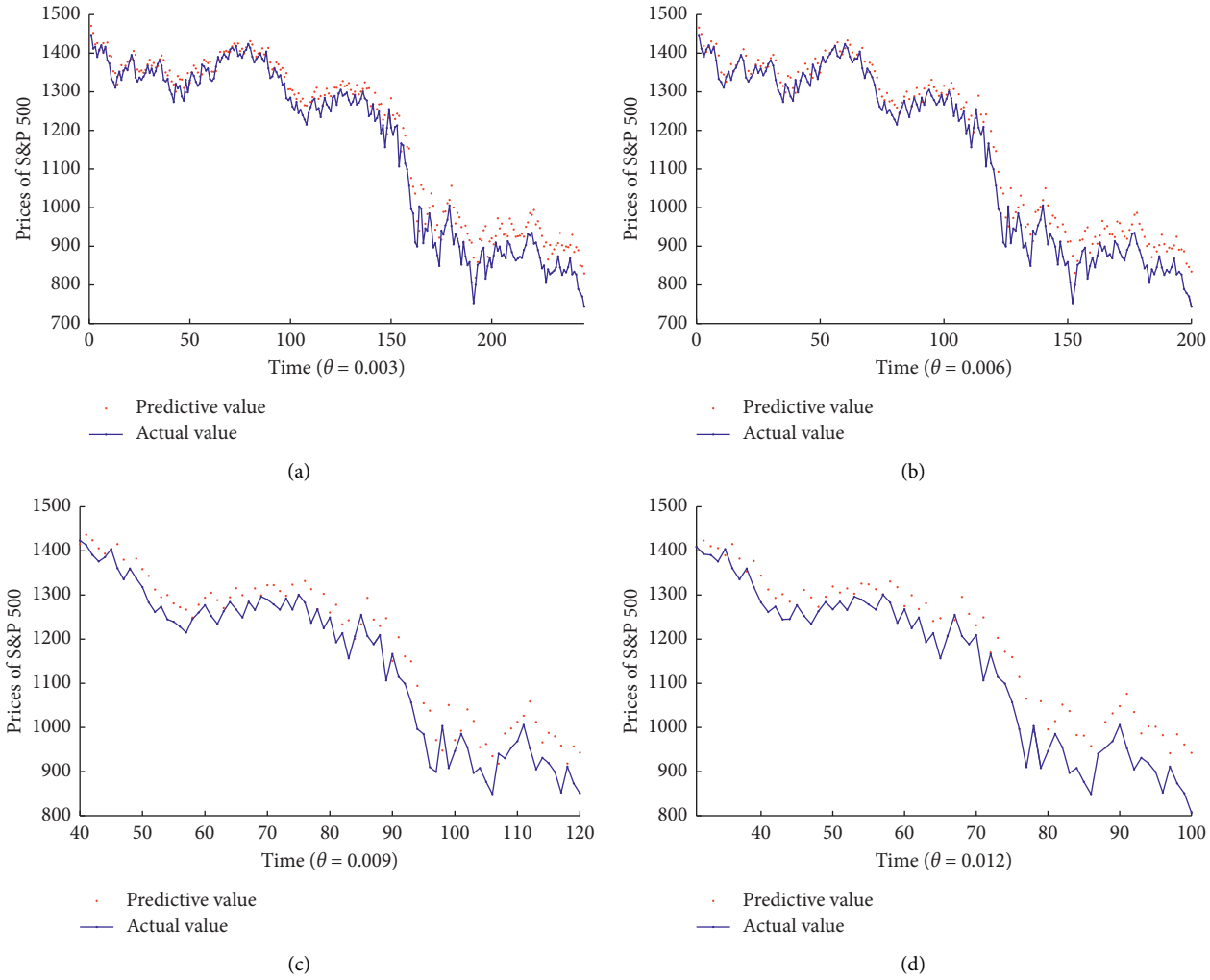


Figure 6: Volatility degree prediction of S&P 500 with different thresholds θ.

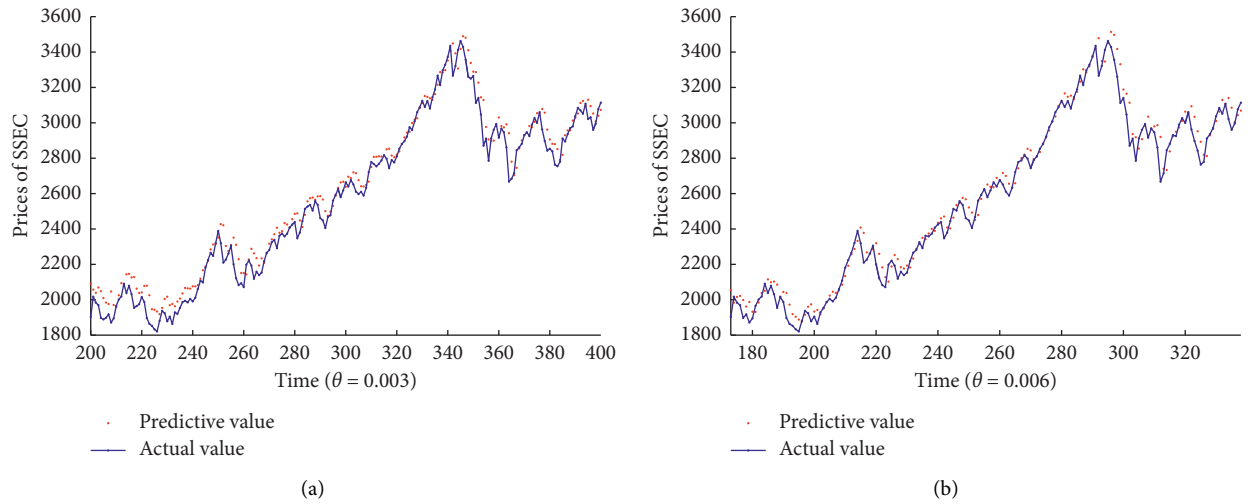


Figure 7: Continued.

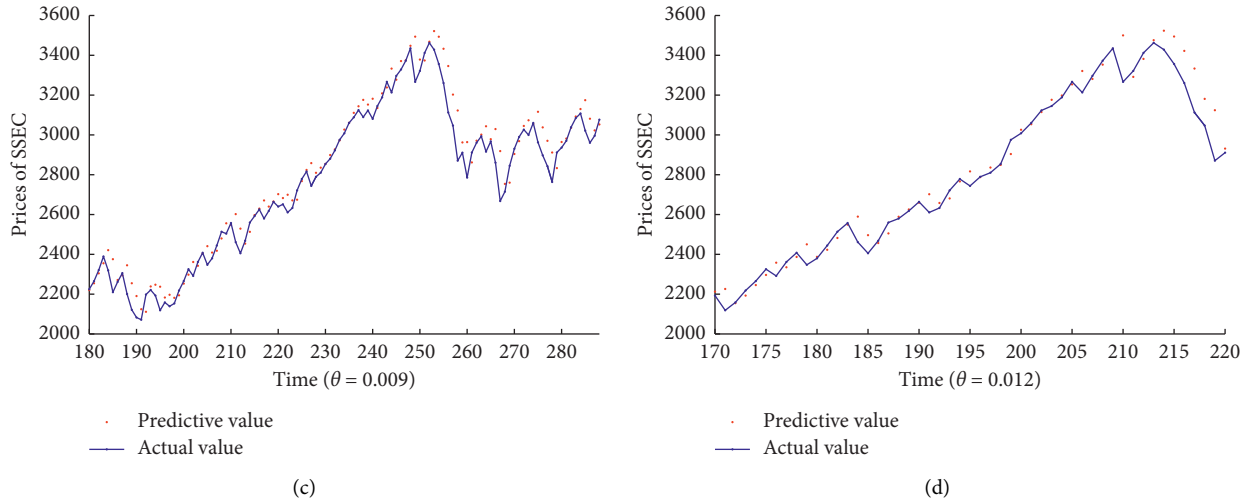


Figure 7: Volatility degree prediction of SSEC with different thresholds θ.

Table 4: Forecasting errors (MAPE) of the FM-Elman (k = 3) model as threshold value θ varies.

MAPE	S&P 500	DJIA	SSEC	SZI
θ = 0.003	3.6657	2.9006	2.3593	2.5499
θ = 0.006	4.4312	3.2554	2.3608	2.7811
θ = 0.009	5.3726	3.2998	2.7977	3.1299
θ = 0.012	6.3368	3.7571	2.9498	3.1579

Table 5: Forecasting errors of the FM-Elman model when θ = 0.

Dimension	S&P 500			DJIA		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
k = 1	40.3236	3.8834	46.6309	263.665	2.7759	335.7891
k = 3	43.5088	4.1692	49.7034	280.9011	2.9379	348.8277
k = 6	39.5023	3.8293	46.4366	254.584	2.6649	321.9889
k = 9	39.041	3.7597	45.4894	227.0823	2.3703	290.28
Dimension	SSEC			SZI		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
k = 1	47.0132	1.6545	68.2408	222.8488	2.032	293.7208
k = 3	50.4433	1.8008	69.7871	228.2216	2.0552	298.6005
k = 6	50.8593	1.8233	70.5519	241.7771	2.1825	317.8106
k = 9	48.4128	1.7097	68.4333	240.4362	2.1684	316.2495

Table 6: Trend-type forecasting evaluation of the FM-Elman (k = 3) model as threshold value θ varies.

Evaluation errors	S&P 500			DJIA		
	DS	CP	CD	DS	CP	CD
θ = 0	48.662	52.7473	43.8445	49.1576	52.3985	45.3961
θ = 0.003	48.5199	52.1327	44.2254	49.2063	52.451	45.4023
θ = 0.006	51.1111	52.9412	49.1039	47.6103	48.9362	46.1832
θ = 0.009	49.011	50	47.9821	47.0998	47.7273	46.4455
θ = 0.012	43.7143	43.9306	43.5028	43.1193	44.6429	41.5094
Evaluation errors	SSEC			SZI		
	DS	CP	CD	DS	CP	CD
θ = 0	50	51.8987	50.9221	51.1879	53.2164	52.2541

Table 6: Continued.

Evaluation errors	S&P 500			DJIA		
	DS	CP	CD	DS	CP	CD
$\theta = 0.003$	50.4459	50.646	50.2513	51.335	51.7677	50.9346
$\theta = 0.006$	51.0109	50.641	51.3595	52.1802	52.439	51.9444
$\theta = 0.009$	52.1154	52.5097	51.7241	52.847	53.7037	52.0548
$\theta = 0.012$	51.3382	50.2538	52.3364	50.211	53.7778	51.9481

to θ with large value. For example, when θ changes from the value 0.009 to 0.012, the values of DS, CP, and CD for stock indexes S&P 500 and DJIA decrease sharply.

5. Conclusion

In this study, we developed an improved Elman recurrent neural network by introducing the factorization machine. Through extensive numerical experiments on the data from stock indexes S&P 500, DJIA, SSEC, and SZI, we demonstrated the effectiveness of the FM-Elman neural network. The prediction accuracy for all financial time series shows that our proposed FM-Elman model outperforms the BP neural network and the original Elman NN. We select training and testing datasets under different volatility degrees, i.e., the threshold value θ varies, to predict. The prediction performance of the FM-Elman model will degrade as θ becomes larger. We also investigate the effect of the user-specified dimension on the prediction performance by the FM-Elman neural model.

The contribution of this work includes the following two points: (1) a technique FM combined with Elman NN to form an FM-Elman neural model for nonstationary analysis which enjoys benefits from both FM and Elman NN and (2) we demonstrate the prediction accuracy in various metrics. The numerical experiments show significant improvements in prediction accuracy over the existing methods. However, the limitation of this research is that the proposed model is data dependent which does not guarantee excellent predictions on all datasets. And further study on high-order interactions among the inputs is also a challenging work.

The power of combining FM with neural network to achieve better performance will likely exist for the area of classification and regression which will be useful for future studies. We believe that FM can be used in conjunction with other deep learning network such as LSTM to form the high quality predicting method. Various combinations in techniques and approaches can be investigated in the future to solve problems occurring in different applications.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the R&D Program of Beijing Municipal Education Commission (grant no. KJZD20191000401). This research was also supported by the Program of the Co-construction with Beijing Municipal Commission of Education of China (grant nos. B20H100020 and B19H100010) and funded by the Key Project of Beijing Social Science Foundation Research Base (grant no. 19JDYJA001). Fang Wang is grateful for the support by the Beijing Laboratory of National Economic Security Early-Warning Engineering (grant no. B19H100030).

References

[1] K. Oh and K. J. Kim, "Analyzing stock market tick data using piecewise nonlinear model," *Expert Systems with Applications*, vol. 22, no. 3, pp. 249–255, 2002.

[2] Y. Wang, "Mining stock price using fuzzy rough set system," *Expert Systems with Applications*, vol. 24, no. 1, pp. 13–23, 2003.

[3] Y. Wang, L. Wang, F. Yang, W. Di, and Q. Chang, "Advantages of direct input-to-output connections in neural networks: the Elman network for stock index forecasting," *Information Sciences*, vol. 547, no. 8, pp. 1066–1079, 2021.

[4] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, and M. Woźniak, "A novel training method to preserve generalization of RBPNN classifiers applied to ECG signals diagnosis," *Neural Networks*, vol. 108, pp. 331–338, 2018.

[5] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 4, pp. 1–23, 2020.

[6] M. Woźniak and D. Połap, "Hybrid neuro-heuristic methodology for simulation and control of dynamic systems over time interval," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 93, pp. 45–56, 2017.

[7] J. Zhang, J. Li, and R. Wang, "Instantaneous mental workload assessment using time-frequency analysis and semi-supervised learning," *Cognitive Neurodynamics*, vol. 14, no. 5, pp. 619–642, 2020.

[8] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[9] J. Y. Zheng, "Forecast of opening stock price based on Elman neural network," *Chemical Engineering Transactions*, vol. 46, pp. 565–570, 2015.

[10] B. Wu and T. Duan, "A performance comparison of neural networks in forecasting stock price trend," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 336–346, 2017.

[11] B. Wu and T. Duan, "The fractal feature and price trend in the gold future market at the Shanghai Futures Exchange (SFE),"

Physica A: Statistical Mechanics and Its Applications, vol. 474, no. 15, pp. 99–106, 2017.

[12] R. H. J. Rani and T. A. A. Victoire, "A hybrid Elman recurrent neural network, group search optimization, and refined VMD-based framework for multi-step ahead electricity price forecasting," *Soft Computing*, vol. 23, no. 18, pp. 8413–8434, 2019.

[13] M. Tripathy, "Power transformer differential protection using neural network principal component analysis and radial basis function neural network," *Simulation Modelling Practice and Theory*, vol. 18, no. 5, pp. 600–611, 2010.

[14] H. Niu and J. Wang, "Financial time series prediction by a random data-time effective RBF neural network," *Soft Computing*, vol. 18, no. 3, pp. 497–508, 2014.

[15] S. Yousefi, I. Weinreich, and D. Reinarz, "Wavelet-based prediction of oil prices," *Chaos, Solitons & Fractals*, vol. 25, no. 2, pp. 265–275, 2005.

[16] L. Huang and J. Wang, "Global crude oil price prediction and synchronization based accuracy evaluation using random wavelet neural network," *Energy*, vol. 151, no. 15, pp. 875–888, 2018.

[17] M. Bildirici, E. A. Alp, and Ö. Ö. Ersin, "TAR-cointegration neural network model: an empirical analysis of exchange rates and stock returns," *Expert Systems with Applications*, vol. 37, no. 1, pp. 2–11, 2010.

[18] R. Dash, S. Samal, R. Dash, and R. Rautray, "An integrated topsis crow search based classifier ensemble: in application to stock index price movement prediction," *Applied Soft Computing*, vol. 85, pp. 1–14, 2019.

[19] K. J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[20] X. Y. Qian and S. Gao, "Financial series prediction: comparison between precision of time series models and machine learning methods," 2018.

[21] G. Armano, M. Marchesi, and A. Murru, "A hybrid genetic-neural architecture for stock indexes forecasting," *Information Sciences*, vol. 170, no. 1, pp. 3–33, 2005.

[22] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.

[23] J. Wang, H. Pan, and F. Liu, "Forecasting crude oil price and stock price by jump stochastic time effective neural network model," *Journal of Applied Mathematics*, vol. 2012, Article ID 646475, 15 pages, 2012.

[24] J. Wang and J. Wang, "Forecasting energy market indices with recurrent neural networks: case study of crude oil price fluctuations," *Energy*, vol. 102, no. 1, pp. 365–374, 2016.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, http://arxiv.org/abs/1409.0473.

[27] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[28] S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, pp. 995–1000, Sydney, Australia, December 2010.

[29] X. He, H. Zhang, M. Y. Kan, and T. S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 549–558, Pisa, Italy, July 2016.

[30] Y. Liu, W. Guo, D. Zang, and Z. Li, "A hybrid neural network model with non-linear factorization machines for collaborative recommendation," *Lecture Notes in Computer Science*, vol. 11168, pp. 213–224, 2018.

[31] Q. Wang, F. Liu, S. Xing, X. Zhao, and T. Li, "Research on CTR prediction based on deep learning," *IEEE Access*, vol. 7, pp. 12779–12789, 2018.

[32] W. Ni, T. Liu, Q. Zeng, X. Zhang, H. Duan, and N. Xie, "Robust factorization machines for credit default prediction," in *Proceedings of the PRICAI 2018: Trends in Artificial Intelligence*, pp. 941–953, Nanjing, China, August 2018.

[33] F. Zhou, H. Zhou, Z. Yang, and L. Yang, "EMD2FNN: a strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction," *Expert Systems with Applications*, vol. 115, pp. 136–151, 2019.

[34] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.

[35] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," *Neural Computing & Applications*, vol. 10, no. 2, pp. 184–192, 2001.

Hindawi
Complexity
Volume 2021, Article ID 6647534, 16 pages
https://doi.org/10.1155/2021/6647534

Research Article

Forecasting Foreign Exchange Volatility Using Deep Learning Autoencoder-LSTM Techniques

Gunho Jung[1] **and Sun-Yong Choi**[2]

[1]*Department of Artificial Intelligence, Korea University, Seoul 02841, Republic of Korea*
[2]*Department of Financial Mathematics, Gachon University, Seongnam-si, Gyeoggi 13120, Republic of Korea*

Correspondence should be addressed to Sun-Yong Choi; sunyongchoi@gachon.ac.kr

Received 28 December 2020; Revised 30 January 2021; Accepted 20 March 2021; Published 31 March 2021

Academic Editor: Benjamin Miranda Tabak

Since the breakdown of the Bretton Woods system in the early 1970s, the foreign exchange (FX) market has become an important focus of both academic and practical research. There are many reasons why FX is important, but one of most important aspects is the determination of foreign investment values. Therefore, FX serves as the backbone of international investments and global trading. Additionally, because fluctuations in FX affect the value of imported and exported goods and services, such fluctuations have an important impact on the economic competitiveness of multinational corporations and countries. Therefore, the volatility of FX rates is a major concern for scholars and practitioners. Forecasting FX volatility is a crucial financial problem that is attracting significant attention based on its diverse implications. Recently, various deep learning models based on artificial neural networks (ANNs) have been widely employed in finance and economics, particularly for forecasting volatility. The main goal of this study was to predict FX volatility effectively using ANN models. To this end, we propose a hybrid model that combines the long short-term memory (LSTM) and autoencoder models. These deep learning models are known to perform well in time-series prediction for forecasting FX volatility. Therefore, we expect that our approach will be suitable for FX volatility prediction because it combines the merits of these two models. Methodologically, we employ the Foreign Exchange Volatility Index (FXVIX) as a measure of FX volatility. In particular, the three major FXVIX indices (EUVIX, BPVIX, and JYVIX) from 2010 to 2019 are considered, and we predict future prices using the proposed hybrid model. Our hybrid model utilizes an LSTM model as an encoder and decoder inside an autoencoder network. Additionally, we investigate FXVIX indices through subperiod analysis to examine how the proposed model's forecasting performance is influenced by data distributions and outliers. Based on the empirical results, we can conclude that the proposed hybrid method, which we call the autoencoder-LSTM model, outperforms the traditional LSTM method. Additionally, the ability to learn the magnitude of data spread and singularities determines the accuracy of predictions made using deep learning models. In summary, this study established that FX volatility can be accurately predicted using a combination of deep learning models. Our findings have important implications for practitioners. Because forecasting volatility is an essential task for financial decision-making, this study will enable traders and policymakers to hedge or invest efficiently and make policy decisions based on volatility forecasting.

1. Introduction

Among various financial asset markets, the foreign exchange (FX) market has become increasingly volatile and fluid over the past decade. According to data released by BIS (Bank for International Settlements) in April of 2019, the global trading volume of FX commodity markets was $6.6 trillion per day, representing a 30% increase compared to April of 2016 ($5.1 trillion). With the advent of globalization and increased demand for overseas investment, the number of FX transactions has increased rapidly based on investments in companies in various countries. Additionally, FX rates significantly affect the estimation of currency risks and profits for international trades. Governments and policymakers are keeping a close watch on FX fluctuations to perform risk management. Therefore, FX is considered to be

the most important financial index for international monetary markets (Huang et al. [1]).

In addition to FX rates, FX volatility has also been a significant source of concern for practitioners. FX volatility is defined by fluctuations in FX rates, so it is also known as a measure of FX risk. Because FX risk is directly linked to transaction costs related to international trade, it is of great importance for multinational firms, financial institutions, and traders who wish to hedge currency risks. In this regard, FX volatility has affected the external sector competitiveness of international trade and the global economy.

In particular, financial asset price volatility is a crucial concern for scholars, investors, and policymakers. This is because volatility is important for derivative pricing, hedging, portfolio selection, and risk management (see Vasilellis and Meade [2], Knopf et al. [3], Brownlees and Gallo [4], Gallo and Otranto [5], and Bollerslev et al. [6]). Therefore, the forecasting and modeling of volatility have recently become the focus of many empirical studies and theoretical investigations in academia. Forecasting volatility accurately remains a crucial challenge for scholars.

Because many academics and practitioners are interested in volatility, many studies on volatility prediction have been reported. In these studies, many approaches have been utilized for forecasting. The autoregressive conditional heteroscedasticity (ARCH) and generalized ARCH (GARCH) models proposed by Bollerslev [7] are mainly used to predict volatility (Vee et al. [8], Dhamija and Bhalla [9], Bala and Asemota [10], Kambouroudis et al. [11], and Köchling et al. [12]). Various characteristics of volatility, such as leverage effects, volatility clustering, and persistence (Cont [13] and Cont [14]), are the main reasons for employing GARCH-based models. Based on the recent development of artificial neural network (ANN) models, the use of ANN methods for forecasting volatility has increased (Pradeepkumar and Ravi [15], Liu [16], Ramos-Pérez et al. [17], and Bucci [18]). Previous studies have employed various ANN models, such as the random forest (RF) (Breiman [19]), support vector machine (SVM) (Cortes and Vapnik [20]), and long short-term memory (LSTM) (Hochreiter and Schmidhuber [21]). Several studies have shown that ANN methods outperform GARCH-based models for forecasting time series (see Pradeepkumar and Ravi [15], Liu [16], and Bucci [18]). Additionally, hybrid models based on ANNs and GARCH-type models have been introduced (Hajizadeh et al. [22], Kristjanpoller et al. [23], Kristjanpoller and Minutolo [24], Kim and Won [25], Baffour et al. [26], and Hu et al. [27]). Such models are reported to have advantages compared to using ANNs or GARCH-based models alone. Additional literature on this topic will be covered in Section 2.

Based on the discussion above, we focus on volatility forecasting based on FX volatility. As measures of FX volatility, we adopt three FX volatility indexes (FXVIXs), namely, the FX euro volatility index (EUVIX), FX British pound volatility index (BPVIX), and FX yen volatility index (JYVIX), which are equally weighted indices of the Chicago Board Option Exchange's (CBOE's) 30 day implied volatility readings for the euro (EUR), pound sterling (GBP), and Japanese yen (JPY), respectively. Because the three currency pairs of EUR/USD, USD/JPY, and GBP/USD are the three most heavily traded currency pairs on the FX market, we selected the three corresponding FXVIX indices. Additionally, these indexes reflect global economic trends (see Ishfaq et al. [28], Dicle and Dicle [29], and Pilbeam [30]). As mentioned previously, the forecasting of volatility in the FX market is important for global firms, financial institutions, and traders who wish to hedge currency risks (see Guo et al. [31], Abdalla [32], and Menkhoff et al. [33]).

Practically, the FX market consists of three associated components: spot transactions, forward transactions, and derivative contracts (Baffour et al. [26]). Additionally, because FX was originally defined by two currencies, FX has more observable factors that affect changes compared to other financial indices. Furthermore, according to Liu et al. [34], the periodic characteristics of the FX market are some of the main reasons why it is difficult to predict changes in the FX market. Therefore, we utilize ANN models as data-driven methods, rather than model-driven methods such as GARCH-type models, to forecast the three aforementioned FXVIXs. In particular, we employ the LSTM and autoencoder (Rumelhart et al. [35]) models as ANN techniques. We propose a hybrid neural network model based on these two models. To combine an autoencoder with LSTM, we apply LSTM as an encoder and decoder for sequence data inside an autoencoder network. Therefore, the proposed hybrid model can leverage the advantages of both the autoencoder and LSTM. A detailed discussion of this topic is presented in Section 3.

Methodologically, we adopt a machine learning algorithm (LSTM) to implement an autoencoder-LSTM model for forecasting FXVIXs from 2010 to 2019. We optimize the adopted algorithms using a grid search procedure provided by Full-Stack Python. Testing is also performed using subperiod analysis to investigate whether data deviations and outliers affect model training. Such subperiod analysis has been commonly implemented in previous studies (Sharma et al. [36], García and Kristjanpoller [37], Ramos-Pérez et al. [17], and Choi and Hong [38]). Specifically, we split the entire sample period into three subperiods called Period 1 (January, 2010 to December, 2015), Period 2 (January, 2016 to December, 2016), and Period 3 (January, 2017 to December, 2019). Period 2 exhibits uncertainty in the European market based on the Brexit movement. In this manner, we investigate the accuracy of prediction and model performance according to different data states.

There are two major aspects of this study that differ from previous studies. First, we use FXVIXs, which play key roles in the FX market. Although previous empirical studies have predicted various types of financial asset price volatility using various models, research on forecasting FXVIXs is scarce. Additionally, research on FX price prediction and volatility prediction using various approaches is being conducted, but research on the prediction of the FXVIX is relatively rare. Therefore, it is necessary to predict FXVIX volatility. Second, we propose a hybrid model based on an autoencoder and LSTM to forecast the three FXVIXs. LSTM is known to be good at forecasting time series (Fischer and

Krauss [39], Kumar et al. [40], and Muzaffar and Afshari [41]), and one of the advantages of an autoencoder is that it can automatically extract features from input data (Phaisangittisagul and Chongprachawat [42], Zhang et al. [43], and Zeng et al. [44]). Therefore, the autoencoder technique has been widely used to predict time series data (Saha et al. [45], Lv et al. [46], Sagheer and Kotb [47], and Boquet et al. [48]). The proposed hybrid model has excellent potential as a novel method for forecasting the FXVIX and time series.

The main contributions of this paper can be summarized as follows:

First, we expand upon previous studies by forecasting the FXVIX using ANN models. Our experiments were motivated by the observation that previous studies on the FX market have mainly focused on the FX rate, volatility of returns, or historical volatility. In particular, FXVIXs represent future FX risk measures for market participants. Therefore, our findings have important implications for practitioners managing FX risk exposure.

Second, we propose a hybrid ANN model based on an autoencoder and LSTM. Forecasting performance results demonstrate that the proposed hybrid model outperforms traditional LSTM models. Consequently, this study contributes to the literature on developing ANN models by introducing a novel hybrid model.

Our third major contribution is the optimization of model forecasting performance through subperiod analysis. Based on the empirical results of subperiod analysis, we can conclude that a wide distribution of input data and acceptable number of outliers improve forecasting performance.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review on FX volatility and studies using machine learning in finance. Section 3 describes the data and methodology adopted in this study. Section 4 presents the results of empirical analysis for the full sample period and subperiod analysis. Finally, we provide concluding remarks in Section 5.

2. Literature Review

There is a vast body of literature on forecasting financial time series. In this section, we divide previous research into FX rate and FX volatility research according to the main focus of previous papers. Additionally, we also discuss literature on time-series forecasting using ANNs.

First, because the FX rate directly affects the income of multinational firms, many studies have focused on the forecasting FX rate and many studies have used ANN models to predict future FX rates. For example, Liu et al. [34] predicted EUR/USD, GBP/USD, and JPY/USD rates using a model based on a convolutional neural network (CNN). They demonstrated that such a model is suitable for processing 2D structural exchange rate data. Fu et al. [49] developed evolutionary support vector regression (SVR) models to forecast four Renminbi (RMB, Chinese yuan) exchange rates (CNY against USD, EUR, JPY, and GBP). They also demonstrated that the proposed model outperforms the multilayer perceptron (MLP) neural network, Elman neural network, and SVR models in terms of level forecasting accuracy measures. The authors of Sun et al. [50] introduced a novel ensemble deep learning approach based on LSTM and a bagging ensemble learning strategy to predict four major currencies (EUR/USD, GBP/USD, JPY/USD, and USD/CNY). According to their empirical results, their proposed model provided significantly improved forecasting accuracy compared to a traditional LSTM model.

As discussed in the previous section, FX volatility is also important for many academics and practitioners, so many studies have focused on FX volatility forecasting. In general, GARCH-based models have been used in many studies to predict FX volatility. Additionally, some studies have predicted FX volatility by incorporating different methodologies into GARCH models to improve forecasting power. For example, the authors of Vilasuso [51] predicted various FX rate volatilities (Canadian dollar, French franc, German mark, Italian lira, Japanese yen, and British pound) using a fractionally integrated GARCH (FIGARCH) model (Baillie et al. [52]). The empirical results of their study demonstrated that the FIGARCH model is better at capturing the features of FX volatility compared to the original GARCH model. The authors of Rapach and Strauss [53] demonstrated that structural breaks in the unconditional variance of FX rate returns can improve the forecasting performance of GARCH(1,1) models for FX volatility by incorporating the daily returns of the US dollar against the currencies of Canada, Denmark, Germany, Japan, Norway, Switzerland, and the UK. Pilbeam and Langeland [54] investigated whether various GARCH-based models can effectively forecast the FX volatility of the four currency pairs of the euro, pound, Swiss franc, and yen against the US dollar. In particular, their empirical results demonstrated that GARCH models perform better in periods of low volatility compared to periods of high volatility. You and Liu [55] employed the GARCH-MIDAS approach (Engle et al. [56]) to forecast the short-run volatility of six FX rates based on monetary fundamentals. They demonstrated that the forecasting power of daily FX volatility is significantly improved by including monthly monetary fundamental volatilities.

Various machine learning models have also been used to forecast time series originating from various fields, including engineering and finance. In finance, many studies have used machine learning to predict future stock prices. For example, Trafalis and Ince [57] compared SVR with backpropagation to a radial basis function network on the task of forecasting daily stock prices. Similarly, Henrique et al. [58] utilized SVR and a random walk (RW) method to predict daily stock prices in three different markets (Brazilian, American, and Chinese). Based on comparisons of the price prediction results of the SVR and RW models, they determined that SVR models may perform better than RW models in terms of predictive performance. Recently, various studies using machine learning methods and deep learning methodologies have been reported. For example, the authors of Selvin et al. [59] employed deep learning models, namely, a recurrent

neural network (RNN), LSTM, and CNN to predict minute-wise stock prices. They determined that the CNN algorithm provided the best performance. Chong et al. [60] employed an autoencoder to extract features from stock data and constructed a deep neural network (DNN) to predict future stock returns. They determined that it is possible to extract features from a large set of raw data without relying on prior knowledge regarding predictors, which is one of the main advantages of DNNs. Pradeepkumar and Ravi [15] proposed a particle swarm optimization-trained quantile RNN to forecast FX volatility. Their model provides superior forecasting performance compared to the GARCH model. In [16] and [18], various ANN models were employed to predict the volatility of the S&P 500 stock index. According to the findings of these studies, ANN models are able to outperform traditional econometric methods, including GARCH and autoregressive moving average models. In particular, LSTM models seem to improve the accuracy of volatility forecasts. Additionally, Ramos-Pérez et al. [17] predicted S&P 500 index volatility using a stacked ANN model based on a set of various machine learning techniques, including gradient descent boosting, RF, and SVM. They demonstrated that volatility forecasts can be improved by stacking machine learning algorithms. Additionally, regardless of the volatility model adopted, high-volatility regimes lead to higher error rates.

Several studies have proposed hybrid models based on GARCH-based models and ANN models. For example, various GARCH-based models have been combined with ANNs based on MLPs and many hybrid models have been used to enhance the ability of GARCH models to forecast the volatility of stocks, gold, and FX rate returns (Hajizadeh et al. [22], Kristjanpoller et al. [23], Kristjanpoller and Minutolo [24], and Baffour et al. [26]). Additionally, some studies have proposed hybrids of LSTM and GARCH models and have used such models to predict the volatility of financial assets (Kim and Won [25] and Hu et al. [27]). According to empirical results, hybrid models based on GARCH and ANN techniques exhibit improved forecasting performance in terms of volatility accuracy.

In particular, we focus on studies using LSTM and autoencoder approaches for forecasting time series. LSTM, which was introduced by Hochreiter and Schmidhuber [21], has been widely used to forecast time series in many prediction studies. This method is mainly used to analyze time-series data because it can keep records of past data. Some studies have compared LSTM to traditional methods using neural networks or investigated such models by reconstructing both types of methods. As discussed by Siami-Namini et al. [61] and Ohanyan [62], as computing power improves, implementing deep learning models becomes more practical, and their performance exceeds that of traditional models. Additionally, Deorukhkar et al. [63] demonstrated that neural network models combined with autoregressive integrated moving average or LSTM models provide greater accuracy than either type of model individually. In [64], the method of applying preprocessed stock prices to an LSTM model using a wavelet transform was shown to be superior to traditional methods.

The autoencoder presented in [35] aims to generate a representation as close to an original input as possible from reduced encoding results. This method is a transformation of the basic model using stacked layers, denoising, and sparse representation and is used for financial time series prediction. Bao et al. [65] used LSTM and stacked autoencoders to forecast stock prices and demonstrated that this type of hybrid model is more powerful than an RNN or LSTM model alone. In [66], a stacked denoising autoencoder applied to gravitational searching was effective at predicting the direction of stock index movement, which is affected by underlying assets. Additionally, Sun et al. [67] explained that a stacked denoising autoencoder formed through the selection of training sets based on a K-nearest neighbors approach can improve the accuracy compared to traditional methods.

This study enhances the existing literature in two main aspects. We first propose a hybrid model that combines LSTM and an autoencoder to forecast FX volatility. There are other studies that have used hybrid models, but they have used models other than autoencoders and LSTM. Additionally, most studies have developed hybrid models based on GARCH models. However, as discussed above, LSTM and autoencoders perform well at time-series prediction, so we adopted these two types of models to forecast FX volatility. Second, as discussed in Section 1, FX volatility has great significance, but there is a significant lack of research on forecasting its changes. We contribute to the finance literature by forecasting FXVIXs using the proposed hybrid model.

3. Data Description and Methodologies

3.1. Data Description. The VIX was firstly implemented on the CBOE in 1993. This index is based on the real-time prices of options in the S&P 500 index. Because it is derived from the price inputs of S&P 500 index options, this index not only represents market expectations regarding 30 day forward-looking volatility but also provides a measure of market risk and investor sentiments. Subsequently, various VIXs with different basic assets were developed.

In this study, we investigated whether machine learning methods are suitable for forecasting FX volatility time-series data. Our data samples come from the CBOE. The CBOE is one of the world's largest exchange holding companies, and it provides several derivatives related to implied VIXs. We adopted three currency-related volatility indices, namely, the BPVIX, JYVIX, and EUVIX. Similar to a VIX, FX volatility is calculated using a formula that averages the weighted prices of out-of-the-money puts and calls.

We collected 2520 daily time series FXVIX data from January of 2010 to December of 2019. Based on fluctuations caused by the Brexit movement, the data were divided into subsets from 2010 to 2015, 2016, and 2017 to 2019 based on instabilities in 2016. The first period represents the period of recovery following the subprime mortgage crisis and contains the most data (1514 daily data). As shown in Figure 1, the variability of the entire section appears to be large. This observation is confirmed by Table 1. The standard deviations

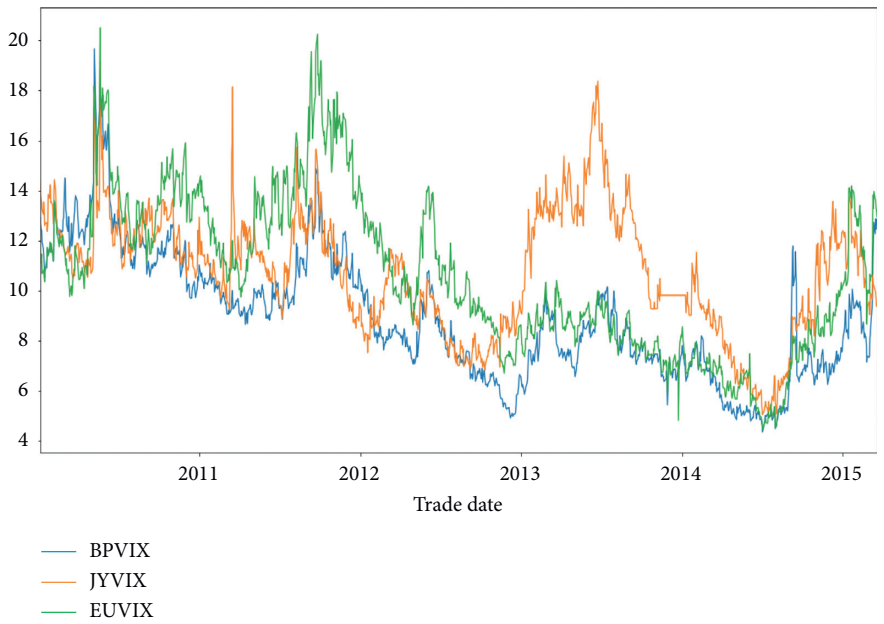


Figure 1: FXVIXs in period 1.

Table 1: Summary statistics for the three daily volatility indices.

	Max	Min	Mean	Std
Period 1				
BPVIX	19.67	4.33	8.93	2.46
JYVIX	19.17	5.03	10.58	2.36
EUVIX	20.51	4.43	10.78	3.04
Period 2				
BPVIX	29.10	5.31	12.02	4.21
JYVIX	17.54	9.60	12.80	1.70
EUVIX	19.41	4.62	9.67	1.75
Period 3				
BPVIX	15.68	5.16	9.24	2.03
JYVIX	14.66	4.29	8.18	1.81
EUVIX	12.74	3.99	7.11	1.42

of BPVIX, JYVIX, and EUVIX in this section are the largest among all periods, excluding BPVIX in 2016.

The second period represents the time around Brexit, which caused fluctuations in the global stock market, particularly in the European market. As shown in Figure 2, the UK index fluctuates the most, which affects the volatility of the European index. According to Table 1, BPVIX not only exhibits a high standard deviation but also has the largest difference between the maximum and minimum values.

The final period represents the time of uncertainty following the Brexit movement and recovery around the world. This period exhibits cyclic characteristics because the same problems arise repeatedly. Because intermediate trends between features of the first and second sections are visible, this section does not have any noteworthy features relative to the other sections. As shown in Figure 3, this period is longer than the second period, shorter than the first period, and less volatile than both periods, except JYVIX.

In this paper, for convenience, the three periods are referred to as Period 1, Period 2, and Period 3. Specifically, Period 1 ranges from 2010 to 2015, Period 2 covers 2016, and Period 3 ranges from 2017 to 2019. Similar subperiod analysis has been conducted in other studies (Gazioglu [68] and Grammatikos and Vermeulen [69]).

In machine learning, when constructing a model, performance evaluations are conducted. At this time, if a model trained on a particular training data set is evaluated on the same set, performance will be inflated by overfitting. Therefore, an original dataset should be divided into training and testing data, and a model should be trained on the training data. When evaluating performance, testing data, which were not used for training, are fed into the trained model. There is no ideal data allocation ratio for training and testing. With more training data, a model can see more examples and find better solutions, but overfitting may occur. Conversely, more testing data can lead to better generalization, but there underfitting may occur (Hastie et al. [70]).

According to Gu et al. [71], a simple data organization strategy generally uses 90% of the data for training and 10%

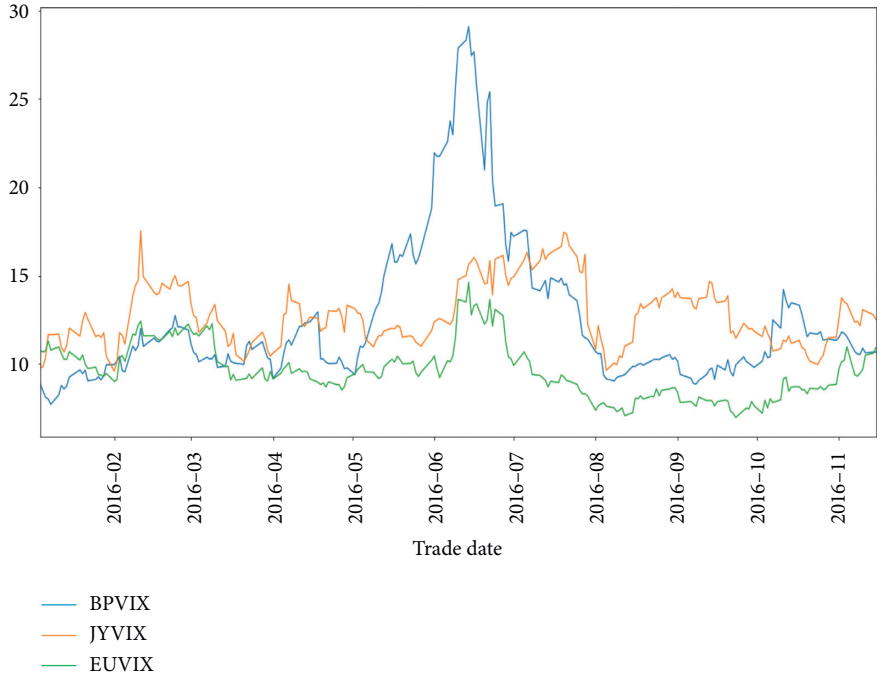


Figure 2: FXVIXs in period 2.

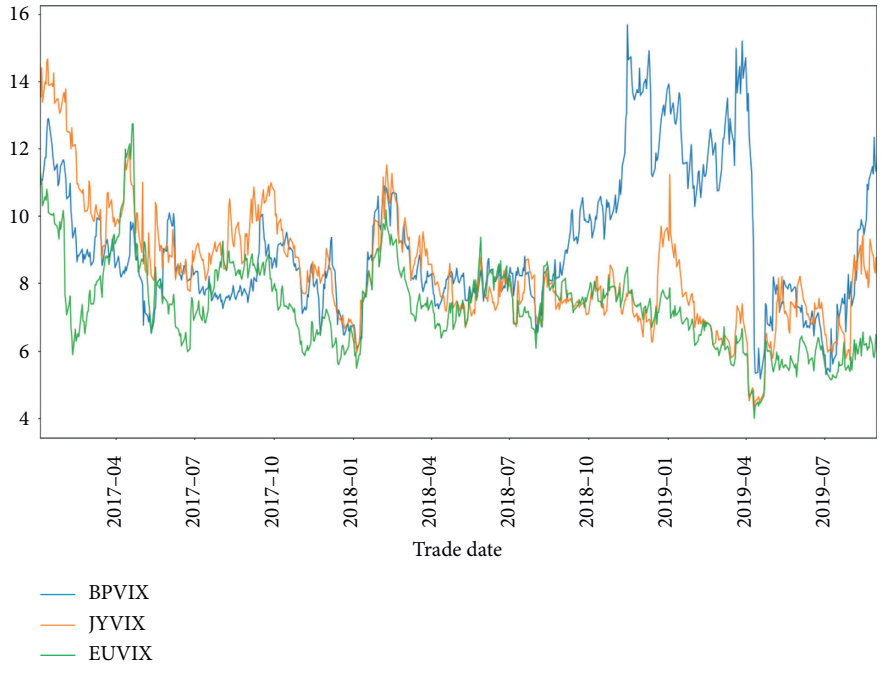


Figure 3: FXVIXs in period 3.

of the data for testing. This strategy was applied to the development of the Cubist regression tree model. We organized our data to use 85% of the data for training and 15% of the data for testing to avoid overfitting. Various data divisions are summarized in Table 2.

Cross-validation techniques were also applied to prevent overfitting. However, when a cross-validation method that selects random samples (e.g., K-fold cross-validation) is applied to time-series data, past values are predicted using future values. Therefore, in this study, time-series nested cross-validation was adopted to maintain the temporal order of the dataset for gradual overlapping and learning. The proposed model was trained and tuned on training and validation sets in each fold and then evaluated on a testing set. This allowed errors to be averaged to obtain an unbiased error estimate (Varma and Simon [72]).

Table 2: Descriptions of the training and testing datasets for each period.

Period	Training set	Test dataset
Period 1	(2010/01/04–2015/03/20)	(2015/03/23–2015/12/31)
Period 2	(2016/01/04–2016/11/15)	(2016/11/16–2016/12/30)
Period 3	(2017/01/03–2019/08/30)	(2019/09/03–2019/12/31)

3.2. LSTM. An RNN is a representative neural network with a recurrent hidden layer. Through this hidden layer, updates are backpropagated to train the model. By taking the results of previous hidden nodes as input data, it is possible to learn continuous forms. Therefore, RNNs are often used to analyze or predict time-series sequential data, such as stocks.

LSTM, which is a specific case of an RNN, was proposed by Hochreiter and Schmidhuber [21]. This model is designed to overcome the vanishing gradient problem of RNNs, where early layers are not trained properly when a network becomes deeper. Figure 4 presents the flow of RNN and LSTM progression. In contrast to the hidden units of the RNN, the LSTM structure consists of memory blocks. There are three steps (layers) in the LSTM model: the forget layer, which is the main advantage of LSTM, the input layer, and output layer. First, the forget gate f_t uses the sigmoid function, which is an activation function that converts current input data x_t and the previous hidden state h_{t-1} into numbers ranging from zero to one. Specifically, if an output is close to zero, it means that information cannot be passed to the next cell. In contrast, an output close to one means that information is passed to the next cell.

Second, the input gate i_t is a sigmoid function and decides which information in x_t and h_{t-1} is stored in the cell state C_t. At this step, there is also a tanh layer which creates a vector of new candidate values ($\check{C}_t$) that could be added to the cell state C_t. The cell state C_t is updated by combining the outputs from the forget gate f_t and input gate i_t. By multiplying f_t and C_{t-1}, the amount of information from the previous time step cell that will be retained is determined. Furthermore, i_t times $\check{C}_t$ represents the update information from the input gate.

Finally, the output gate O_t applies the sigmoid function to the previous hidden state h_{t-1} and current input x_t to decide what the next hidden state should be. In addition, the current cell state C_t is passed through a tanh function. We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry. In summary, the LSTM transition equations are defined as follows:

Gates:

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \\ i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \\ O_t &= \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right). \end{aligned} \tag{1}$$

Input transformation:

$$\check{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right). \tag{2}$$

Memory update:

$$\begin{aligned} C_t &= f_t \times C_{t-1} + i_t \times \check{C}_t, \\ h_t &= O_t \times \tanh\left(C_t\right), \end{aligned} \tag{3}$$

where W and b are the weights and biases, respectively. The $\times$ denotes elementwise multiplication.

In Figure 5, the input values x_t travel through three layers to overcome long-term dependencies using the following activation functions: σ (sigmoid) and tanh. The sigmoid function outputs a number between zero and one, which is a measure of how much information each component should convey. tanh helps keep the gradient as long as possible to prevent vanishing gradient problems.

3.3. Autoencoder

3.3.1. Basic Autoencoder. The autoencoder, which was first introduced in [35], utilizes a neural network consisting of an input layer, output layer, and hidden layers for self-supervised learning. Although this structure is similar to that of a typical neural network, the output and input layers have isomorphic vectors. The goal of this model is to derive a representation for an input dataset (e.g., dimensionality reduction) and make the reorganized data as close as possible to the input data. As shown in Figure 6, the encoder represents a stage at which the model can learn important characteristics of inputs and the decoder forms outputs similar to the inputs. The output represents a state in which the noise of the inputs is removed, resulting in more distinct characteristics. Based on these features, autoencoders are mainly used for image restoration or noise reduction.

$$Y = f\left(W_1 \cdot X + b\right), \tag{4}$$

$$\widetilde{X} = \widetilde{f}\left(W_2 \cdot Y + b\right), \tag{5}$$

where W_1 is the weight between input an X and hidden representation Y, W_2 is the weight between a hidden representation Y and output $\widetilde{X}$, and b is the bias, f and $\widetilde{f}$ represent the encoder and decoder, respectively, f accepts and compresses the input data (X) into a latent space (Y), and $\widetilde{f}$ is responsible for accepting latent space (Y) representations and reconstructing original inputs ($\widetilde{X}$).

This type of model is utilized in several methods to improve performance by manipulating hidden layers. A stacked autoencoder is used to solve the vanishing gradient problem by stacking hidden layers when a neural network is deep. Figure 7(a) presents a simple example of a stacked autoencoder. This structure increases the number of hidden nodes by stacking autoencoders hierarchically. A denoising autoencoder aims to extract stable structured data from dependent data by adding noise to input data and

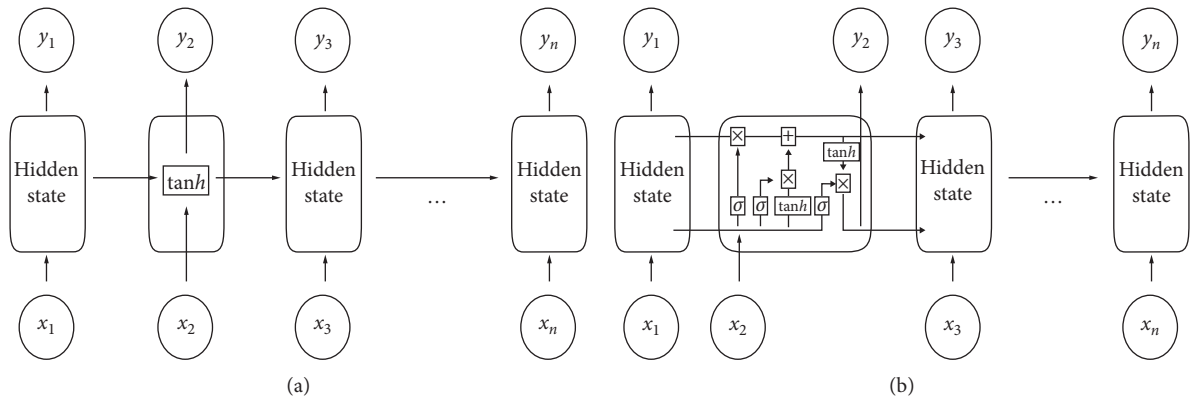


Figure 4: (a) RNN process. (b) LSTM process.

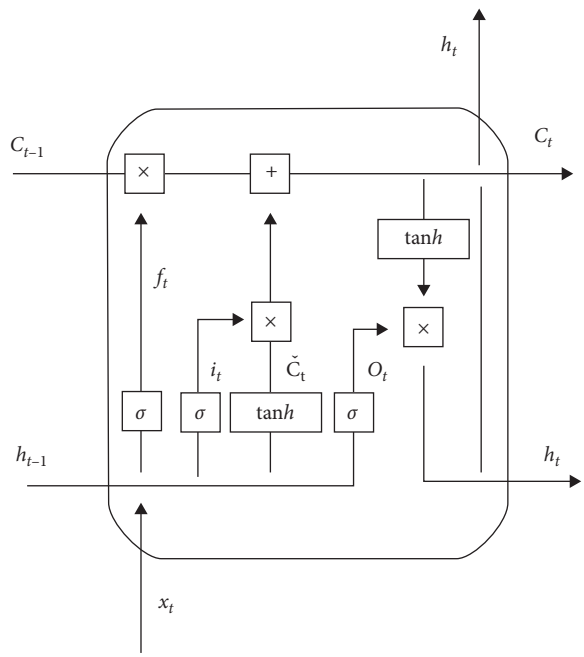


Figure 5: Structure of LSTM.

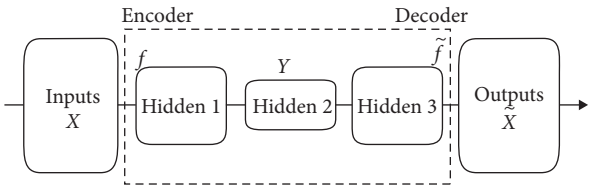


Figure 6: Structure of an autoencoder.

confirming that the output data correspond to pure input values. As shown in Figure 7(b), this model has a structure similar to that of a typical autoencoder, but it takes input data with added noise as new input data.

3.3.2. Autoencoder-LSTM. The autoencoder-LSTM model, which combines an autoencoder and advanced RNN, is implemented with an LSTM encoder and decoder for sequence data. This model has the same basic frame as an autoencoder, but is composed of LSTM layers, as shown in Figure 8(a). This model can learn complex and dynamic input sequence data from adjacent periods by using memory cells to remember long input sequence data.

The encoder and decoder components consist of two LSTM layers. To implement this structure, we adopted the "RepeatVector" tool provided by Keras, which is a deep learning API. Figure 8(b) presents the resulting structure.

3.4. Hyperparameter Optimization. A hyperparameter is a parameter that has a significant impact on the learning process. Maximizing model performance by finding optimal hyperparameter values to minimize a loss function is called hyperparameter optimization. This method is widely used in machine learning and deep learning. In this study, the well-known grid search method was adopted.

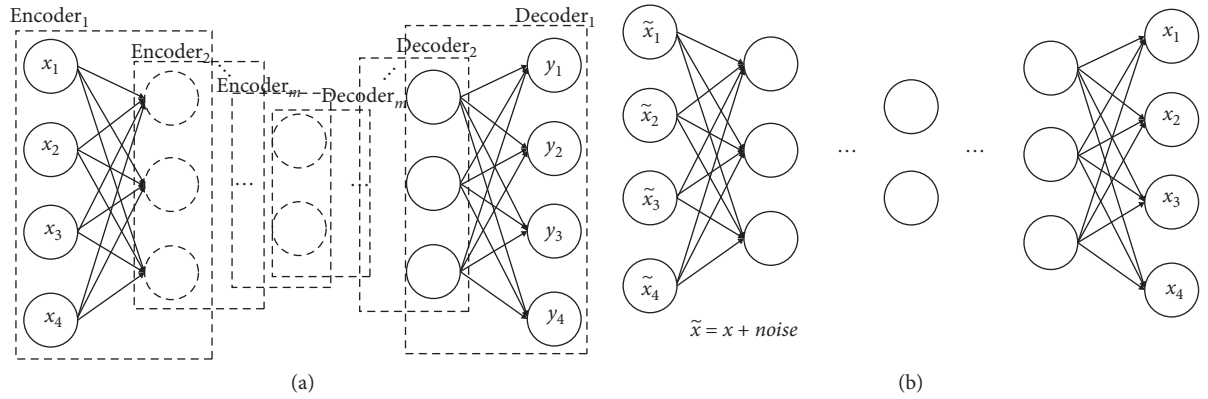


Figure 7: (a) Deep autoencoder. (b) Denoising and deep autoencoder (x_N denotes inputs with added noise).

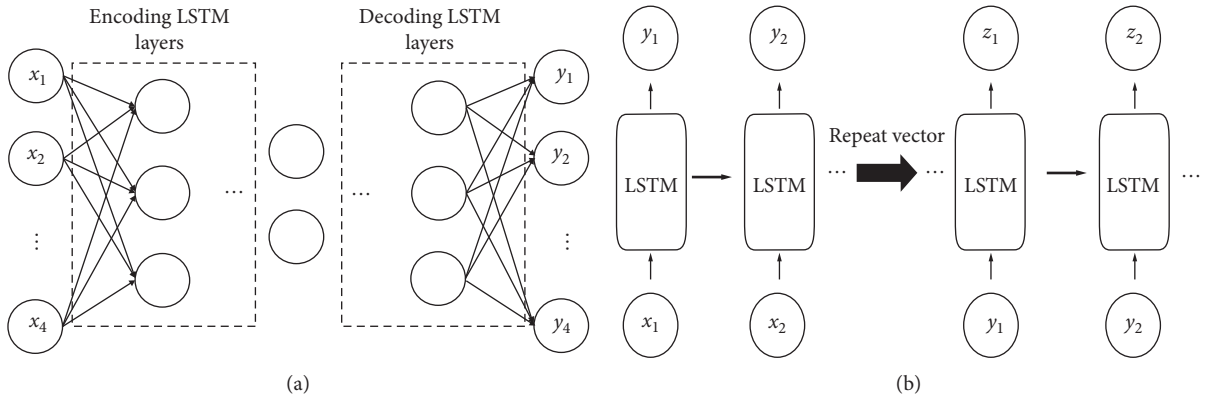


Figure 8: (a) Autoencoder-LSTM. (b) Structure of "RepeatVector."

A grid search finds the best parameters among a parameter set defined by a user and applies several parameter candidates to the model sequentially to identify the cases with the best performance. If there are few parameter candidates, optimal values can be obtained rapidly. However, if there are many candidates, optimization requires exponentially more time.

In this study, we adopted the grid search algorithm because it is the simplest and most widely used algorithm for obtaining optimal hyperparameters (Schilling et al. [73]). Although a random search can perform much better than a grid search on high-dimensional problems according to Hutter et al. [74], our data represent a simple time series and the candidate parameter set is limited. These are the main reasons why we adopted the grid search algorithm (Sun et al. [75] and Thornton et al. [76]). The Python technological stack was used for our experiments. We implemented the machine learning algorithms and grid search using the Scikit Learn, Keras, and TensorFlow packages.

We used a grid search to identify and apply optimal parameters for each section of our model. The optimized parameters are the batch size, activation function, and optimizer function. Two or three candidate groups were defined for each parameter.

More parameters and candidate groups could be defined, but it would increase training time significantly. We divided the data into three intervals and attempted to compare two models, thereby limiting the candidate groups to make the most of our limited resources.

Next, we optimized three parameters for stochastic gradient descent. The candidate batch sizes were 50 and 100, the activation functions were linear and ReLU, and the optimization functions were Adam, rmsprop, and nadam. The learning rates were default values built into each activation function (sprosprop: 0.001, Adam: 0.001, and nadam: 0.002).

Finally, the autoencoder and autoencoder-LSTM models were unified into four layers: two encoding layers and two decoding layers. Based on the small amount of testing data, this small depth was determined to be sufficient.

4. Empirical Results

We used the aforementioned grid search to find optimal parameter combinations. Among a total of 12 parameter combinations, the best parameters were identified and six optimizations were performed for the two models (LSTM and autoencoder-LSTM) and three periods in the same

Table 3: Hyperparameter table.

	Activation	Optimizer	Batch size
Period 1			
LSTM	ReLU	RMSProp	50
AutoEncoder-LSTM	Linear	Nadam	100
Period 2			
LSTM	Linear	RMSProp	100
AutoEncoder-LSTM	Linear	RMSProp	100
Period 3			
LSTM	Linear	Adam	50
AutoEncoder-LSTM	Linear	Adam	100

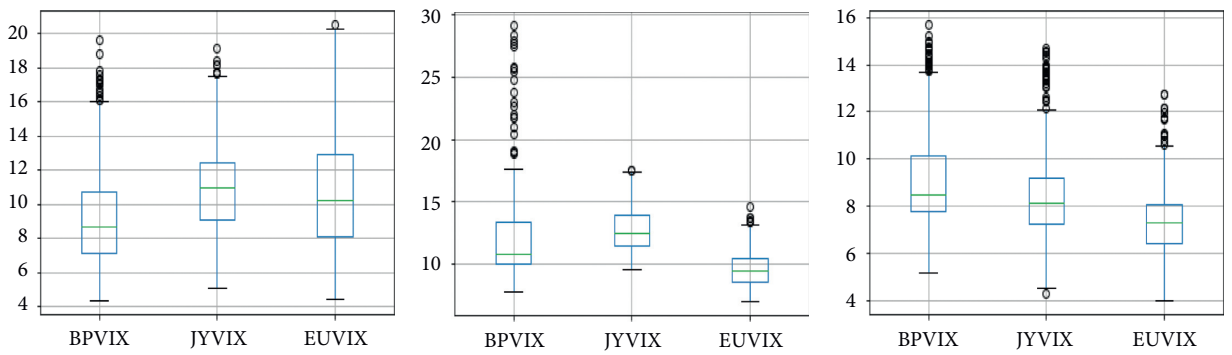


Figure 9: (a) Outliers in period 1. (b) Outliers in period 2. (c) Outliers in Period 3.

Table 4: Five error measures for each model in the BPVIX.

	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.1550	0.3937	0.2656	−0.4904	3.0105
Autoencoder-LSTM	0.1846	0.4296	0.2839	−0.2830	3.2280
Period 2					
LSTM	2.6645	1.6323	1.3887	9.5562	19.0619
Autoencoder-LSTM	1.4706	1.2127	0.8508	−3.0532	10.1669
Period 3					
LSTM	2.2017	1.4838	1.0805	−6.1679	11.1552
Autoencoder-LSTM	1.6874	1.2990	0.9039	−0.5791	9.1082

Table 5: Five error measures for each model in the JYVIX.

	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.6152	0.7843	0.4611	−0.4230	4.6438
Autoencoder-LSTM	0.4359	0.6602	0.4430	−1.3870	4.5153
Period 2					
LSTM	0.3321	0.5763	0.4525	−0.2644	3.5579
Autoencoder-LSTM	0.4261	0.6528	0.4933	0.79419	3.9160
Period 3					
LSTM	3.8273	1.9563	1.0422	−5.3252	15.0497
Autoencoder-LSTM	2.7176	1.6485	0.9447	2.6840	13.4985

Table 6: Five error measures for each model in the EUVIX.

	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.4104	0.6406	0.4893	2.5107	4.0954
Autoencoder-LSTM	0.2874	0.5361	0.3974	−0.2569	3.3006
Period 2					
LSTM	4.5074	2.1231	1.4746	3.8257	19.3755
Autoencoder-LSTM	3.8284	1.9566	1.4937	−6.4252	18.0702
Period 3					
LSTM	1.5205	1.2331	0.4530	0.5383	7.8089
Autoencoder-LSTM	1.3250	1.1511	0.5018	−1.9384	8.7721

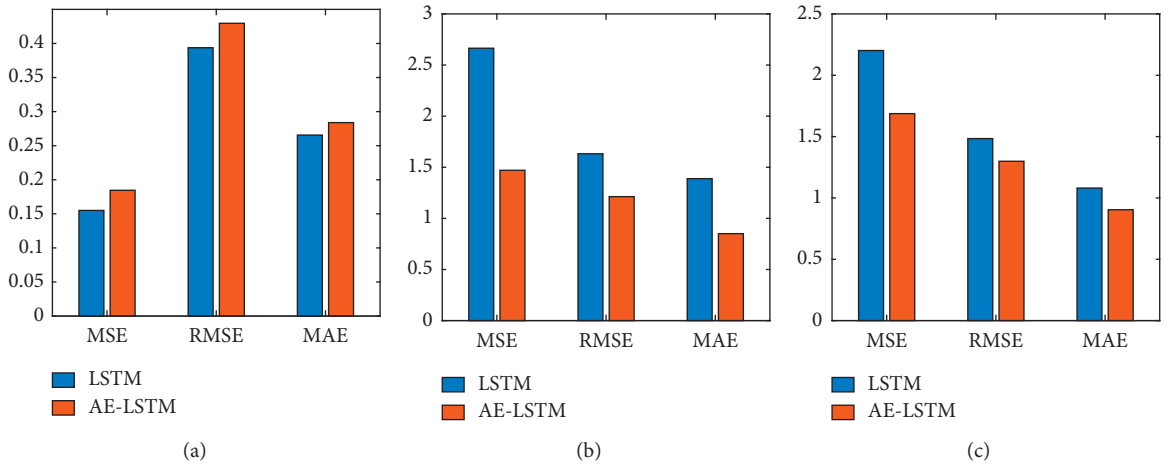


Figure 10: MSE, RMSE, and MAE values of the BPVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

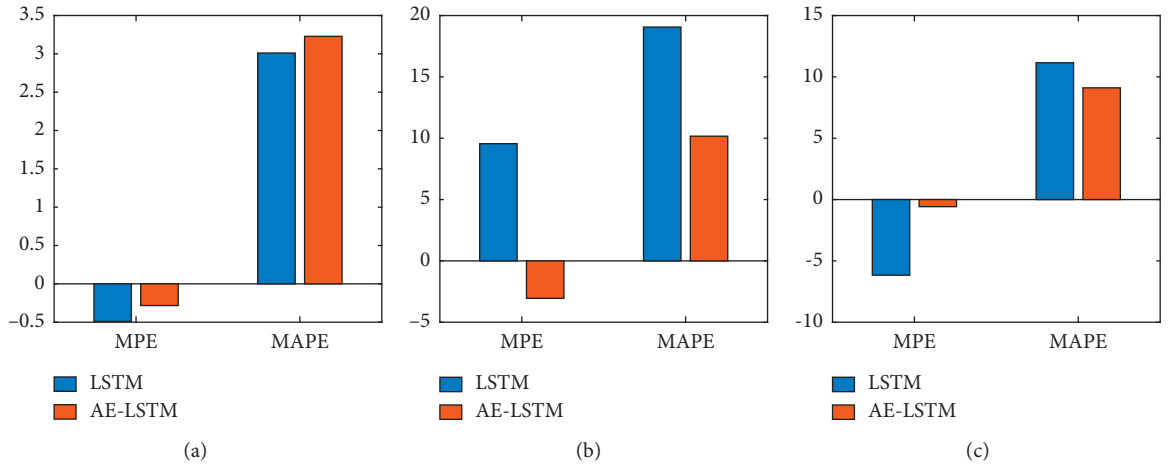


Figure 11: MPE and MAPE values of the BPVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

manner. The results obtained via hyperparameter optimization are listed in Table 3.

The goal of this study was to obtain an accurate model for forecasting FXVIXs. We considered three FXVIXs with different distributions and outliers were different. We compare the forecasting performances of our models in terms of distributions and outliers. To this end, the forecasting results are split by period and separated by index. As methods for measuring error, the regression error metrics of mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were adopted. Additionally, distributions were defined by variances and standard deviations. Outlier detection was applied using Tukey's box plot method, which defines outliers as samples that do not fall within the scope defined below:

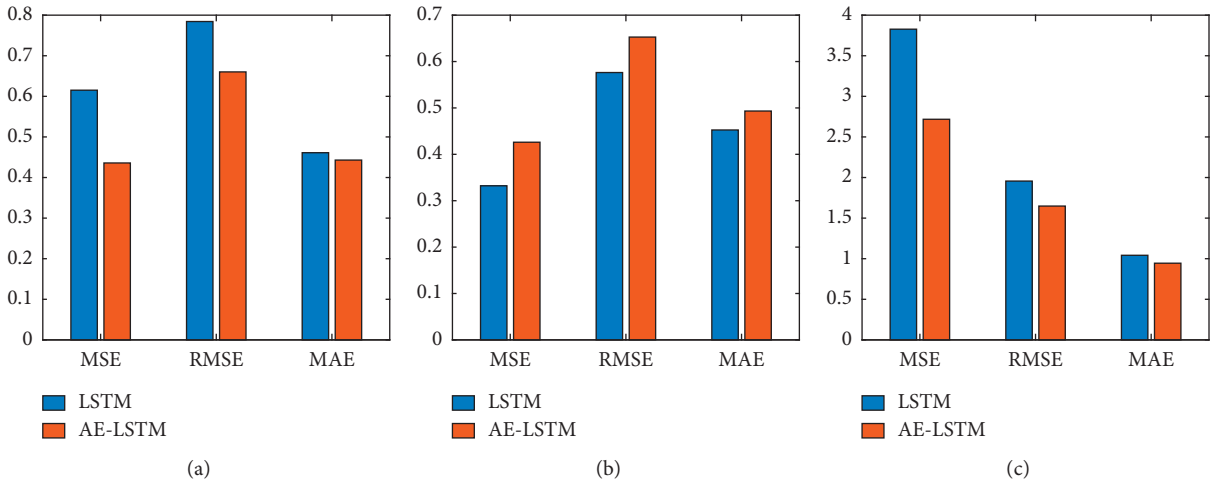


Figure 12: MSE, RMSE, and MAE values of the JYVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

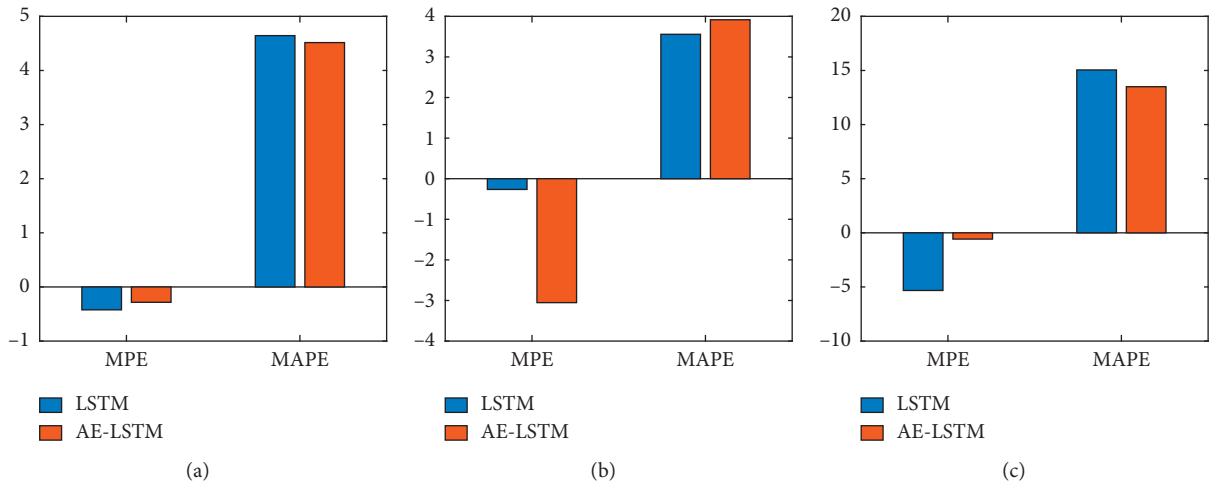


Figure 13: MPE and MAPE values of the JYVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

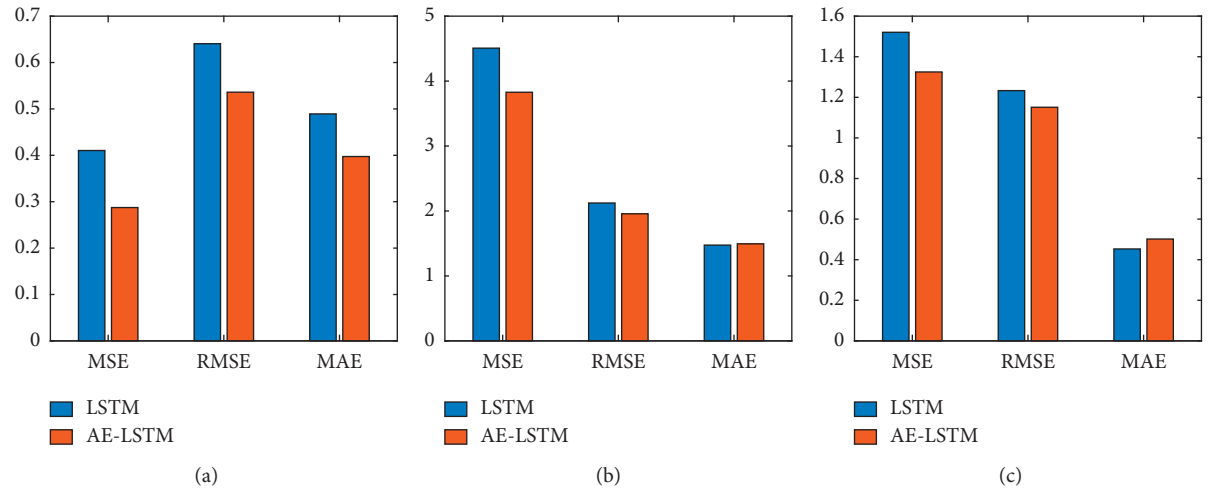


Figure 14: MSE, RMSE, and MAE values of the EUVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

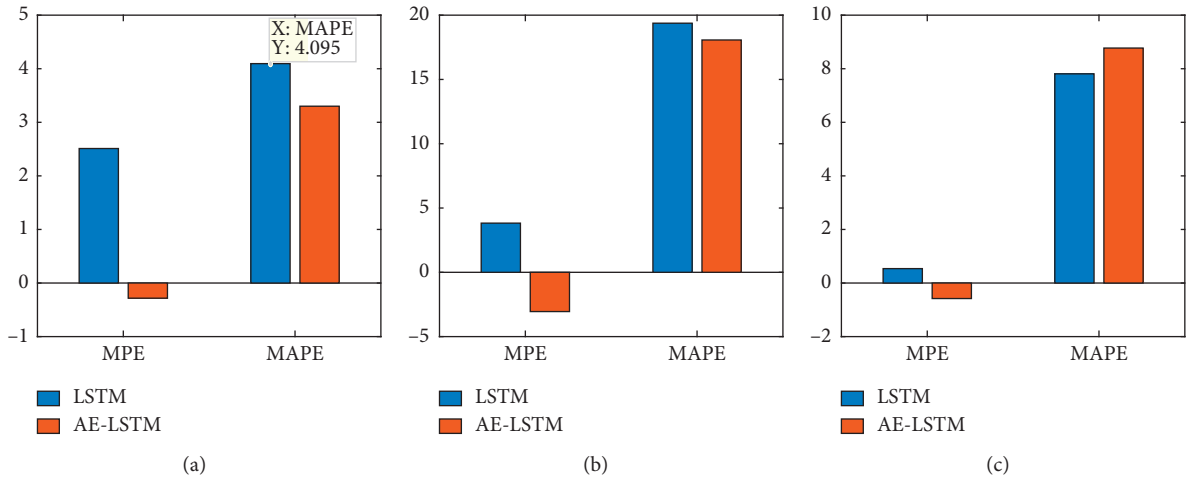


Figure 15: MPE and MAPE values of the EUVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

$$[Q_1 - 1.5IQR, Q_3 + 1.5IQR], \tag{6}$$

where Q is the quantile and IQR is an interquantile range defined as follows: $IQR = Q_3 - Q_1$. To identify extreme outliers, multiplication by 1.5 was replaced with multiplication by 3.

Our main findings can be summarized as follows. First, the opportunities to learn volatility and forecast accuracy have a proportional relationship. In other words, there are many sections that rise and fall in the training data and learning these trends can improve prediction accuracy. As shown in Figure 9, the distribution is broad and there are many outliers in the order of (a) outliers in Period 1, (c) outliers in Period 3, and (b) outliers in Period 2. Second, as shown in Tables 4–6 and Figures 10–15, autoencoder-LSTM is affected more by variance and outliers than LSTM alone. In situations where variance and outliers exist in moderation, the LSTM model using an autoencoder, which can derive the features of inputs accurately, performs better than the model without an autoencoder. Third, among the deep learning methods, the autoencoder-LSTM exhibits the best prediction performance. In Tables 4–6, the results of the autoencoder-LSTM were analyzed to verify that the input data characteristics outperformed those of the general LSTM. Visual graphs of this trend are presented in Figures 10–15.

5. Summary and Concluding Remarks

The goal of this study was to develop a hybrid model based on deep learning models for forecasting FX volatility. In particular, we utilized the three FXVIXs as measures of FX volatility. An FXVIX represents the relationship between the currency of a country and the US dollar. Therefore, this study is meaningful because the FXVIX, which is related to the US and the global economy, sensitively reflects international economic trends.

Data-driven methods are more powerful than model-driven methods for forecasting asset price time-series data (see Kim et al. [77]). In this study, we investigated how event-driven data, which focus on events such as outliers in data-driven analysis, contribute to model performance. According to Shahid et al. [78], events and outliers are different, but outliers can be considered as a type of event. Because there is only one type of outlier in the data considered in this study, comparing differences in model performance accordingly is meaningful.

Our empirical results provide several interesting conclusions with useful practical implications. Our main findings can be summarized as follows. First, the spread of data and presence of outliers increase the accuracy of forecasting performance of the proposed model. Second, improvements in prediction accuracy are more pronounced with autoencoder-LSTM than with LSTM. Finally, for predicting FXVIXs, the autoencoder-LSTM model is superior to the LSTM.

Based on the empirical findings in Section 4, some implications can be observed. First, because the neural network model is a model created by mimicking the human brain, the data to be learned are important. As shown in this study, the forecasting accuracy of the hybrid model is affected by the number of cases for which variability and outliers can be learned. However, extreme outliers in Period 2 degraded the model's performance. Next, the use of an autoencoder, which can transform important properties of input data, similar to principal component analysis, is meaningful. Autoencoders are used for denoising images, watermark removal, dimensionality reduction, and feature variation among other tasks. In this study, we conceived the concept of feature variation. Additionally, several studies using autoencoders to predict time series have been recently published (Gensler et al. [79], Bao et al. [65], and Sagheer and Kotb [47]). Our study contributes to the literature by introducing a new approach called the autoencoder-LSTM for forecasting time series.

In practice, our findings can be helpful to researchers in economic research laboratories or policy managers who determine national economic policies because FXVIXs reveal important trends for FX that impact the global economy and volatility, meaning they can reveal market participant psychology. For example, Menkhoff et al. [33] demonstrated

that global exchange volatility has a significant effect trading strategies based on financial data. Guo et al. [31] confirmed the effects of exchange rate volatility on the stock market. Similar experiments can be considered for future research on different financial indices, such as the S&P 500 and Dow Jones Industrial Average, which are important indices for understanding US and global markets (Ivanov et al. [80] and Liu et al. [81]). The US has the world's largest financial market and plays an important role in determining the trends of the international financial market. Therefore, we expect that predicting these indices will be as meaningful as predicting FXVIXs. Additionally, we expect that we can improve prediction accuracy by learning and incorporating data that can affect each index based on the results of this study, where we only considered FXVIXs. Additionally, hyperparameter optimization was performed using only a grid search, which is a commonly used machine learning algorithm, but we could increase the reliability of prediction by considering additional optimization algorithms.

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 have declared that there are no conflicts of interest.

Acknowledgments

The work of S. Y. Choi was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (no. 2019R1G1A1010278).

References

[1] W. Huang, K. K. Lai, Y. Nakamori, and S. Wang, "Forecasting foreign exchange rates with artificial neural networks: a review," *International Journal of Information Technology & Decision Making*, vol. 3, no. 1, pp. 145–165, 2004.

[2] G. A. Vasilellis and N. Meade, "Forecasting volatility for portfolio selection," *Journal of Business Finance & Accounting*, vol. 23, no. 1, pp. 125–143, 1996.

[3] J. D. Knopf, J. Nam, and J. H. Thornton, "The volatility and price sensitivities of managerial stock option portfolios and corporate hedging," *The Journal of Finance*, vol. 57, no. 2, pp. 801–813, 2002.

[4] C. T. Brownlees and G. M. Gallo, "Comparison of volatility measures: a risk management perspective," *Journal of Financial Econometrics*, vol. 8, no. 1, pp. 29–56, 2010.

[5] G. M. Gallo and E. Otranto, "Forecasting realized volatility with changing average levels," *International Journal of Forecasting*, vol. 31, no. 3, pp. 620–634, 2015.

[6] T. Bollerslev, B. Hood, J. Huss, and L. H. Pedersen, "Risk everywhere: modeling and managing volatility," *The Review of Financial Studies*, vol. 31, no. 7, pp. 2729–2773, 2018.

[7] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[8] D. N. C. Vee, P. N. Gonpot, and S. Noor, "Forecasting volatility of usd/mur exchange rate using a garch (1, 1) model with ged and student'st errors," *University of Mauritius Research Journal*, vol. 17, no. 1, pp. 1–14, 2011.

[9] A. K. Dhamija and V. K. Bhalla, "Financial time series forecasting: comparison of various arch models," *Global Journal of Finance and Management*, vol. 2, no. 1, pp. 159–172, 2010.

[10] D. A. Bala and J. O. Asemota, "Exchange-rates volatility in Nigeria: application of garch models with exogenous break," *CBN Journal of Applied Statistics*, vol. 4, no. 1, pp. 89–116, 2013.

[11] D. S. Kambouroudis, D. G. McMillan, and K. Tsakou, "Forecasting stock return volatility: a comparison of garch, implied volatility, and realized volatility models," *Journal of Futures Markets*, vol. 36, no. 12, pp. 1127–1163, 2016.

[12] G. Köchling, P. Schmidtke, and P. N. Posch, "Volatility forecasting accuracy for bitcoin," *Economics Letters*, vol. 191, Article ID 108836, 2020.

[13] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," 2001.

[14] R. Cont, "Volatility clustering in financial markets: empirical facts and agent-based models," in *Long Memory in Economics*, Springer, Berlin, Germany, 2007.

[15] D. Pradeepkumar and V. Ravi, "Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network," *Applied Soft Computing*, vol. 58, pp. 35–52, 2017.

[16] Y. Liu, "Novel volatility forecasting using deep learning-long short term memory recurrent neural networks," *Expert Systems with Applications*, vol. 132, pp. 99–109, 2019.

[17] E. Ramos-Pérez, P. J. Alonso-González, and J. J. N. Velázquez, "Forecasting volatility with a stacked model based on a hybridized artificial neural network," *Expert Systems with Applications*, vol. 129, pp. 1–9, 2019.

[18] A. Bucci, "Realized volatility forecasting with neural networks," *Journal of Financial Econometrics*, vol. 18, no. 3, pp. 502–531, 2020.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

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

[21] S. Schmidhuber and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] E. Hajizadeh, A. Seifi, M. H. Fazel Zarandi, and I. B. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[23] K. Werner, A. Fadic, and M. C. Minutolo, "Volatility forecast using hybrid neural network models," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2437–2442, 2014.

[24] K. Werner and M. C. Minutolo, "Gold price volatility: a forecasting approach using the artificial neural network–garch model," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7245–7251, 2015.

[25] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating lstm with multiple garch-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[26] A. A. Baffour, J. Feng, and K. Evans, "A hybrid artificial neural network-gjr modeling approach to forecasting currency exchange rate volatility," *Neurocomputing*, vol. 365, pp. 285–301, 2019.

[27] Y. Hu, J. Ni, and L. Wen, "A hybrid deep learning approach by integrating lstm-ann networks with garch model for copper price volatility prediction," *Physica A: Statistical Mechanics and Its Applications*, vol. 557, Article ID 124907, 2020.

[28] M. Ishfaq, B. Q. Zhang, and S. M. R. Shah, "Global macroeconomic announcements and foreign exchange implied volatility," *International Journal of Economics and Financial Issues*, vol. 7, no. 5, p. 119, 2017.

[29] M. F. Dicle and B. Dicle, "Scottish independence referendum: risky or not?" 2017.

[30] P. Keith, "Brexit and its impact on the pound in the foreign exchange market," *The Economists' Voice*, vol. 16, no. 1, 2019.

[31] H. Guo, C. Neely, and J. Higbee, "Foreign exchange volatility is priced in equities," Technical report, Federal Reserve Bank of St. Louis, St. Louis, MO, USA, 2006.

[32] S. Z. S. Abdalla, "Modelling exchange rate volatility using garch models: empirical evidence from arab countries," *International Journal of Economics and Finance*, vol. 4, no. 3, pp. 216–229, 2012.

[33] L. Menkhoff, L. Sarno, M. Schmeling, and A. Schrimpf, "Carry trades and global foreign exchange volatility," *The Journal of Finance*, vol. 67, no. 2, pp. 681–718, 2012.

[34] C. Liu, W. Hou, and D. Liu, "Foreign exchange rates forecasting with convolutional neural network," *Neural Processing Letters*, vol. 46, no. 3, pp. 1095–1119, 2017.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, San Diego, CL, USA, 1985.

[36] P. Qaisar, "Forecasting stock index volatility with garch models: international evidence," *Studies in Economics and Finance*, vol. 32, no. 4, 2015.

[37] D. García and W. Kristjanpoller, "An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models," *Applied Soft Computing*, vol. 74, pp. 466–478, 2019.

[38] S.-Y. Choi and C. Hong, "Relationship between uncertainty in the oil and stock markets before and after the shale gas revolution: evidence from the ovx, vix, and vkospi volatility indices," *PLoS One*, vol. 15, no. 5, Article ID e0232508, 2020.

[39] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[40] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676–682, 2018.

[41] S. Muzaffar and A. Afshari, "Short-term load forecasts using lstm networks," *Energy Procedia*, vol. 158, pp. 2922–2927, 2019.

[42] E. Phaisangittisagul and R. Chongprachawat, "Receptive field resolution analysis in convolutional feature extraction," in *Proceedings of the 2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 485–489, Samui Island, Thailand, September 2013.

[43] M. Zhang, H. Wang, K. Zhou, and P. Cao, "Low probability of intercept radar signal recognition by staked autoencoder and svm," in *Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Zhejiang, China, October 2018.

[44] J. Zeng, J. Wang, L. Guo, G. Fan, K. Zhang, and G. Gui, "Cell scene division and visualization based on autoencoder and k-means algorithm," *IEEE Access*, vol. 7, pp. 165217–165225, 2019.

[45] M. Saha, P. Mitra, and R. S. Nanjundiah, "Autoencoder-based identification of predictors of indian monsoon," *Meteorology and Atmospheric Physics*, vol. 128, no. 5, pp. 613–628, 2016.

[46] S.-X. Lv, L. Peng, and L. Wang, "Stacked autoencoder with echo-state regression for tourism demand forecasting using search query data," *Applied Soft Computing*, vol. 73, pp. 119–133, 2018.

[47] A. Sagheer and M. Kotb, "Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems," *Scientific Reports*, vol. 9, no. 1, pp. 1–16, 2019.

[48] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection," *Transportation Research Part C: Emerging Technologies*, vol. 115, Article ID 102622, 2020.

[49] S. Fu, Y. Li, S. Sun, and H. Li, "Evolutionary support vector machine for rmb exchange rate forecasting," *Physica A: Statistical Mechanics and Its Applications*, vol. 521, pp. 692–704, 2019.

[50] S. Sun, S. Wang, and Y. Wei, "A new ensemble deep learning approach for exchange rates forecasting and trading," *Advanced Engineering Informatics*, vol. 46, Article ID 101160, 2020.

[51] J. Vilasuso, "Forecasting exchange rate volatility," *Economics Letters*, vol. 76, no. 1, pp. 59–64, 2002.

[52] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 74, no. 1, pp. 3–30, 1996.

[53] D. E. Rapach and J. J. K. Strauss, "Structural breaks and garch models of exchange rate volatility," *Journal of Applied Econometrics*, vol. 23, no. 1, pp. 65–90, 2008.

[54] P. Keith and K. N. Langeland, "Forecasting exchange rate volatility: garch models versus implied volatility forecasts," *International Economics and Economic Policy*, vol. 12, no. 1, pp. 127–142, 2015.

[55] Y. You and X. Liu, "Forecasting short-run exchange rate volatility with monetary fundamentals: a garch-midas approach," *Journal of Banking & Finance*, vol. 116, Article ID 105849, 2020.

[56] R. F. Engle, E. Ghysels, and B. Sohn, "Stock market volatility and macroeconomic fundamentals," *Review of Economics and Statistics*, vol. 95, no. 3, pp. 776–797, 2013.

[57] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pp. 348–353, Como, Italy, July 2000.

[58] B. M. Henrique and V. A. Sobreiro, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018.

[59] R. V. Sreelekshmy Selvin, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (Icacci)*, pp. 1643–1647, Udupi, India, September 2017.

[60] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.

[61] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401, Orlando, FL, USA, December 2018.

[62] H. Ohanyan, "Stock price forecast with deep learning lstm and econometric arima models," 2018.

[63] O. S. Deorukhkar, S. H. Lokhande, V. R. Nayak, and A. A. Chougule, "Stock price prediction using combination of lstm neural networks, arima and sentiment analysis," 2019.

[64] X. Liang, Z. Ge, L. Sun, M. He, and H. Chen, "Lstm with wavelet transform based data preprocessing for stock price prediction," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1340174, 17 pages, 2019.

[65] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, vol. 12, no. 7, Article ID e0180944, 2017.

[66] J. Li, G. Liu, H. W. F. Yeung, J. Yin, Y. Y. Chung, and X. Chen, "A novel stacked denoising autoencoder with swarm intelligence optimization for stock index prediction," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, August 2017.

[67] H. Sun, W. Rong, J. Zhang, Q. Liang, and X. Zhang, "Stacked denoising autoencoder based stock market trend prediction via k-nearest neighbour data selection," in *Proceedings of the International Conference on Neural Information Processing*, pp. 882–892, Guangzhou, China, November 2017.

[68] S. Gazioglu, "Stock market returns in an emerging financial market: Turkish case study," *Applied Economics*, vol. 40, no. 11, pp. 1363–1372, 2008.

[69] T. Grammatikos and R. Vermeulen, "Transmission of the financial and sovereign debt crises to the emu: stock prices, cds spreads and exchange rates," *Journal of International Money and Finance*, vol. 31, no. 3, pp. 517–533, 2012.

[70] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Berlin, Germany, 2009.

[71] Y. Gu, B. Wylie, S. Boyte et al., "An optimal sample data usage strategy to minimize overfitting and underfitting effects in regression tree models based on remotely-sensed data," *Remote Sensing*, vol. 8, no. 11, p. 943, 2016.

[72] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 1, p. 91, 2006.

[73] N. Schilling, W. Martin, D. Lucas, and L. Schmidt-Thieme, "Hyperparameter optimization with factorized multilayer perceptrons," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 87–103, Porto, Portugal, September 2015.

[74] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, 2015.

[75] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning svm hyperparameters," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 305–318, 2010.

[76] C. Thornton, F. Hutter, H. Holger, and K. Leyton-Brown, "Auto-weka: combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, Chicago, IL, USA, August 2013.

[77] W. J. Kim, G. Jung, and S.-Y. Choi, "Forecasting Cds term structure based on nelson–siegel model and machine learning," *Complexity*, vol. 2020, Article ID 2518283, 15 pages, 2020.

[78] N. Shahid, I. H. Naqvi, and I. H. Naqvi, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 193–228, 2015.

[79] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002858–002865, Budapest, Hungary, October 2016.

[80] S. I. Ivanov, F. J. Jones, and J. K. Zaima, "Analysis of DJIA, S&P 500, S&P 400, NASDAQ 100 and Russell 2000 ETFs and their influence on price discovery," *Global Finance Journal*, vol. 24, no. 3, pp. 171–187, 2013.

[81] C. Liu, J. Wang, D. Xiao, and Q. Liang, "Forecasting s&p 500 stock index using statistical learning models," *Open Journal of Statistics*, vol. 6, no. 6, pp. 1067–1075, 2016.

Hindawi
Complexity
Volume 2021, Article ID 6618841, 13 pages
https://doi.org/10.1155/2021/6618841

Research Article

Research on Credit Card Default Prediction Based on *k*-Means SMOTE and BP Neural Network

Ying Chen and Ruirui Zhang

School of Business, Sichuan Agricultural University, Chengdu 611830, China

Correspondence should be addressed to Ruirui Zhang; zhangruiruisw@163.com

Received 12 November 2020; Revised 28 January 2021; Accepted 17 February 2021; Published 13 March 2021

Academic Editor: Benjamin Miranda Tabak

Aiming at the problem that the credit card default data of a financial institution is unbalanced, which leads to unsatisfactory prediction results, this paper proposes a prediction model based on *k*-means SMOTE and BP neural network. In this model, *k*-means SMOTE algorithm is used to change the data distribution, and then the importance of data features is calculated by using random forest, and then it is substituted into the initial weights of BP neural network for prediction. The model effectively solves the problem of sample data imbalance. At the same time, this paper constructs five common machine learning models, KNN, logistics, SVM, random forest, and tree, and compares the classification performance of these six prediction models. The experimental results show that the proposed algorithm can greatly improve the prediction performance of the model, making its AUC value from 0.765 to 0.929. Moreover, when the importance of features is taken as the initial weight of BP neural network, the accuracy of model prediction is also slightly improved. In addition, compared with the other five prediction models, the comprehensive prediction effect of BP neural network is better.

1. Introduction

Recently, the state vigorously promotes the economic construction of large- and medium-sized cities, which not only improves people's living standards but also changes people's consumption concept and consumption mode. People are more and more inclined to spend ahead of time and mortgage their "credit" to the bank to enjoy certain things in advance. However, when consuming, people often lack rational thinking and overestimate their ability to repay loans to banks in time. On the one hand, it increases the loan risk of banks; on the other hand, it increases the credit crisis of consumers themselves [1]. With a large number of banks selling credit cards, the phenomenon of credit card default emerges one after another. It is very important for banks to effectively identify high-risk credit card default users. Generally speaking, compared with the credit card customers who have not paid their loans overdue, there are fewer overdue repayments [2, 3]. This variable feature of overdue and overdue loan repayment is called "two classifications" in machine learning prediction. In the prediction of "two classifications," a few categories are called positive examples (default), and most categories are called counterexamples (nondefault). However, most of the credit card loan data are unbalanced. In view of this situation, domestic and overseas scholars have taken up on a large scale a lot of researches. Khoshgoftaar et al. [4] proposed an evolutionary sampling method for unbalanced data, which uses genetic algorithms to selectively delete most types of samples and retain samples with a lot of feature information. Compared with other existing data sampling technologies, evolutionary sampling technology has better performance and is more conducive to empirical replication. The FN undersampling method used by Zhao et al. [5] regarded the minority class as a cluster, which was divided into multiple regions. And they calculated the distance from the negative class samples to the sample mean point in each region, reserving only one sample point in each region. Finally, the remaining negative class samples were used as new negative class samples and the original positive class samples for

training and analysis. Zan et al. [6] used the generative countermeasure network (GAN) to synthesize a few samples to balance the data, then used AdaBoost to change the weight of the input samples, and established a prediction model based on the decision tree classifier. To a certain extent, the recognition rate of unbalanced data was improved. Hu et al. [7] used an improved version of oversampling and undersampling techniques to solve the problem of data imbalance and synthesized the new samples by assigning higher weights to adjacent minority samples through a weight vector. Based on the Euclidean distance standard undersampling most types of samples and keeping the number constant during the resampling process, they found that this method was superior to using a single data sampling technique. Han et al. [8] used an improved version of the smooth algorithm: borderline-smote, which essentially synthesizes new samples from minority samples. However, the original smooth algorithm selects a small number of samples around k nearest neighbors, while scholars use an improved version of the algorithm to find the minority class at the boundary line and use this method to synthesize new samples. Wang et al. [9] constructed a deep learning prediction model for imbalanced data. The model proposed a new loss function on the basis of the original neural network. This method does not need to balance the data in advance. Predictive analysis can be performed directly, and it can effectively reduce the classification error of positive and negative examples. Jiao et al. [10] proposed a reinforcement learning cumulative reward mechanism to improve the attribute selection of the classification regression tree, so as to improve the model's prediction probability for a small number of samples.

We can see that the problem of category imbalance is mainly solved from the following two perspectives: the first perspective is to balance the data by changing the number of samples. This method can also be divided into three aspects. On the one hand, it is to improve the oversampling method. On the other hand, it is based on the principle of undersampling to change the data distribution. On the third hand, it is the method of combining oversampling and undersampling. The second perspective is to improve the classifier algorithm to improve the prediction performance of the model and at the same time use relevant evaluation indicators to evaluate the prediction results. Under normal circumstances, since undersampling will lose information, oversampling is the most widely used technique, and smote is the more common method. However, we have found that most scholars cannot reduce the imbalance between and within the sample categories at the same time when using the improved version of the smooth method, and the applicability of the improved version of the classifier is also limited. Therefore, this paper proposes an improved version of the smooth algorithm with better applicability, which combines the k-means algorithm. This method clusters all samples using the k-means unsupervised learning algorithm, finds clusters with more samples in the minority class, and then uses the smote method that synthesizes new samples in the cluster to change the data distribution. It can not only reduce the imbalance between the categories but also reduce the imbalance within the categories. At the same time, it combines the BP neural network method to predict the credit card default situation to help the bank to identify credit card risks effectively.

2. Basic Theory

2.1. PCA. The main idea of the principal component analysis (PCA) method is to transform the n-dimensional feature variable through the coordinate axis and the origin to form a new m-dimensional feature (usually, m is less than n) [11]. This m-dimensional feature is also called principal component. Its essence is to replace a series of related sample features with newly generated comprehensive features that are irrelevant to each other. When analyzing the data, you can set the cumulative variance ratio determination factor in advance. The working steps of PCA are as follows:

The first step is to standardize the original sample. This step is automatically executed by the software that analyzes the data.

The second step is to determine the correlation between the sample features and calculate the correlation coefficient matrix.

The third step is to determine the number m of principal components after dimensionality reduction, calculate the eigenvalues and their corresponding eigenvectors, and then synthesize these eigenvectors to obtain each principal component.

The fourth step is to determine the comprehensive evaluation index, calculate the information contribution rate of each feature value and principal component, and then weight these values to obtain the final evaluation value.

2.2. Feature Importance Calculation of Random Forest. Random forest is a relatively basic machine learning algorithm, which is widely used in predictive analysis [12], data labeling [13], tag ranking [14], feature importance calculation [15], and other fields. The principle of the algorithm is as follows: using bootstrap method to randomly construct n decision trees, each decision tree is split and pruned and finally combined to form a random forest. In this paper, random forest is used to calculate feature importance, which is used as the initial weight of BP neural network. The basic algorithm steps are as follows:

The first step is to calculate the out-of-bag data error (error1) by using the sample data that has not been selected (out-of-bag data) when drawing samples to construct a decision tree.

The second step is to randomly add noise interference to all the sample features of the data outside the bag and then calculate the error again and record it as error2.

The third step is to calculate the importance of a feature = $\sum_i^n (\text{error2} - \text{error1})/n$ (n is the number of decision trees constructed).

2.3. BP Neural Network. The prediction model used in this paper is the BP neural network algorithm, which is a feed-forward neural network for error backward update. It is often used for bank risk analysis [16], geological disaster monitoring [17], image and handwritten digit recognition [18, 19], and other fields. BP neural network consists of three parts: input layer, middle layer, and output layer. In the model, data samples enter the input layer through a weighted combination of different weights, then pass through the middle layer, and finally get the result from the output layer. Different weights and activation functions make the output of the model very different. In this experiment, the following steps were taken:

The first step is to assign some parameters and initialize some parameters. In the experiment, this paper takes the feature importance calculated by the random forest as the weight of the input layer X_i and sets the same value for the weight of one input variable corresponding to multiple hidden layers. In addition, the number of nodes in the input layer, hidden layer, and output layer is determined.

The second step is to calculate the output of the hidden layer Z_i:

$$Z_i = f\left(\sum_{i=1}^{l} W_{ij}X_i + a_j\right). \tag{1}$$

The third step is to calculate the output layer Y_i:

$$Y_k = \sum_{k}^{n} f_j W_{jk} + b_k. \tag{2}$$

Among them, both *aj* and *bk* in the second and third steps are offset.

The fourth step is to calculate the error E:

$$E = \frac{1}{2}\sum_{k=1}^{s}\left(y_k - Y_k\right)^2. \tag{3}$$

Among them, *yk* is the expected output value, and *Yk* is the actual output value.

The fifth step is to update the weights and biases in reverse.

3. *k*-Means SMOTE Algorithm

We know that smote is a method for synthesizing new samples and solving data imbalance proposed by Chawla et al. [20] and is widely used in various fields. Smote is an improved method of random oversampling technology. It is not a simple random sampling, repeating the original sample, but a new artificial sample generated by a formula. But the smote algorithm will also increase the imbalance between the positive and negative classes of the sample to a certain extent. Therefore, according to the problem of imbalance of credit card sample categories, this paper uses an improved smote algorithm called *k*-means SMOTE algorithm. This algorithm can reduce the imbalance between categories on the one hand and reduce the imbalance within categories on the other hand. In this experiment, we first cluster all samples (30,000), then use *k*-means method to filter clusters with more minority categories, select clusters with more minority categories after filtering, and finally perform smote oversampling in the filtered clusters. The detailed steps of the *k*-means SMOTE algorithm are as follows:

The first step is to randomly select k points among all samples $D = x_1, x_2, x_3, \ldots, x_{30000}$ and use them as the sample cluster centers $C_1, C_2, C_3, \ldots, C_k$.

The second step is to calculate the distance from each sample to the cluster center:

$$d = \sqrt{\sum\left(x_i - C_k\right)^2}. \tag{4}$$

Among them, $x_1, x_2, x_3, \ldots x_i \in D$; $C_1, C_2, C_3, \ldots, C_K \in C$.

The third step is to allocate the sample into the closest clusters:

$$x^i \in C_{\text{nearest}}. \tag{5}$$

The fourth step is to recalculate the cluster center:

$$\mu_i = \frac{1}{|C_i|}\sum_{x\epsilon C_i} x. \tag{6}$$

The fifth step is to repeat the above second, third, and fourth steps until the cluster center no longer changes.

The sixth step is to filter clusters with fewer minority classes and select clusters with more minority classes to synthesize new minority samples.

The seventh step is to perform smote oversampling of *CK* in each filtered cluster:

$$X_{\text{new}} = x_c + \text{rand}(0, 1) \times \left(\widetilde{x} - x_c\right). \tag{7}$$

Among them, rand(0, 1) represents a random number between 0 and 1, X_{new} represents a new synthesized negative class sample, and *xc* represents a negative class randomly selected from m nearest neighbors in the filtered clusters. $\widetilde{x}$ represents the negative samples in the filtered clusters except m neighbors. The *k*-means SMOTE algorithm flow is shown in Figure 1.

4. Experimental Data and Preliminary Analysis

4.1. Preliminary Analysis of Data. This paper uses data on credit card usage, which comes from the kaggle website (https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset). The sample size of this data is 30,000, of which 6,636 are in the positive category (default) and 23,364 in the negative category (no default). The sample has a total of 25 variables. In this experiment, considering that the variable ID has no relationship with the target variable, the deletion process was

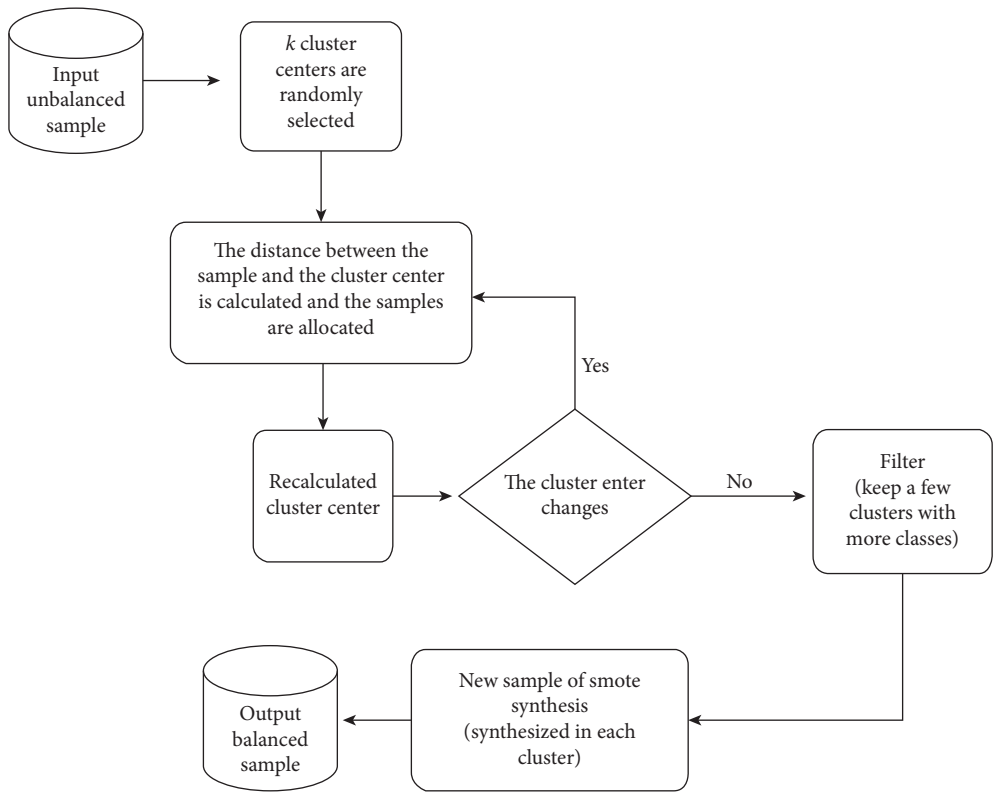


Figure 1: *k*-means SMOTE algorithm flowchart.

performed. 23 characteristic variables and 1 target variable were selected. The variables are shown in Table 1:

Among these 23 features, each feature has been processed accordingly. For the feature limit_bal, we draw a density map according to the default type, and the result is shown in Figure 2.

It can be found from Figure 2 that when the given credit amount is approximately below 150,000, the probability of default is greater than that of nondefault. This shows that when the credit amount is low, there may be more defaulters. For the feature age, we also performed a visual analysis, as shown in Figure 3.

Figure 3 shows that the probability of nondefault of age between approximately 25 and 40 is higher, which indicates that consumers in this age group are more capable of repaying credit card loans. This may be because their work and family tend to be stable without too much pressure. For the feature sex, we draw a stacked histogram according to the target variable, as shown in Figure 4.

As shown in Figure 4, whether it is male or female, the proportion of default consumers is still relatively low, which is in line with the general situation. Conventionally, most of the default data such as credit card fraud are uneven, and we need to make some adjustments to the model based on the actual situation. For the feature education, we find that the feature has six attribute values, and the meanings of the numbers 5 and 6 are unknown, in order to avoid causing a "dimensional disaster" when processing data. We merge them into one meaning (unknown) and draw a stacked histogram to visualize this feature, as shown in Figure 5.

Table 1: Variable attributes.

Number	Variable	Type
1	limit_bal	Continuous
2	Sex	Category
3	Age	Continuous
4	Education	Category
5	Marriage	Category
6	pay_0	Category
7	pay_2	Category
8	pay_3	Category
9	pay_4	Category
10	pay_5	Category
11	pay_6	Category
12	bill_amt1	Continuous
13	bill_amt2	Continuous
14	bill_amt3	Continuous
15	bill_amt4	Continuous
16	bill_amt5	Continuous
17	bill_amt6	Continuous
18	pay_amt1	Continuous
19	pay_amt2	Continuous
20	pay_amt3	Continuous
21	pay_amt4	Continuous
22	pay_amt5	Continuous
23	pay_amt6	Continuous
24	Default.payment.next.month	Category

For the feature marriage, we draw the same graph as the feature sex and education. The default and nondefault conditions of this feature are shown in Figure 6.

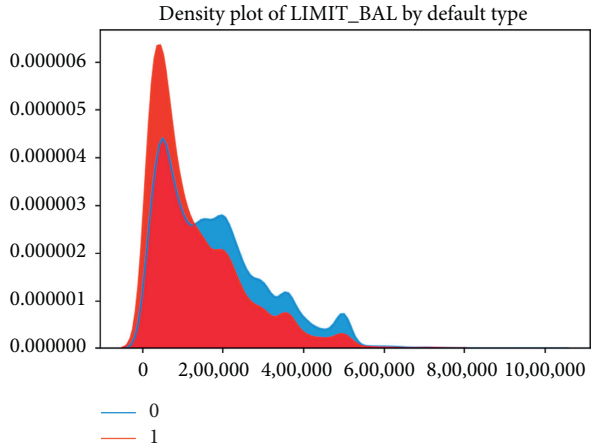


Figure 2: Density diagram of limit_bal.

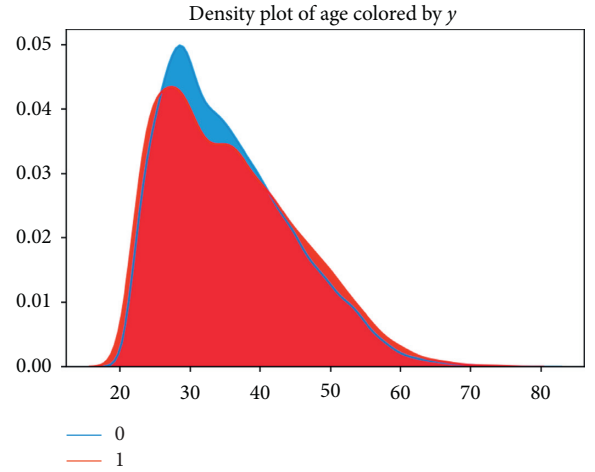


Figure 3: Density diagram of age.

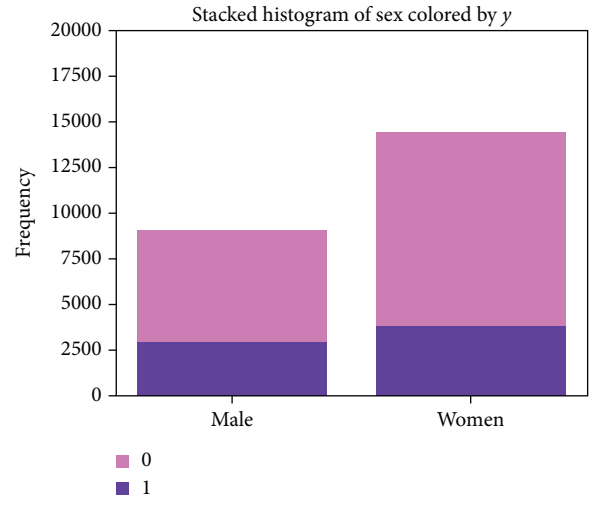


Figure 4: Stacked histogram of gender.

It can be seen from the above three figures that the sample set is unbalanced in the corresponding attribute values of the three characteristics of gender, education, and

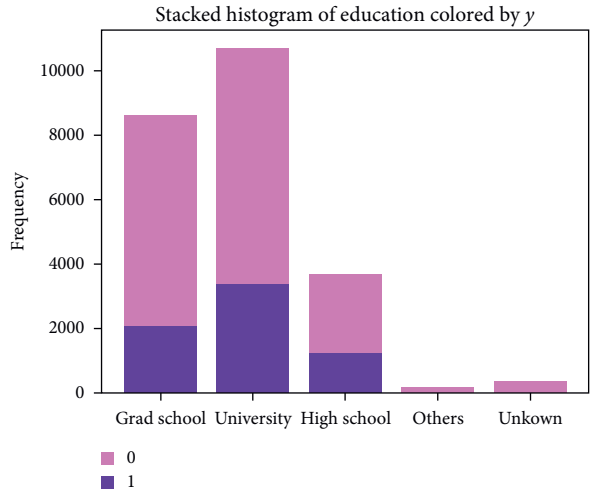


Figure 5: Stacked histogram of education.

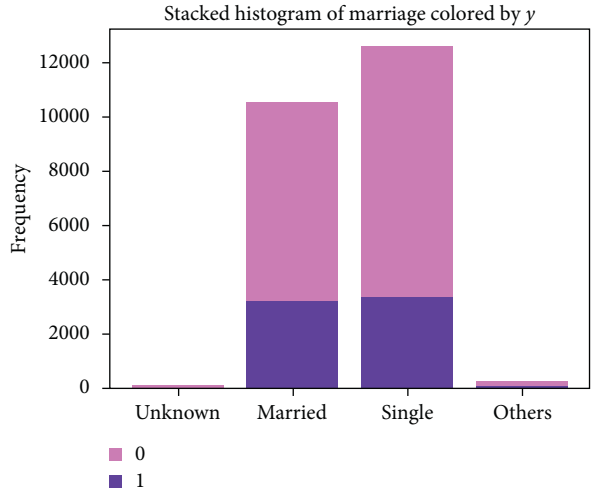


Figure 6: Stacked histogram of marriage.

marriage. For the feature series payment status, we draw different stacked histograms according to different months, and the results are shown in Figure 7.

It can be seen from Figure 7 that consumers who delay payment by one month or less have fewer credit card defaults and almost never happen. In the three months of May, August, and September, for consumers who delayed payment for more than 2 months, the greater the probability of their credit card default is, the more likely it is to increase the loan risk of financial institutions. For the feature series BillAMT and PayAMT, we also perform the corresponding analysis and draw a line graph to visualize the two features, as shown in Figures 8 and 9.

As shown in Figures 8 and 9, due to the imbalance of the data, the line of default only occupies the front part of the figure. Figure 8 shows the amount of the bill, and Figure 9 shows the amount previously paid. Comparing these two images, we find that the six subimages in Figure 9 have greater fluctuations and greater range than the six subimages in Figure 8. Moreover, the uncertainty of the previous

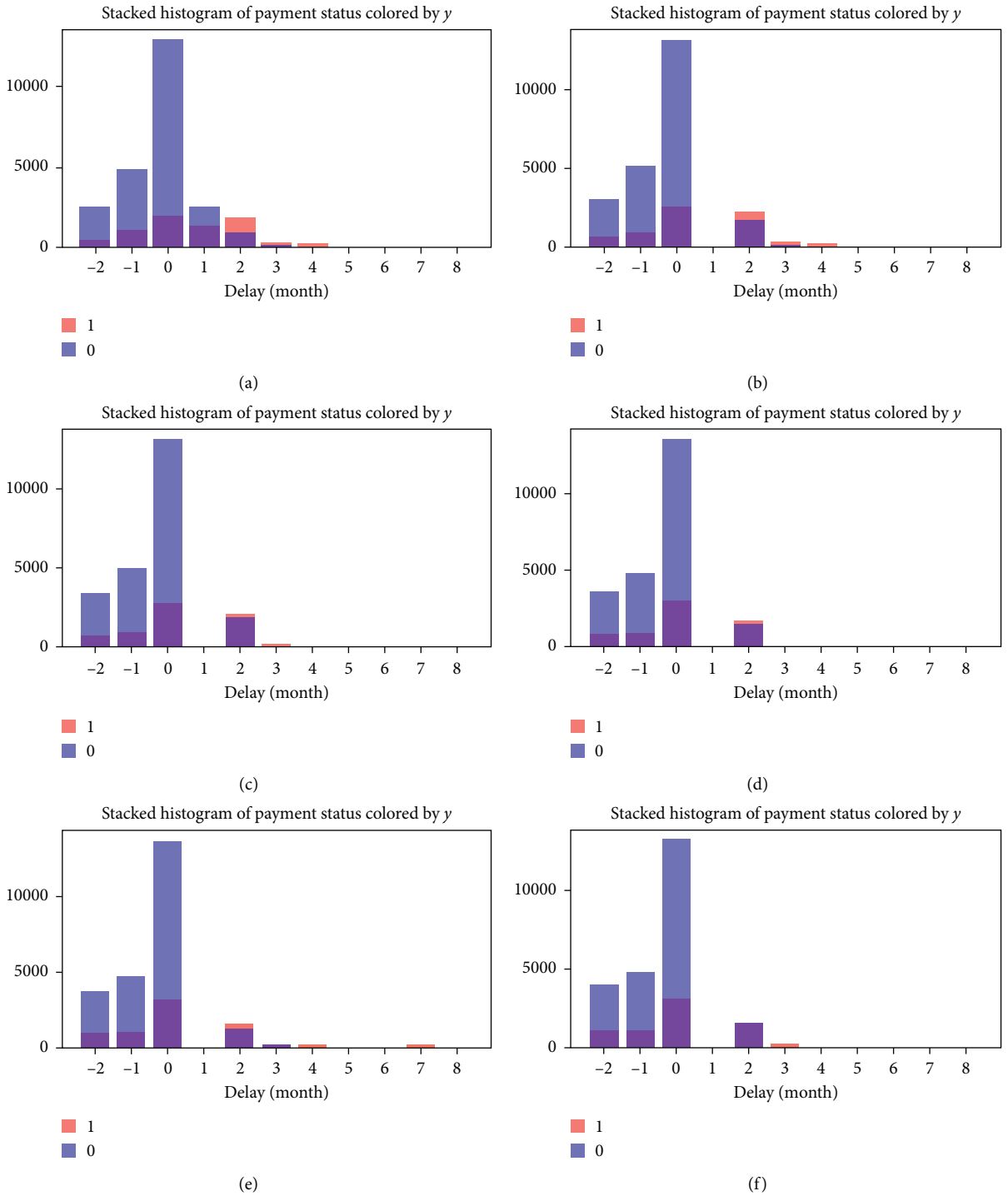


Figure 7: Stacked histogram of payment status. (a) Payment status in September. (b) Payment status in August. (c) Payment status in July. (d) Payment status in June. (e) Payment status in May. (f) Payment status in April.

payment amount has also increased the difficulty for banks to adjust the credit card loan limit.

4.2. Data Processing and Feature Importance. In this experiment, there are a total of 23 features and 1 target variable. After coding and data cleaning, 23 features become 89 input variables. This is a heavy load for model operation and is not conducive to the prediction results of this paper. For comparative analysis with other models, this paper uses PCA for dimensionality reduction, finally obtains 27 input variables, then uses random forest to calculate the importance of these 27 variables, and uses them as the initial weight of the BP neural network. The calculation results of the feature importance are shown in Table 2.

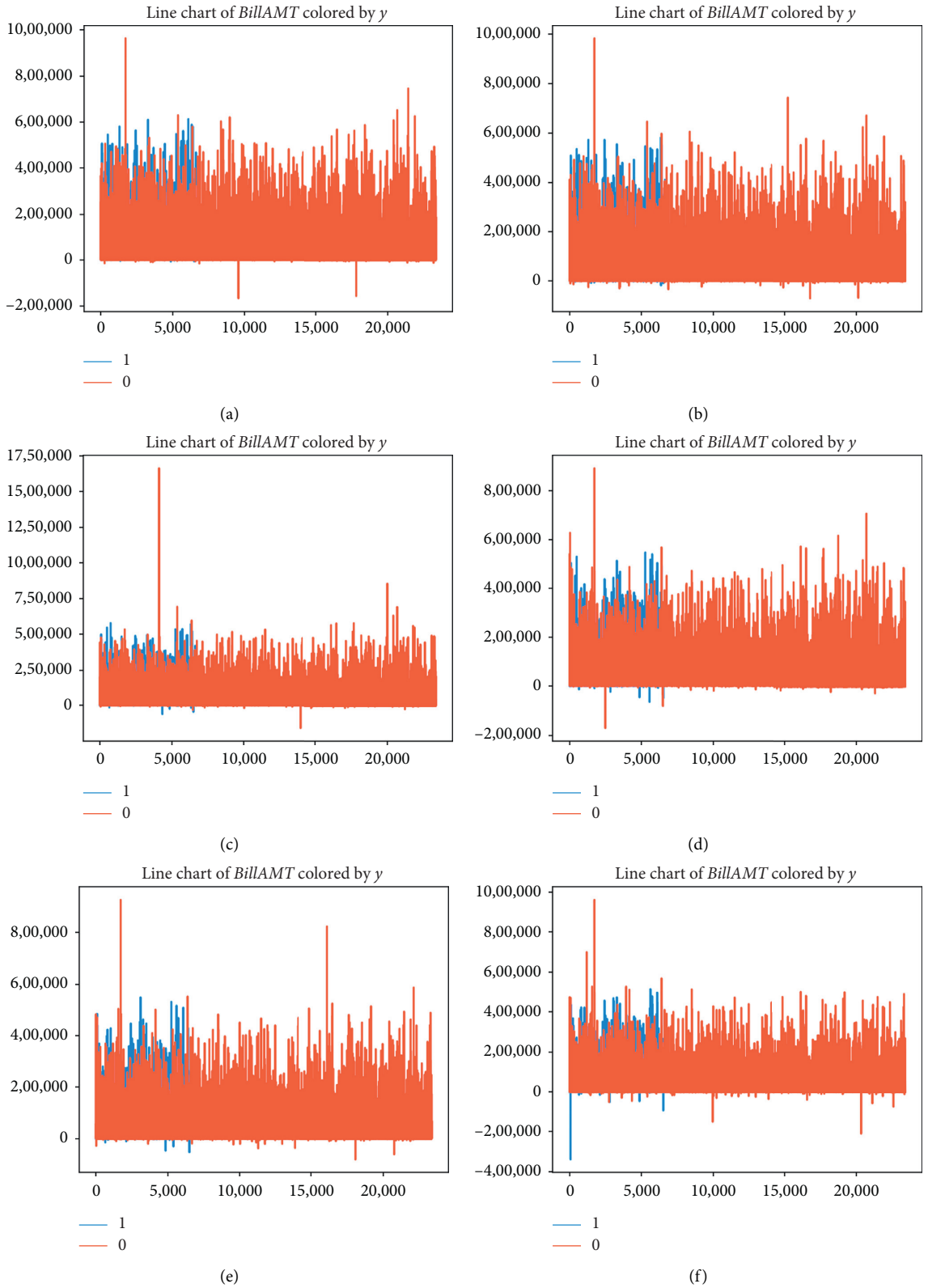


Figure 8: Line chart of billamt. (a) Amount of bill statement in September. (b) Amount of bill statement in August. (c) Amount of bill statement in July. (d) Amountof bill statement in June. (e) Amount of bill statement in May. (f) Amount of bill statement in April.

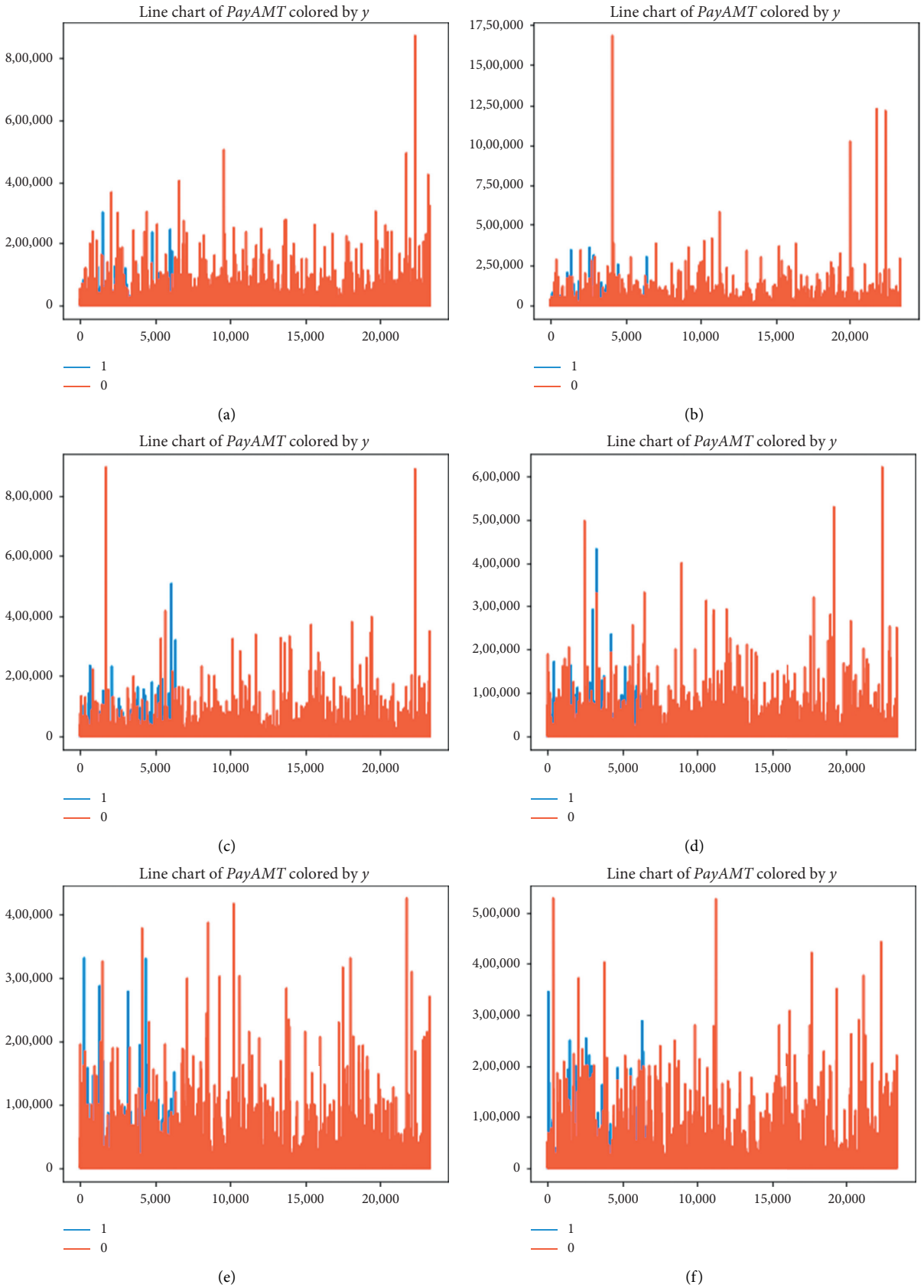


Figure 9: Line chart of payment. (a) Amount of previous payment in September. (b) Amount of previous payment in August. (c) Amount of previous payment in July. (d) Amount of previous payment in June. (e) Amount of previous payment in May. (f) Amount of previous payment in April.

Table 2: Feature importance.

Number	Importance
1	0.06085758
2	0.06335107
3	0.0198413
4	0.13245482
5	0.10467543
6	0.07704057
7	0.02595485
8	0.0214903
9	0.05299085
10	0.02292236
11	0.0207296
12	0.02803694
13	0.02700749
14	0.02271594
15	0.03059199
16	0.02795678
17	0.02408269
18	0.01549738
19	0.01641339
20	0.04804207
21	0.01622095
22	0.0165196
23	0.052694
24	0.02696936
25	0.01606919
26	0.01431037
27	0.01456314

5. Model Prediction and Comparative Analysis

5.1. Model Evaluation Method. According to the actual situation, for unbalanced data, we should use the evaluation index of unbalanced data [21], but because at the beginning of the experiment, we have balanced the number of positive and negative classes in the sample. And we are still using the two-class evaluation indicators commonly used in the past: hybrid matrix, recall, precision, f1-score, AUC value, and so on.

5.2. BP Neural Network Prediction Model. This paper constructs a BP neural network prediction model based on credit card default data. Since this paper has 27 input variables, 55 neurons in the hidden layer, and 2 output layers, the BP neural network model used is shown in Figure 10.

Then, we use the 27 features after principal component dimensionality reduction as input variables $X_1, X_2, \ldots, X_{27}$ and use the feature importance calculated by the random forest as the initial weight of BP neural network. For example, the calculation formula for the weight W of the hidden layer is as follows:

$$\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14}\cdots & W_{155} \\ W_{21} & W_{22} & W_{23} & W_{24}\cdots & W_{255} \\ W_{31} & W_{32} & W_{33}\cdots & W_{34}\cdots & W_{355} \\ W_{271} & W_{272} & W_{273} & W_{274}\cdots & W_{2755} \end{bmatrix} \Rightarrow \begin{bmatrix} 0.06085758 & 0.06085758 & 0.06085758 & 0.06085758\ldots & 0.06085758 \\ 0.06335107 & 0.06335107 & 0.06335107 & 0.06335107\ldots & 0.06335107 \\ 0.0198413 & 0.0198413 & 0.0198413\cdots & 0.0198413\ldots & 0.0198413 \\ 0.01456314 & 0.01456314 & 0.01456314 & 0.01456314\ldots & 0.01456314 \end{bmatrix}. \tag{8}$$

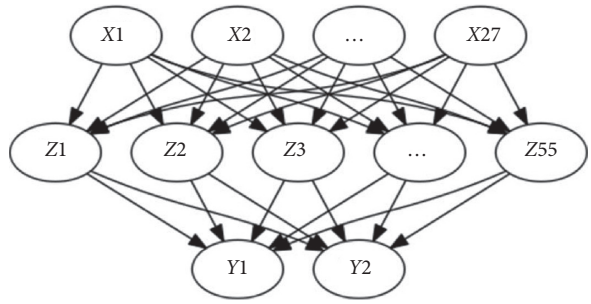


Figure 10: BP neural network model.

In formula (8), there are 27 rows and 55 columns. 27 rows are the number of input variables, and 55 columns are the number of hidden layer neurons. In this experiment, we set each row in the matrix to be the corresponding feature importance (as in the above formula matrix 2) and substitute the result into the model for prediction. We find that when the weights are initialized, the accuracy of the model prediction is 0.8796, and when the feature importance is assigned to the weights, the accuracy of the model prediction is 0.8811. In terms of amount, the accuracy of the second case is slightly higher.

When building the model, we used a three-layer BP neural network to build a credit card default prediction model. The input layer has 27 neurons, the hidden layer has 55 neurons, and the output layer has 2 neurons. The hidden layer is calculated using the following empirical formula:

$$n = 2 \times n1 + 2, \ (n1 \text{ is the number of input layers}). \quad (9)$$

In addition to the initial weight of the hidden layer and the number of neurons in the hidden layer, we have performed a simple process, and the other parameters are default values.

Due to the uneven distribution of the experimental data, we use the k-means SMOTE algorithm to solve this problem. For the parameter k in the k-means SMOTE algorithm, we use the following empirical formula to calculate:

$$k = \sqrt[2]{\frac{N}{2}}, \ (N \text{ is the total number of samples}). \quad (10)$$

Then we substitute the sample size of 30000 (N) into the above formula, can calculate the value of k to be about 122, substitute it into the k-means SMOTE algorithm, and draw the ROC curve graph to intuitively compare the prediction performance of the model before and after k-means SMOTE. And we find that k-means SMOTE greatly improves the prediction performance of the model. The result is shown in Figure 11.

In Figure 11, we find that after the sample is processed by the k-means SMOTE algorithm, the prediction of the model has been greatly improved. The AUC value has been increased from 0.765 to 0.930, the ROC curve of the model is closer to the straight line 1 above the coordinate axis, and the accuracy rate has changed from 0.8252 to 0.8796.

Normally, the BP neural network model with more parameters is prone to overfitting. Because of the high fitting degree of the model, it is possible to learn the noise. We compare the performance of the prediction model in the training set and the testing set, and the results are as follows.

It can be seen from the above table that the values of performance indexes of the prediction model in these two groups of data set have little difference, so we judge that the possibility of overfitting the model in this experiment is relatively low. And the performance of the model can achieve the desired results.

5.3. Comparative Analysis with Other Models. In order to verify the effectiveness of the method used in this experiment, we also establish five other common machine learning models for predictive analysis under the same conditions. We have compared and analyzed the prediction results of these five models in the same situation and used several common performance indicators to evaluate the model. Since the confusion matrix is used to show the prediction results according to different situations, it is not easy to compare the performance of these five models. We adjust it slightly (e.g., the accuracy rate is approximately equal to the average of the accuracy of model positive and negative examples) as shown in Table 3.

It can be seen from Table 4 that the F1 values of these six models have reached above 0.8, indicating that these six models can effectively predict the credit imbalance data in this paper, but the comprehensive prediction performance of the BP neural network is slightly better. The AUC value is the highest among the six models, and the accuracy rate is higher for SVM. But the running time of the SVM model is too long, close to 6 minutes; compared to other models, the running efficiency of SVM is very low. If the amount of data is very large, it is not a wise choice for us to use SVM for prediction. In addition, we can find that except the lower AUC value of the decision tree, the difference in the AUC value of other models is not particularly large. This situation can also be intuitively seen through the ROC curve. The result is shown in Figure 12.

In Figure 12, we can find that if we do not look at the numbers in Table 3, we cannot see the obvious difference in the ROC curves of the first five models from Figure 12. In the above figure, the sixth image is the ROC curve of the decision tree, which is obviously different from the previous five images. This also shows that the tree has the worst performance among the six prediction models.

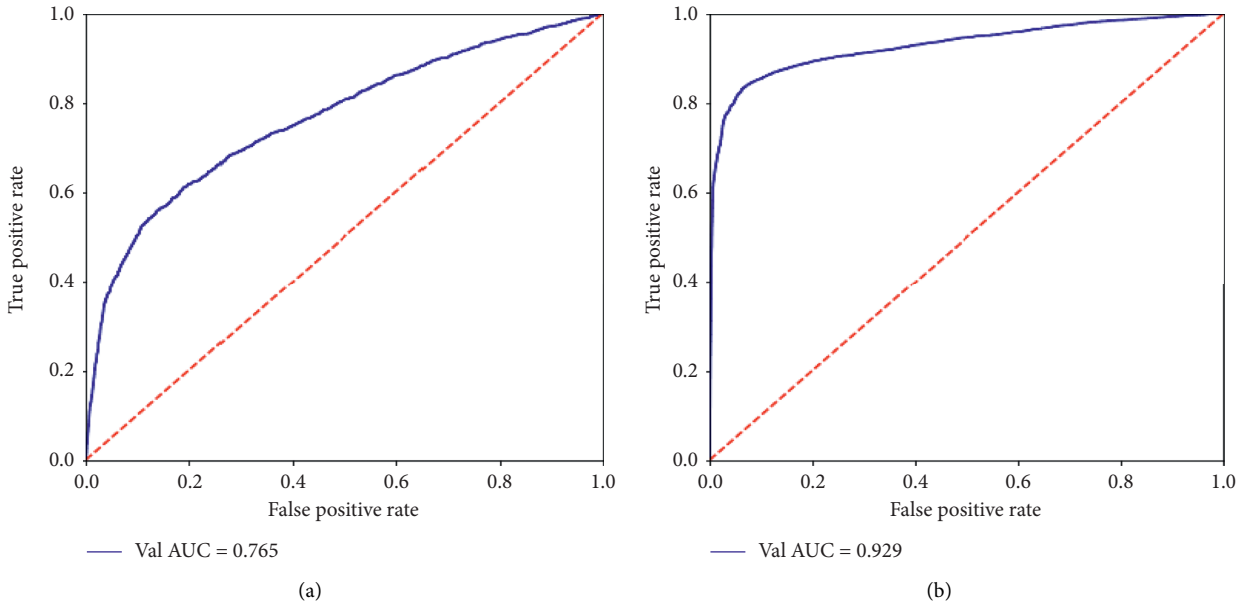


Figure 11: Comparison of ROC curves (a) before and (b) after k-means SMOTE.

Table 3: Comparison results of different data sets.

Data set	Accuracy	Precision	Recall	F1_score
Training set	0.881	0.923	0.840	0.880
Testing set	0.884	0.924	0.837	0.874

Table 4: Comparison results of six models.

Model	AUC	Accuracy	Precision	Recall	F1	Time (s)
BPnn	0.929	0.881	0.923	0.84	0.880	23.89
SVM	0.92	0.884	0.885	0.885	0.885	5 m 55.17
Logistic	0.92	0.876	0.875	0.88	0.877	52
RandomForest	0.92	0.817	0.875	0.875	0.875	35.62
KNN	0.91	0.869	0.87	0.87	0.87	4.75
Tree	0.82	0.871	0.82	0.815	0.817	3.62

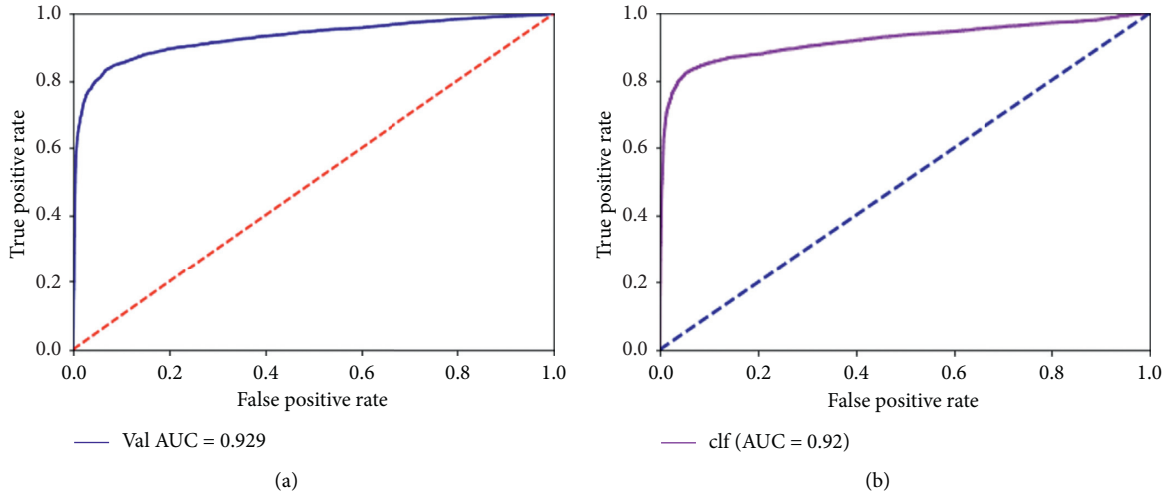


Figure 12: Continued.

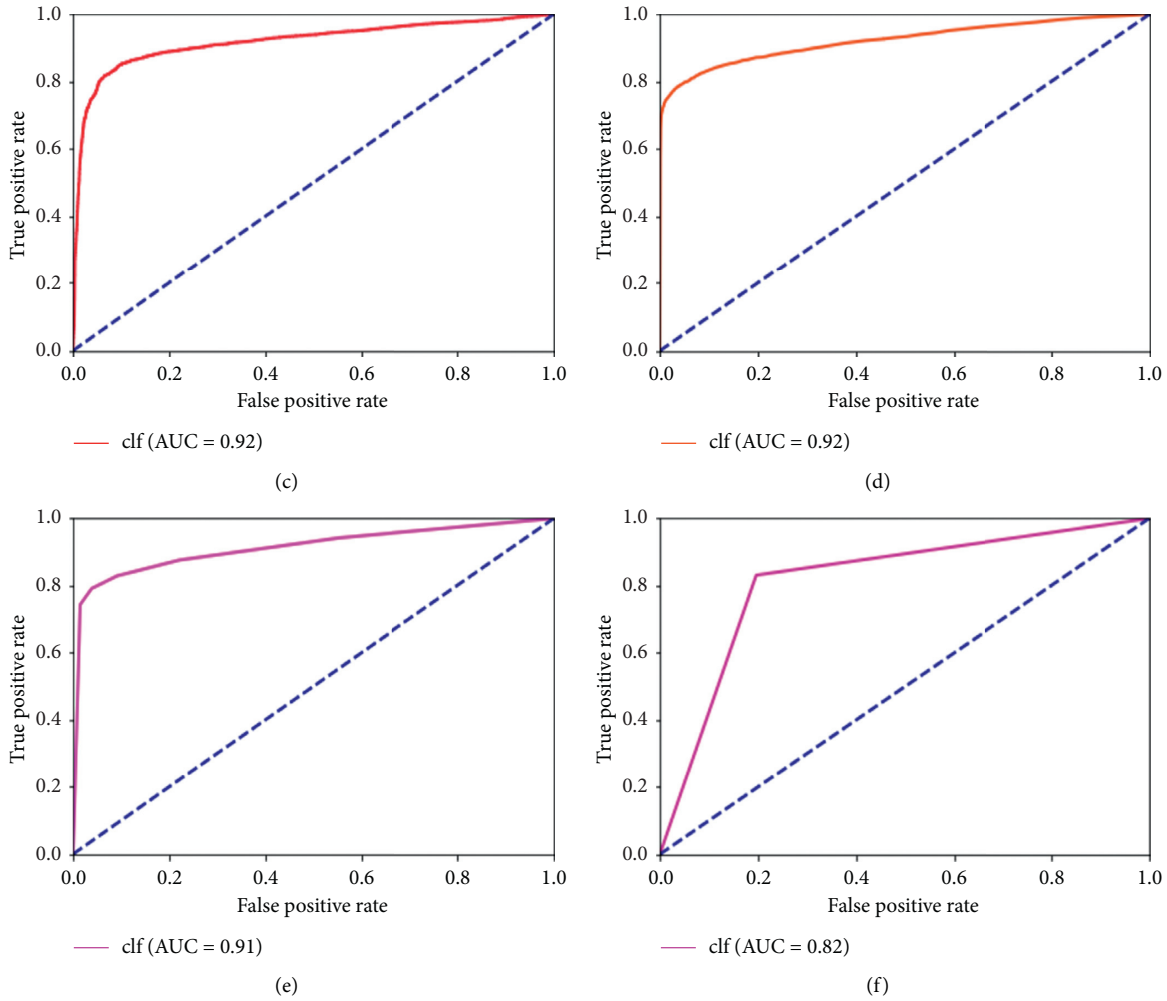


Figure 12: Comparison of six models' ROC curves. (a) BPnn ROC. (b) SVM ROC. (c) Logistic ROC. (d) RandomForest ROC. (e) KNN ROC. (f) Tree ROC.

6. Summary

This paper proposes a comprehensive way by using *k*-means SMOTE and BP neural network algorithms for data imbalance. We find that the improved version of the smote algorithm (*k*-means SMOTE) not only effectively solves the problem of data imbalance but also improves the prediction performance of the model. In addition, we also find that using the feature importance calculated by the random forest as the initial weight of the hidden layer of the BP neural network can slightly improve the prediction performance of the model to a certain extent. However, this change is not obvious. On the one hand, it may be because the credit card default data has many influencing factors and is more complicated. We cannot take all such influencing factors into account, which may indirectly affect the calculation results of feature importance. On the other hand, we think that the amount of sample data may not be enough, the model of BP neural network is relatively simple, and there is no better interpretation of these data for predictive analysis.

In addition, with the gradual increase in the penetration rate of credit cards in our country, the research on its default risk has the following suggestions. On the one hand, we should further improve the construction of the credit indicator system. A good credit index system is conducive to better assessment of personal credit, and a risk prediction model with better classification performance can be established. Specifically, methods such as Delphi expert method, analytic hierarchy process, and regression analysis can be used to find the most representative individual credit indicators, then determine the weight of each indicator, and finally dynamically manage the evaluation system. On the other hand, we should strengthen risk management and control. Since credit card loan default involves personal moral issues, it is highly subjective and uncontrollable. Although major financial institutions are committed to developing the best methods for credit card loan risk avoidance, they have not been able to completely resolve the problem of credit defaults. Therefore, financial institutions should focus on controlling and avoiding risks and try their best to reduce risk losses. Based on the idea of machine learning integration methods, they can comprehensively use each superior classifier to develop a more versatile risk control model.

Data Availability

This paper uses data on credit card usage, which comes from the Kaggle website (https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset).

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] Z. Feng and M. Feng, "Research on credit card scoring model based on AHP," *Finance Theory and Practice*, vol. 1, pp. 74–77, 2016.

[2] R. Mei, Y. Xu, and G. Wang, "Study on analysis and influence factors of credit card default prediction model," *Statistics and Applications*, vol. 5, no. 3, pp. 263–275, 2016.

[3] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020.

[4] T. Khoshgoftaar, N. Seliya, and D. Drown, "Evolutionary data analysis for the class imbalance problem," *Intelligent Data Analysis*, vol. 14, no. 1, pp. 69–88, 2010.

[5] Z. Zhao, G. Wang, and X. Li, "Improved undersampling method for imbalanced data classification based on support vector machine," *Journal of Sun Yat-Sen University (Natural Science Edition)*, vol. 6, pp. 10–16, 2012.

[6] M. Zan, G. Yanrong, and F. Guanlong, "Credit card fraud classification based on GAN-AdaBoost-DT imbalance classification algorithm," *Journal of Computer Applications*, vol. 39, no. 2, pp. 314–318, 2019.

[7] L. Hu, Z. Peng, W. Xiang, and X. Rongze, "A new combination sampling method for imbalanced data," in *Proceedings of the 2013 Chinese Intelligent Automation Conference: Intelligent Information Processing*, vol. 256, pp. 547–554, Yangzhou, China, 2013.

[8] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," *Advances in Intelligent Computing*, vol. 3644, no. 1, pp. 878–887, 2005.

[9] S. Wang, W. Liu, and J. Wu, "Training deep neural networks on imbalanced data sets," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 4368–4374, IEEE, Vancouver, Canada, July 2016.

[10] J. Jiao, X. Zhang, F. Li, and Z. Niu, "Identically distributed multi-decision tree based on reinforcement learning and its application in imbalanced data sets," *Journal of Central South University (Science and Technology)*, vol. 50, no. 5, pp. 1112–1118, 2019.

[11] D. Hong, L. Balzano, and J. A. Fessler, "Asymptotic performance of PCA for high-dimensional heteroscedastic data," *Journal of Multivariate Analysis*, vol. 167, pp. 435–452, 2018.

[12] K. Mens, E. Elzinga, and M. Nielen, "Applying machine learning on health record data from general practitioners to predict suicidality," *Internet Interventions*, vol. 21, Article ID 100337, 2020.

[13] V. A. Sylvester Emma, B. Paul, and R. Bradbury Ian, "Applications of random forest feature selection for fine scale genetic population assignment," *Evolutionary Applications*, vol. 11, no. 2, pp. 153–165, 2018.

[14] Y. Zhou and G. Qiu, "Random forest for label ranking," *Expert Systems with Applications*, vol. 112, pp. 99–109, 2018.

[15] B. Gregorutti, M. Bertrand, and S.-P. Philippe, "Correlation and variable importance in random forests," *Statistics and Computing*, vol. 27, no. 3, pp. 659–678, 2017.

[16] Z. Jin-Hua, "Modeling based on RS and BPNN for credit risk assessment in commercial banks," *Computer Simulation*, vol. 32, pp. 372–379, 2011.

[17] C. D. Li, H. M. Tang, Y. F. Ge, X. L. Hu, and L. Q. Wang, "Application of back-propagation neural network on bank destruction forecasting for accumulative landslides in the three Gorges Reservoir Region, China," *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 6, pp. 1465–1477, 2014.

[18] J. Zhu, A. Wu, X. Wang, and H. Zhang, "Identification of grape diseases using image analysis and BP neural networks," *Multimedia Tools & Applications*, vol. 79, no. 21-22, pp. 14539–14551, 2020.

[19] C. Min-Rong, C. Bi-Peng, Z. Guo-Qiang, L. Kang-Di, and P. Chu, "An adaptive fractional-order BP neural network based on extremal optimization for handwritten digits recognition," *Neurocomputing*, vol. 391, pp. 260–272, 2020.

[20] N. Chawla, K. Bowyer, and L. Hall, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[21] N. Zhao, X. Zhang, and L. Zhang, "Overview of research on unbalanced data classification," *Computer Science*, vol. 45, no. A1, pp. 22–27, 2018.

Hindawi
Complexity
Volume 2021, Article ID 5511802, 13 pages
https://doi.org/10.1155/2021/5511802

Research Article

Forecasting Volatility of Stock Index: Deep Learning Model with Likelihood-Based Loss Function

Fang Jia [1] **and Boli Yang** [2]

[1]*School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*Investment Product Department, Creditease Corp., Beijing 100020, China*

Correspondence should be addressed to Fang Jia; jiafanghust@hust.edu.cn

Received 7 January 2021; Revised 1 February 2021; Accepted 16 February 2021; Published 25 February 2021

Academic Editor: Benjamin Miranda Tabak

Volatility is widely used in different financial areas, and forecasting the volatility of financial assets can be valuable. In this paper, we use deep neural network (DNN) and long short-term memory (LSTM) model to forecast the volatility of stock index. Most related research studies use distance loss function to train the machine learning models, and they gain two disadvantages. The first one is that they introduce errors when using estimated volatility to be the forecasting target, and the second one is that their models cannot be compared to econometric models fairly. To solve these two problems, we further introduce a likelihood-based loss function to train the deep learning models and test all the models by the likelihood of the test sample. The results show that our deep learning models with likelihood-based loss function can forecast volatility more precisely than the econometric model and the deep learning models with distance loss function, and the LSTM model is the better one in the two deep learning models with likelihood-based loss function.

1. Introduction

In finance, volatility refers to the variation degree of asset price, and it measures the uncertainty of the price. It plays an important role in both academic research and financial industry. In risk management and performance measurement, volatility is a risk indicator itself and can be a part of some other indicators, like the Sharpe ratio. In portfolio theory, Markowitz [1] used volatility to measure the risks of assets and the overall risk of the portfolio. Volatility is both the input and the optimization target of the portfolio construction model. In derivative pricing, prices of derivatives can be determined by the volatility of the underlying assets [2].

Since volatility can be useful in different financial areas, and in some situations, the information about future volatility is needed to make financial decisions, it can be valuable to forecast volatility. Econometric methods are widely used to forecast the volatility of financial assets. Engle [3] introduced autoregressive conditional heteroscedasticity (ARCH) model to describe volatility. In this model, the conditional variance is given as a function of the previous variances. The volatility dynamics can be gained by maximizing the likelihood of the model, and then the estimated model can be used to forecast future volatility. Bollerslev [4] extended the ARCH model to be a generalized autoregressive conditional heteroscedasticity (GARCH) model. This model makes conditional variance to be a function of previous errors and previous variances. Nelson [5] further extended the GARCH model and built exponential generalized autoregressive conditional heteroscedasticity (EGARCH) model. An asymmetric response to shocks is allowed in this model. Corsi [6] used intraday high-frequency data to calculate realized volatility and introduced a heterogeneous autoregressive (HAR) model to forecast the volatility.

Using machine learning algorithms is another way to forecast volatility. Compared to econometric models which are based on economic assumptions and statistical logic, machine learning algorithms are more data-driven. A large

number of papers combined neural networks and the GARCH model to be a hybrid model and used the hybrid model to forecast volatility [7–15]. These papers used this type of method and studied different kinds of assets, including stock indices in several nations, metals, oil, and bitcoin. They all concluded that the hybrid neural networks can forecast volatility precisely. Some papers made other kinds of combinations and gained successful predictions. Lahmiri [16] combined neural networks and technical indicators to forecast the volatility of exchange rates. Peng et al. [17] combined support vector regression and GARCH models to forecast volatility of currencies. Ramos-Pérez et al. [18] use a set of machine learning techniques, such as gradient descent boosting, random forest, support vector machine, and neural network, and stack them to forecast volatility of S&P 500.

Some papers used deep learning, which is a special branch of machine learning, to forecast volatility. Since LSTM is an effective machine learning architecture to model time series, some papers combined LSTM and GARCH model to be hybrid model and used the hybrid model to forecast volatility [19–21]. They compared their models with some other models, such as support vector regression (SVR) and GARCH. The hybrid deep learning models showed strong forecasting ability. Xing et al. [22] used text mining to capture the market sentiment from social message streams and incorporated the sentiment signals into the recurrent neural networks (RNNs). Their model defeated nine other models on volatility forecasting. Vidal and Kristjanpoller [23] combined two popular deep learning architectures, which are convolution neural network (CNN) and LSTM, to forecast the volatility of gold. Yu and Li [24] defined extreme value volatility for stock index, which is calculated from daily highest prices and daily lowest prices. They used historical volatilities to forecast future volatility, by LSTM and GARCH separately.

Almost all these machine learning references follow a similar process to forecast volatility. They use historical data as input to forecast volatility for each day, and they use statistical methods to estimate the realized volatility for the same day. Then, they use the distance between the two, for example, mean squared error (MSE) or mean absolute error (MAE), as the loss function, to train the machine learning model. By minimizing the distance function, they can gain effective machine learning models to forecast realized volatility.

This is a common process when researchers try to predict a target by machine learning, but it gains two disadvantages if they follow it to forecast volatility. First, because volatility is unobservable, when researchers use estimated realized volatility as the target, they introduce errors into the forecasting process. These kinds of errors would decrease the accuracy of their prediction because what they try to forecast is not real volatility. Second, econometric models estimate their parameters by maximizing the likelihood of sample data, which is different from the optimization target of these machine learning models. Since these researchers always use distance value to further test the forecasting ability, the training method of the econometric model is not consistent with their testing method. So the comparison between the deep learning model and the econometric model is not fair enough in these papers.

To solve these two problems, we introduce a negative log-likelihood function to be the loss function of the deep learning model in this paper. By using this likelihood-based loss function, we can train the deep learning model without estimating the realized volatility, so the forecasting process can be more straightforward. At the same time, the deep learning model can be compared to the econometric model more fairly, since the training methods of the deep learning model and econometric model are both based on the likelihood function. For consistency, we test the models also by calculating the likelihood function of the test sample.

In addition, the deep learning model we build only uses index returns as the inputs to forecast volatility, including no economic intuition in the model. This is an end-to-end deep learning method since we go straight from the historical return series to volatility prediction. The deep learning model and econometric model use the same inputs to forecast volatility, so the comparison between them is fair enough.

The rest of this paper is organized as follows. Section 2 describes the data and methodology we use. The empirical study and sensitivity analysis are discussed in Section 3. Section 4 makes the conclusion.

2. Materials and Methods

2.1. Data. This research studies three major indices of the US stock market, which are S&P 500 Index, Dow Jones Industrial Average Index, and NASDAQ Composite Index. We obtain closing prices of these stock indices from their start dates to June 30, 2020. In detail, S&P 500 sample data cover from January 2, 1928, to June 30, 2020, Dow Jones sample data cover from May 26, 1896, to June 30, 2020, and NASDAQ sample data are from February 5, 1971, to June 30, 2020. The three sample periods include 23240, 31096, and 12457 trading days separately.

Then we calculate daily returns as the logarithms of relative daily closing prices, using the following equation:

$$r_t = \log\left(P_t/P_{t-1}\right), \tag{1}$$

where r_t is the daily return at time t and P_t is the closing price at time t.

Table 1 shows the summary statistics of the three return series. All three indices gain positive average return within the sample period. S&P 500 and Dow Jones experienced more serious single-day loss than NASDAQ. Standard deviations of the three return series are similar. All the return series show negative skewness and high kurtosis. It can be confirmed from the Jarque–Bera test that these return series are not from normal distribution.

2.2. Deep Neural Network. Artificial neural network (ANN) is one of the best known machine learning algorithms. It is designed to imitate the structure of neurons in the human brain. Artificial neurons are connected to each other in this

Table 1: Descriptive statistics of daily returns of stock indices.

	S&P 500	Dow Jones	NASDAQ
Mean	0.022%	0.021%	0.037%
Maximum	15.36%	14.27%	13.25%
Minimum	−22.90%	−25.63%	−13.15%
Std. dev.	1.19%	1.16%	1.25%
Skewness	−0.47	−0.86	−0.38
Kurtosis	22.12	27.72	13.57
Jarque–Bera test	0.00	0.00	0.00
Observations	23239	31095	12456

model, and the networks can acquire a problem-solving ability by adjusting their connection weights through learning.

Deep neural network is ANN with a certain level of complexity. Under general definition, DNN is a neural network with two or more hidden layers. Figure 1 shows the typical architecture of DNN. Inputs are imported into the model through the input layer. Hidden layers and output layer are calculated sequentially by multiplying the previous layer with the connection weights, and the activation function is applied each time.

To make a DNN model work, we need to train it. By using the predicted value that we gain from the output layer and the target value, we can form a loss function. The connection weights in DNN are learned by minimizing the loss function. We mostly use forward and back propagation to deal with the calculation process when optimizing the networks.

2.3. LSTM. LSTM is a particular type of deep neural network developed by Hochreiter and Schmidhuber [25]. By inheriting the benefits from the recurrent neural network (RNN), LSTM has shown great power to model sequential data, like time series. Compared to vanilla RNN, LSTM uses gates to control information flows through the sequence so that it can be capable of learning long-term dependencies and solving the vanishing gradient problem.

Figure 2 shows the typical architecture of LSTM. Like vanilla RNN, inputs are imported into the model at each time step, and outputs can be given at each time step. Especially, LSTM uses memory block to take place of neuron in RNN. A memory block is composed of a memory cell, an input gate, a forget gate, and an output gate.

We can use vector formulas to describe LSTM, as follows:

$$\begin{aligned} X &= \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}, \\ f_t &= \sigma\left(W_f X + B_f\right), \\ i_t &= \sigma\left(W_i X + B_i\right), \\ o_t &= \sigma\left(W_o X + B_o\right), \\ c_t &= f_t \nabla c_{t-1} + i_t \nabla \tanh\left(W_c X + B_c\right), \\ h_t &= o_t \nabla \tanh\left(c_t\right), \end{aligned} \tag{2}$$

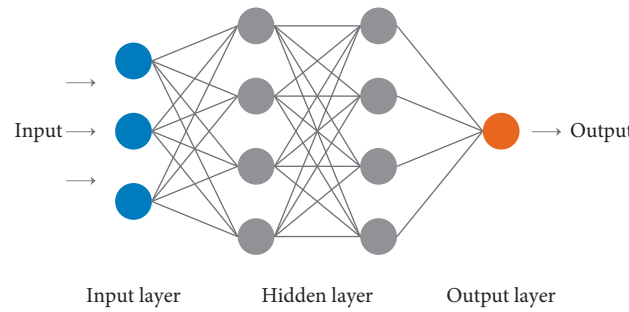


Figure 1: Architecture of DNN.

where h_t is the hidden state at time t and x_t represents the inputs at time t. σ is activation function where logistic sigmoid is popularly used. f_t describes the forget gate, i_t describes the input gate, and o_t describes the output gate. ∇ denotes point-wise multiplication.

2.4. Econometric Model. ARMA-GARCH is the most widely used econometric model when studying volatility. We will use this model to forecast volatility and compare it with deep learning models in this paper. Bollerslev [4] built the GARCH model by generalizing the ARCH model. A general ARMA(m, n)-GARCH(p, q) model is as follows. Firstly, the return is decomposed into AR effects and MA effects:

$$r_t = \mu + \sum_{i=1}^{m} \theta_i \left(r_{t-i} - \mu\right) + \sum_{j=1}^{n} \gamma_j \varepsilon_{t-j} + \varepsilon_t, \tag{3}$$

where μ is the mean of r_t and ε_t is the error term.

It is assumed that ε_t can be given as follows:

$$\varepsilon_t = z_t h_t^{1/2}, \tag{4}$$

where $z_t \sim N(0, 1)$, so h_t is the conditional variance at time t.

Then, the conditional variance is assumed to be a linear function of the errors and its own lags:

$$h_t = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j}. \tag{5}$$

2.5. Simple Historical Statistics as Benchmark. Besides deep learning models and ARMA-GARCH, we further build a simple method to forecast volatility. Similar to Markowitz [1] which used historical statistics to be the expected risk, we calculate the standard deviation of the return series in an n-trading-day window and use it as the prediction of volatility for the next trading day. This is a simple but intuitive method, and we can use it as a benchmark to evaluate other methods.

The parameter which needs to be estimated is just the window length n. We also use the optimization method similar to econometric models to gain the estimation, which means that the best window length is the value that can maximize the likelihood of sample data.

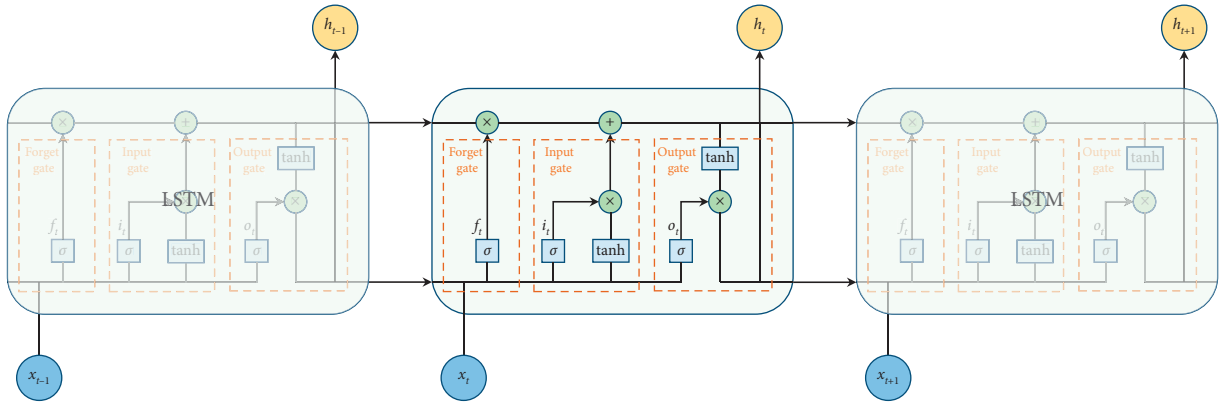


Figure 2: Architecture of LSTM network.

2.6. Experimental Setup of Our Deep Learning Models. As discussed in the Introduction, we use a likelihood-based loss function when training our DNN and LSTM models. If r_t is the return series that we observe from time 1 to time T and σ_t is the volatility that our models forecast at the same time step, we can calculate the sample likelihood under the assumption of normal distribution:

$$L = \prod_{i=1}^{T} \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right). \tag{6}$$

Then, we can have the log-likelihood function:

$$\log L = \sum_{i=1}^{T} \left(-\frac{1}{2}\log(2\pi) - \log\ \sigma_t - \frac{r_t^2}{2\sigma_t^2}\right). \tag{7}$$

The log-likelihood function should be maximized through optimization, while the deep learning models are always trained by minimizing their loss function. So we need to use the negative log likelihood to be the loss function in deep learning models. To further reduce the computational cost of deep learning models, we use a simplified function as the loss function of our DNN and LSTM models:

$$\text{loss} = \sum_{i=1}^{T} \left(2\ \log\ \sigma_t + \frac{r_t^2}{\sigma_t^2}\right). \tag{8}$$

When we test all the models we discuss in this paper, we will use equation (7) to calculate the log-likelihood function of the test sample and compare the values.

No matter in the training process or in the testing process of all the methods, the forecasting procedure is implemented day by day. It means that we forecast the volatility in a rolling window, and the window moves forward one trading day at each time. After we forecast the volatility day by day and get the volatility series σ_t from time 1 to time T, we can use it to calculate equations (7) and (8) in different processes.

The other settings of our DNN model are as follows. The unit number of the input layer is set to be 10, which means that we use 10-day return series as input each time. The number of hidden layers is two. The first hidden layer has 40 units, and the second hidden layer has 80 units. The activation function of hidden layers is chosen to be ReLU, and the dropout of these two layers is set to be 0.3. The activation function of the output layer is set to be sigmoid. RMSprop is used as the optimizer to train the model. The batch size is set to be 2048. There will be early stopping if the loss function of the validation set does not go down anymore.

Our LSTM model combines LSTM with a fully connected layer in the last time step. Its architecture is shown in Figure 3. The length of the input series is set to be 10. The unit number is 20 at each layer of LSTM. The fully connected layer has 40 units, and its activation function is chosen to be ReLU. For the fully connected layer, we set the dropout to be 0.5, while there is no dropout in the LSTM layer. The activation function of the output layer is set to be sigmoid. RMSprop is used as the optimizer to train the model. The batch size is set to be 2048. There will be early stopping if the loss function of the validation set does not go down anymore.

It is noteworthy that the activation function of the output layer in our two deep learning models is an important setting, when we introduce the likelihood-based loss function. If the loss function is the distance function that other research studies use, linear activation function can be good enough. But when we use the likelihood-based loss function as equation (8), some activation functions like linear function cannot help the models to converge when training. Sigmoid activation function is a good choice in the output layer, and our models can be trained successfully.

All the deep learning models in this paper, which include the DNN model and the LSTM model that we have mentioned above, and two more models with different loss functions that are used for comparison are implemented by the Keras framework.

For DNN and LSTM models, we divide the full sample into three parts: train set (70%), validation set (15%), and test set (15%). ARMA-GARCH model and the simple method do not need validation when training, so we combine the train set and validation set for these two methods and estimate their parameters on this full train set (85%). Train-test split for the ARMA-GARCH and simple method and train-validation-test split for the deep learning models are both in

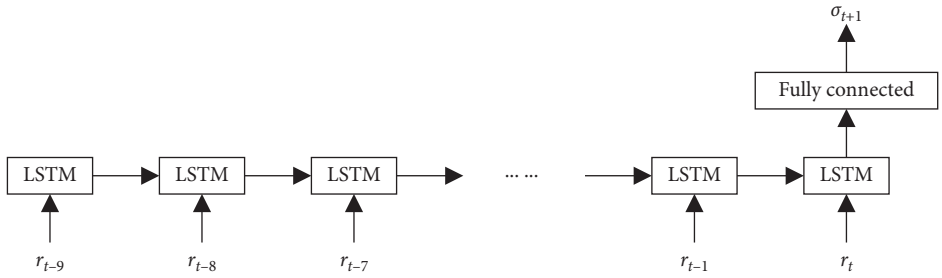


Figure 3: Architecture of our LSTM model.

time order. They share the same test set, so they can be compared directly, by comparing the log-likelihood values of this test sample.

2.7. Experimental Setup of Models for Comparison. As we have discussed, in almost all the reference papers which use machine learning to forecast volatility, the distance between estimated realized volatility and forecasted volatility is used as the loss function to train the models. To compare this kind of common method to our deep learning models with likelihood-based loss function, we further test DNN and LSTM models with distance loss function like them. By doing so, we can also study whether the deep learning models can still have good forecasting ability, when we train them by minimizing the distance loss function and test them by calculating the likelihood value of the test sample. Since their training method is not consistent with their testing method, deep learning models with distance loss function are in a disadvantageous position when we compare them with other models in this paper.

The distance loss function of DNN and LSTM is chosen to be MSE. This loss function is the most chosen one of the related researches. Similar to these researches, the realized volatility is calculated by the standard deviation of the return series in a 21-trading-day window, which is close to the average number of trading days in a month. The activation function of the output layer is set to be the linear function, which is also a common choice of deep learning models with distance loss function, when forecasting volatility. All the other settings are the same as our deep learning models with likelihood-based loss function.

When training ARMA(m, n)-GARCH(p, q) model, we set m, n, p, and q to be 1 to 3 and estimate all different combinations. The model with minimum BIC is chosen and will be used for prediction on the test set. Simple method just needs to choose the best window length on the train set, as we have discussed in the previous section.

3. Results and Discussion

3.1. Empirical Results. When we train the LSTM model with likelihood-based function, we record the loss function values of train samples and validation samples at each epoch and show the learning curves in Figure 4. For S&P 500, the model gains minimum validation loss function at epoch 723, and the value of validation loss function is −15241.51 at this epoch. For Dow Jones, the model gains minimum validation loss function at epoch 568, and the value of validation loss function is −15755.87 at this epoch. For NASDAQ, the model gains minimum validation loss function at epoch 1212, and the value of validation loss function is −14600.25 at this epoch.

When we train the DNN model with likelihood-based loss function, we also record the loss function values of train samples and validation samples at each epoch and draw the learning curves as in Figure 5. For S&P 500, the model gains minimum validation loss function at epoch 1318, and the value of validation loss function is −15197.80 at this epoch. For Dow Jones, the model gains minimum validation loss function at epoch 622, and the value of validation loss function is −15735.40 at this epoch. For NASDAQ, the model gains minimum validation loss function at epoch 688, and the value of validation loss function is −14586.70 at this epoch.

When we train the LSTM model with MSE loss function, the optimization process stops at epoch 90 for S&P 500, at epoch 422 for Dow Jones, and at epoch 70 for NASDAQ. When we train the DNN model with MSE loss function, the optimization process stops at epoch 271 for S&P 500, at epoch 309 for Dow Jones, and at epoch 174 for NASDAQ. It is obvious that the deep learning models with MSE loss function converge faster than the deep learning models with likelihood-based loss function, under the same learning rate.

Then we try to find the suitable ARMA(m, n)-GARCH(p, q) models for each index. After we estimate all ARMA-GARCH models with m, n, p, and q from 1 to 3, we choose the best one with minimum BIC. For S&P 500, it is ARMA(1, 1)-GARCH(1, 2), for Dow Jones, it is ARMA(1, 1)-GARCH(1, 2), and for NASDAQ, it is ARMA(1, 2)-GARCH(1, 1). Table 2 shows the chosen models with their BIC values for each index.

Table 3 shows the parameter estimation results of the chosen ARMA-GARCH models for each index. For S&P 500 and Dow Jones, the AR effect is negative and the MA effect is positive. For NASDAQ, the AR effect is positive and the MA effect is negative. The parameters of ARCH term are all around 0.11 for three indices. The sum of ARCH and GARCH parameters is near 1 for each index. It can be concluded from the t-tests that almost all the parameters are significant at 5% level. Only ω of ARMA(1, 2)-GARCH(1, 1) for NASDAQ is significant at 10% level.

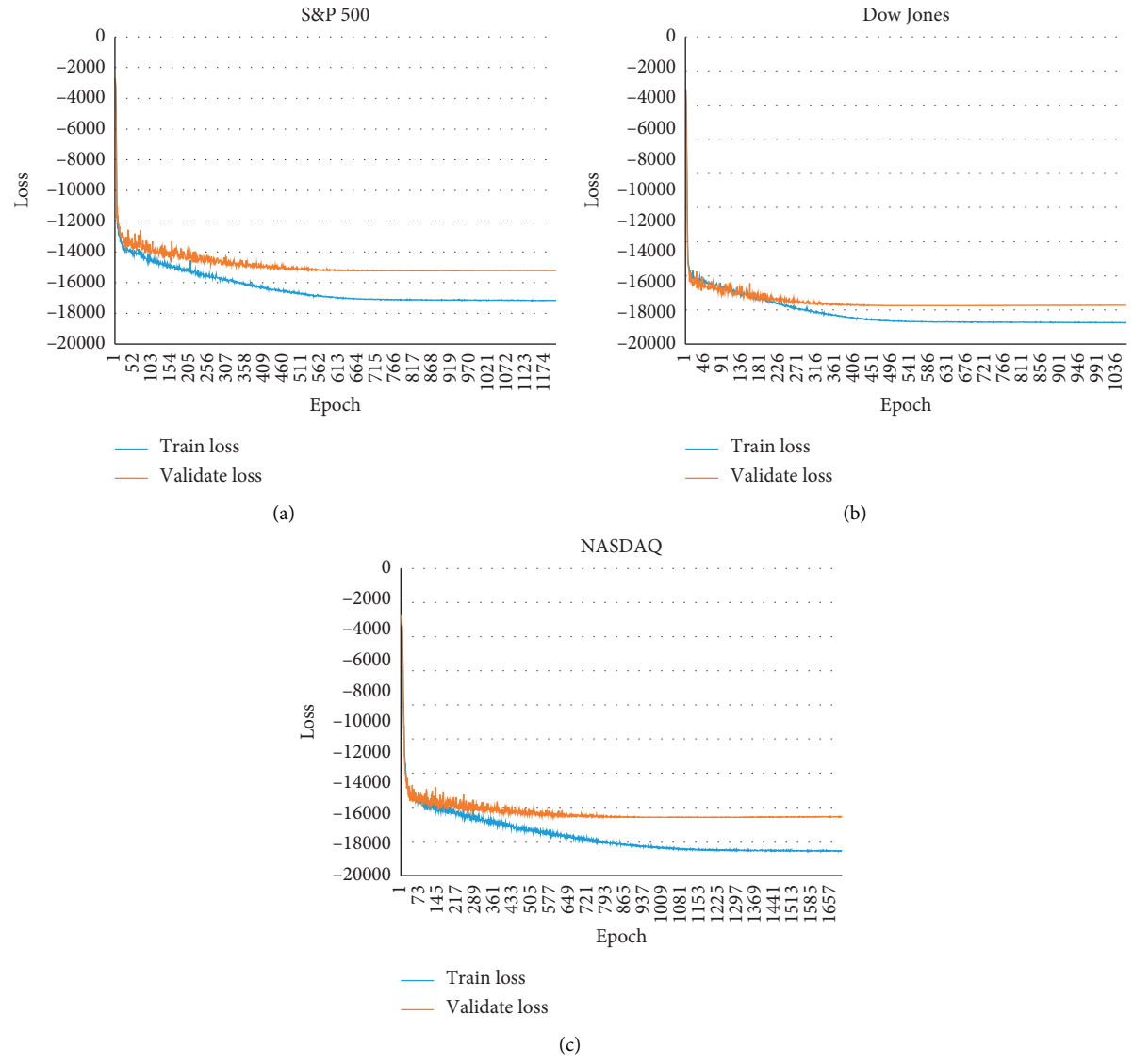


Figure 4: The learning curves when training LSTM (likelihood-based loss).

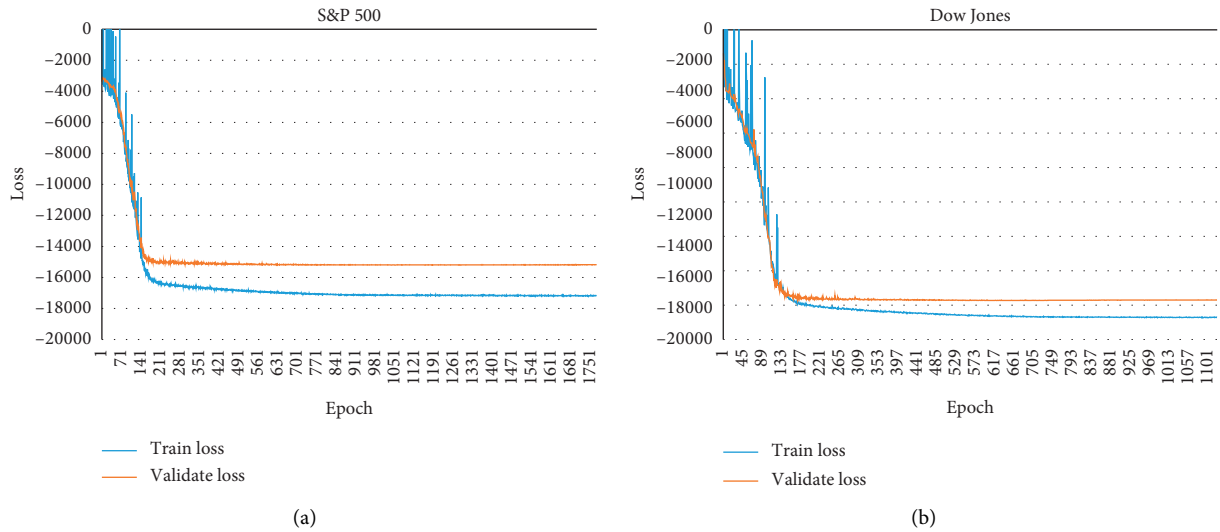
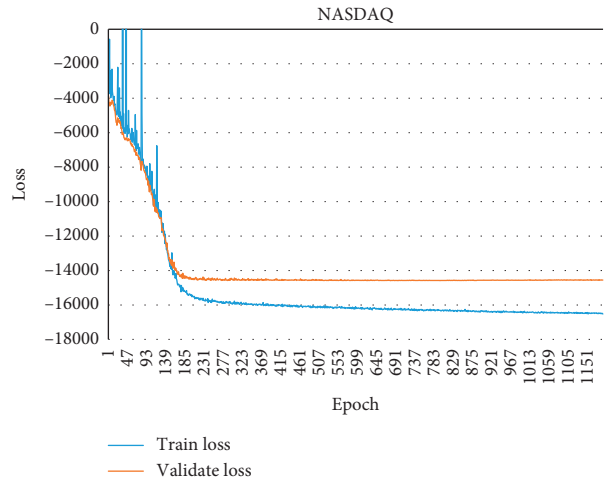


Figure 5: Continued.



(c)

Figure 5: The learning curves when training DNN (likelihood-based loss).

Table 2: ARMA-GARCH model with minimum BIC.

	S&P 500	Dow Jones	NASDAQ
Best model	ARMA(1, 1)-GARCH(1, 2)	ARMA(1, 1)-GARCH(1, 2)	ARMA(1, 2)-GARCH(1, 1)
BIC	−6.6148	−6.5230	−6.5385

Table 3: ARMA-GARCH estimation results.

	S&P 500		Dow Jones		NASDAQ	
	Estimate	T-test	Estimate	T-test	Estimate	T-test
μ	$4.59e-4$	8.10^{***}	$4.45e-4$	8.52^{***}	$6.20e-4$	6.16^{***}
θ_1	−0.150	-2.41^{**}	−0.155	-2.02^{**}	0.866	11.50^{***}
γ_1	0.259	4.26^{***}	0.235	3.12^{***}	−0.668	-8.65^{***}
γ_2					−0.147	-6.43^{***}
ω	$1.23e-6$	3.07^{***}	$2.10e-6$	2.57^{**}	$1.03e-6$	1.73^{*}
α_1	0.115	15.96^{***}	0.113	9.38^{***}	0.103	9.40^{***}
β_1	0.504	59.49^{***}	0.543	6.73^{***}	0.892	86.09^{***}
β_2	0.379	15.04^{***}	0.336	5.05^{***}		

Table 4: Best window length of the simple method.

	S&P 500	Dow Jones	NASDAQ
Best window	39	39	35
Log likelihood	64332.12	84973.79	33611.08

We also train the simple statistical method on the three index return series. As Table 4 shows, the best window length is 39 trading days for S&P 500, 39 trading days for Dow Jones, and 35 trading days for NASDAQ. Under these settings of window length, we can gain maximum log likelihood of the train samples.

After we train all the models, we use them to forecast volatility day by day on the test set. We calculate the log likelihood of the test sample for each model and compare the values of the log-likelihood function, which are shown in Table 5. We can conclude from Table 5 that, for all the three indices, LSTM (likelihood-based loss) gains the largest log likelihood and can be the best model to forecast volatility. The value of log likelihood when using DNN (likelihood-based loss) is not as good as LSTM (likelihood-based loss), but is larger than the value when using ARMA-GARCH. LSTM performs better with likelihood-based loss function than with MSE loss function, and DNN also performs better with likelihood-based loss function than with MSE loss function. LSTM (MSE loss) gains larger log likelihood than ARMA-GARCH in two cases of index, while DNN (MSE loss) gains smaller log likelihood than ARMA-GARCH for all the three indices. Simple historical statistics is the method that gains smallest log likelihood for all the indices.

To intuitively exhibit the forecasting ability of the deep learning models and econometric model, we use the log-likelihood value that simple method gains as the benchmark and calculate the improvements which the five models can produce upon it. Figure 6 shows the percentage of improvements that LSTM (likelihood-based loss), DNN

Table 5: Log likelihoods of the test sample.

	S&P 500	Dow Jones	NASDAQ
LSTM (likelihood-based loss)	11406.25	15448.84	6074.32
DNN (likelihood-based loss)	11346.34	15392.89	6062.46
LSTM (MSE loss)	11348.36	15329.30	6036.95
DNN (MSE loss)	11275.18	15259.39	5999.94
ARMA-GARCH	11291.16	15383.54	6012.85
Simple	11061.59	15123.93	5887.78

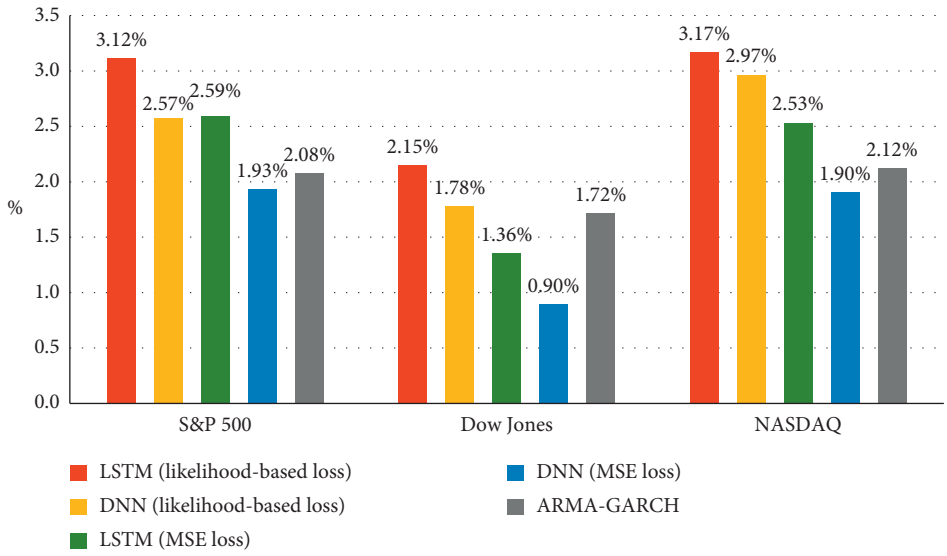


Figure 6: Log-likelihood improvements upon the simple method.

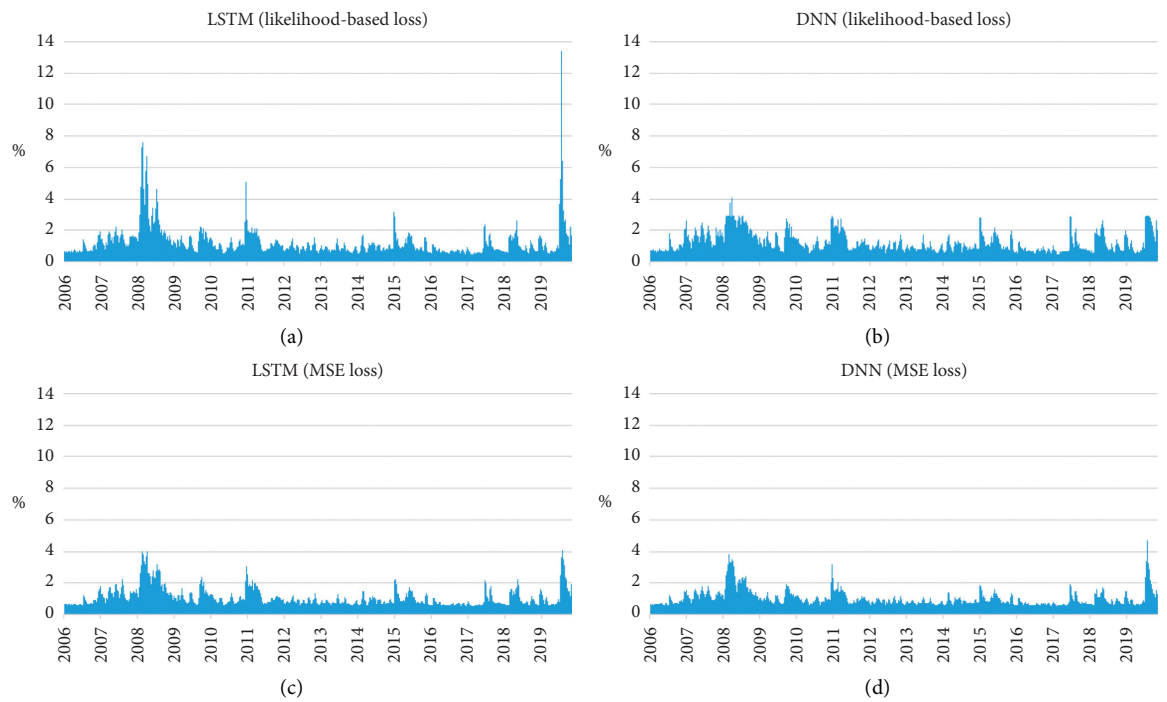


Figure 7: Continued.

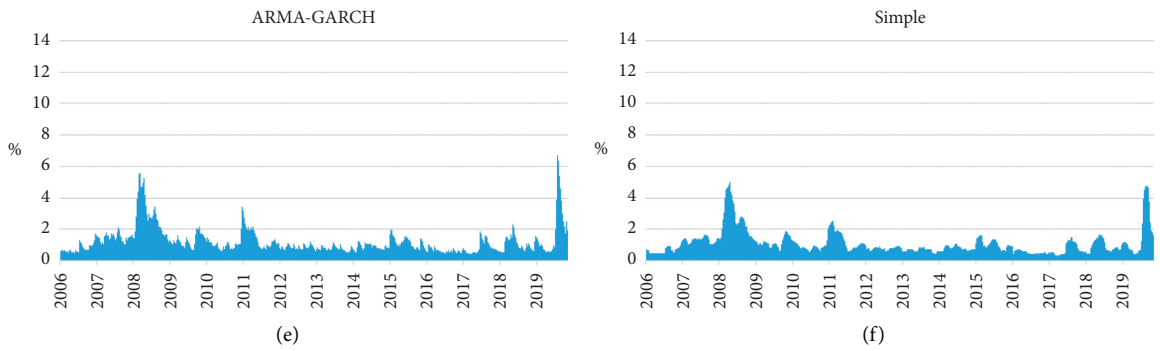


Figure 7: Volatilities of S&P 500 that the models forecast on the test period.

(likelihood-based loss), LSTM (MSE loss), DNN (MSE loss), and ARMA-GARCH make. For S&P 500, the improvements of the five models are 3.12%, 2.57%, 2.59%, 1.93%, and 2.08%. For Dow Jones, the improvements of the five models are 2.15%, 1.78%, 1.36%, 0.90%, and 1.72%. For NASDAQ, the improvements of the five models are 3.17%, 2.97%, 2.53%, 1.90%, and 2.12%.

Figure 7 shows the volatilities that the six models forecast day by day, when we use the trained models on the test sample of S&P 500. They show very similar trends in the long term. The volatilities which LSTM (likelihood-based loss) forecast have extreme peak values. The volatilities which DNN (likelihood-based loss) and LSTM (MSE loss) forecast have relatively low peak values. Although these three models are all deep learning methods, they show different characteristics in this aspect. The volatilities which the simple method forecast change more smoothly than the other five.

To see more details of the volatilities that the models forecast, we plot the six volatility series within the latest one year, which covers from July 1, 2019, to June 30, 2020, and show them together in Figure 8. It is clear in this figure that the forecast of the simple method is smoother than the other five. The forecasts of deep learning models and econometric models have similar values when volatility is relatively low. But when the market is in extreme condition around March 2020, they act very differently. Since the US stock market is under historic shock during this period of time, LSTM (likelihood-based loss) seems to capture the property better.

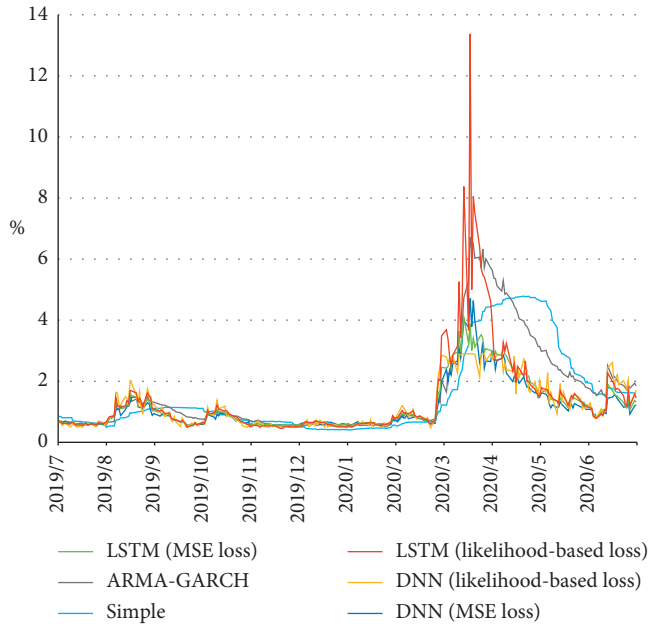


Figure 8: Volatilities of S&P 500 that the models forecast within the latest one year.

3.2. Sensitivity Analysis. In order to test the robustness of the forecasting result, we further use two different train-test sample splits and make the same kind of forecasting. The size of the test sample is set to be 10% and 20% separately, and we make the validation set of the deep learning models always have the same size as the test set. We can check whether the deep learning models with likelihood-based loss function still gain good performance among all the methods.

Table 6 shows the length of the train set and test set in trading days, under different train-test sample splits. When the test set is 10% of the full sample, we train the models in a relatively longer period and test them in a shorter range. When the test set is 20% of the full sample, we train the models in a relatively shorter period and test them in a longer range.

When the test sample size is set to be 10%, we train the ARMA(m, n)-GARCH(p, q) models and choose the best one with minimum BIC. As Table 7 shows, for S&P 500, it is ARMA(1, 1)-GARCH(1, 2), for Dow Jones, it is ARMA(1, 1)-GARCH(1, 2), and for NASDAQ, it is ARMA(1, 2)-GARCH(1, 1). The selections of m, n, p, and q are totally the same as in the previous section.

Then we also train the simple statistical method on the three index return series. As Table 8 shows, the best window length is 39 trading days for S&P 500, 39 trading days for Dow Jones, and 48 trading days for NASDAQ. The best window length only changes for NASDAQ.

After we train all the models, we can use them to forecast volatility day by day on the test set, which is set to be 10% of the

Table 6: Length of the train set and test set in trading days under different train-test sample splits.

	S&P 500		Dow Jones		NASDAQ	
	Train	Test	Train	Test	Train	Test
Test 10%	20917	2322	27987	3108	11212	1244
Test 15%	19755	3484	26433	4662	10590	1866
Test 20%	18594	4645	24878	6217	9967	2489

Table 7: ARMA-GARCH model with minimum BIC when the test set is 10%.

	S&P 500	Dow Jones	NASDAQ
Best model	ARMA(1, 1)-GARCH(1, 2)	ARMA(1, 1)-GARCH(1, 2)	ARMA(1, 2)-GARCH(1, 1)

Table 8: Best window length to maximize the log likelihood when the test set is 10%.

	S&P 500	Dow Jones	NASDAQ
Best window	39	39	48

Table 9: Log likelihoods of the test sample when the test set is 10%.

	S&P 500	Dow Jones	NASDAQ
LSTM (likelihood-based loss)	7901.28	10255.62	3973.68
DNN (likelihood-based loss)	7888.22	10198.73	3963.67
LSTM (MSE loss)	7873.73	10156.86	3942.13
DNN (MSE loss)	7828.06	10089.74	3914.47
ARMA-GARCH	7812.65	10187.24	3923.81
Simple	7631.45	9958.84	3802.13

full sample. We calculate the log likelihood of the test sample for each model and compare the values of the log-likelihood function, which are shown in Table 9. For all indices, LSTM (likelihood-based loss) still gains the best performance among all the methods. DNN (likelihood-based loss) is always ranked second. LSTM performs better with likelihood-based loss function than with MSE loss function, and DNN also performs better with likelihood-based loss function than with MSE loss function. LSTM (MSE loss) gains larger log likelihood than ARMA-GARCH in two cases of index, while DNN (MSE loss) gains larger log likelihood than ARMA-GARCH just for S&P 500. The simple method always gains smallest log likelihood for all the three indices.

To intuitively exhibit the forecasting ability of the deep learning models and econometric model, we also calculate the percentage of improvements that LSTM (likelihood-based loss), DNN (likelihood-based loss), LSTM (MSE loss), DNN (MSE loss), and ARMA-GARCH make upon the simple method and show the result in Figure 9. For S&P 500, the improvements of the five models are 3.54%, 3.36%, 3.17%, 2.58%, and 2.37%. For Dow Jones, the improvements of the five models are 2.98%, 2.41%, 1.99%, 1.31%, and 2.29%. For NASDAQ, the improvements of the five models are 4.51%, 4.25%, 3.68%, 2.95%, and 3.20%.

When the test sample size is set to be 20%, we train the ARMA(m, n)-GARCH(p, q) models and choose the best one with minimum BIC. As Table 10 shows, for S&P 500, it is ARMA(1, 1)-GARCH(1, 2), for Dow Jones, it is ARMA(1, 1)-GARCH(1, 2), and for NASDAQ, it is ARMA(1, 2)-GARCH(1, 2). The unique change also comes from NASDAQ. It may be because the sample size of NASDAQ is the smallest among the three indices.

We also train the simple statistical method and show the best window length for each index in Table 11. The best length is 39 trading days for S&P 500, 39 trading days for Dow Jones, and 35 trading days for NASDAQ. The best window lengths are the same as in the previous section.

After we train all the models, we can use them to forecast volatility day by day on the test set, which is set to be 20% of the full sample. We calculate the log likelihood of the test sample for each model and compare the values of the log-likelihood function, which are shown in Table 12. For all the three indices, LSTM (likelihood-based loss) performs better than DNN (likelihood-based loss), DNN (likelihood-based loss) performs better than LSTM (MSE loss), and LSTM (MSE loss) performs better than DNN (MSE loss). Deep learning models with MSE loss function gain relatively weak forecasting ability, since they cannot beat ARMA-GARCH in most cases. The simple method still makes smallest log likelihood for all the three indices.

We also calculate the percentage of improvements that LSTM (likelihood-based loss), DNN (likelihood-based loss), LSTM (MSE loss), DNN (MSE loss), and ARMA-GARCH make upon the simple method and show the result in

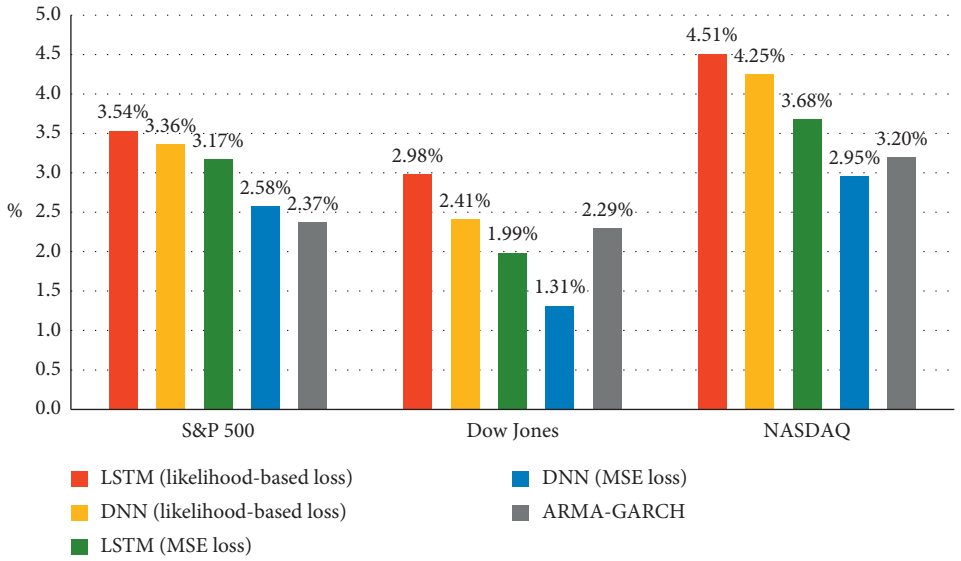


Figure 9: Log-likelihood improvements upon the simple method when the test set is 10%.

Table 10: ARMA-GARCH model with minimum BIC when the test set is 20%.

	S&P 500	Dow Jones	NASDAQ
Best model	ARMA(1, 1)-GARCH(1, 2)	ARMA(1, 1)-GARCH (1, 2)	ARMA(1, 2)-GARCH(1, 2)

Table 11: Best window length to maximize the log likelihood when the test set is 20%.

	S&P 500	Dow Jones	NASDAQ
Best window	39	39	35

Table 12: Log likelihoods of the test sample when the test set is 20%.

	S&P 500	Dow Jones	NASDAQ
LSTM (likelihood-based loss)	15197.71	20290.54	8001.75
DNN (likelihood-based loss)	15193.05	20236.08	7969.07
LSTM (MSE loss)	15140.80	20174.46	7913.75
DNN (MSE loss)	15087.88	20114.91	7856.39
ARMA-GARCH	15124.33	20203.73	7929.03
Simple	14883.33	19902.60	7787.29

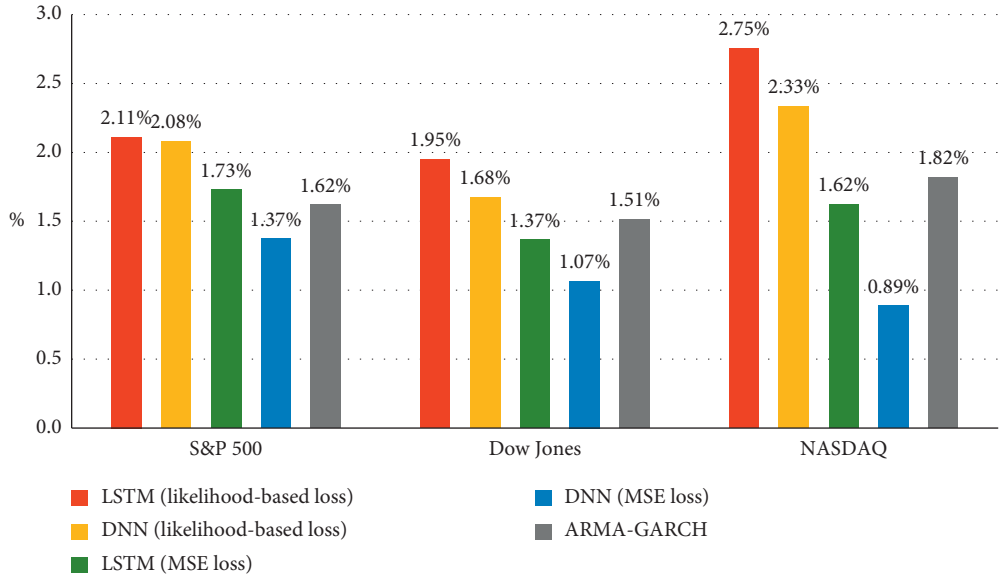


Figure 10: Log-likelihood improvements upon the simple method when the test set is 20%.

Figure 10. For S&P 500, the improvements of the five models are 2.11%, 2.08%, 1.73%, 1.37%, and 1.62%. For Dow Jones, the improvements of the five models are 1.95%, 1.68%, 1.37%, 1.07%, and 1.51%. For NASDAQ, the improvements of the five models are 2.75%, 2.33%, 1.62%, 0.89%, and 1.82%.

The results in this section show that the forecasting abilities of our deep learning models with likelihood-based loss function are robust when we use different sample split settings. Considering the samples of three indices cover different periods, we can further state that their forecasting abilities are not sensitive to different choices of sample periods. The deep learning models with likelihood-based loss function are robustly better in forecasting volatility than the other models, and LSTM (likelihood-based loss) is always the best one among all the methods.

Considering the results when the test set is 10%, 15%, and 20% of the full sample, it is clear that the deep learning models with MSE loss function are not as good as the deep learning models with likelihood-based loss function. The main reason is the inconsistency that we have discussed earlier. But LSTM (MSE loss) still can beat ARMA-GARCH in more than half of the cases, even though it suffers from this disadvantage. DNN (MSE loss) is the weaker one, and it cannot beat ARMA-GARCH in most cases.

4. Conclusions

In this paper, we use deep learning models, an econometric model, and a simple statistical method to forecast the volatility of three US stock indices. Different from related research studies, we further introduce a likelihood-based loss function to train the deep learning models and test all the methods by the likelihood of the test sample. By doing so, we can incorporate fewer errors into the process of volatility forecasting when using deep learning models. At the same time, we can make a fairer comparison among the models we study.

The results of the empirical study show that our deep learning models with likelihood-based loss function forecast volatility more precisely than the econometric model, and LSTM (likelihood-based loss) is the better one in these two deep learning models. The volatility series forecasted by the six models show very similar trends in the long term. LSTM (likelihood-based loss) seems to capture the property better under the extreme condition of the US stock market, around March 2020.

Then we change the setting of the train-test sample split and do the sensitivity analysis. We can conclude that the deep learning models with likelihood-based loss function are robustly better in forecasting volatility than the other models, and LSTM (likelihood-based loss) is always the best. LSTM (MSE loss) can beat ARMA-GARCH in more than half of the cases, even though it suffers from the disadvantage of inconsistency. DNN (MSE loss) is weaker and it cannot beat ARMA-GARCH in most cases. At the same time, all the deep learning models and the econometric model gain positive improvement upon the simple method.

Data Availability

The data used to support the findings of this study are available 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.

References

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[2] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.

[3] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[4] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[5] D. B. Nelson, "Conditional heteroskedasticity in asset returns: a new approach," *Econometrica*, vol. 59, no. 2, pp. 347–370, 1991.

[6] F. Corsi, "A simple approximate long-memory model of realized volatility," *Journal of Financial Econometrics*, vol. 7, no. 2, pp. 174–196, 2009.

[7] M. Bildirici and Ö. Ö. Ersin, "Improving forecasts of GARCH family models with the artificial neural networks: an application to the daily returns in Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7355–7362, 2009.

[8] E. Hajizadeh, A. Seifi, M. H. Fazel Zarandi, and I. B. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[9] W. Kristjanpoller, A. Fadic, and M. C. Minutolo, "Volatility forecast using hybrid Neural Network models," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2437–2442, 2014.

[10] W. Kristjanpoller and E. Hernndez, "Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors," *Expert Systems with Applications*, vol. 84, pp. 290–300, 2017.

[11] W. Kristjanpoller and M. C. Minutolo, "Gold price volatility: a forecasting approach using the Artificial Neural Network-GARCH model," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7245–7251, 2015.

[12] W. Kristjanpoller and M. C. Minutolo, "Forecasting volatility of oil price using an Artificial Neural Network-GARCH model," *Expert Systems with Applications*, vol. 65, pp. 233–241, 2016.

[13] W. Kristjanpoller and M. C. Minutolo, "A hybrid volatility forecasting framework integrating GARCH, Artificial Neural Network, technical analysis and principal components

analysis," *Expert Systems with Applications*, vol. 109, pp. 1–11, 2018.

[14] T. H. Roh, "Forecasting the volatility of stock price index," *Expert Systems with Applications*, vol. 33, no. 4, pp. 916–922, 2007.

[15] A. Baffour, J. Feng, and E. Taylor, "A hybrid artificial neural network-GJR modeling approach to forecasting currency exchange rate volatility," *Neurocomputing*, vol. 365, pp. 285–301, 2019.

[16] S. Lahmiri, "Modeling and predicting historical volatility in exchange rate markets," *Physica A: Statistical Mechanics and Its Applications*, vol. 471, pp. 387–395, 2017.

[17] Y. Peng, P. H. M. Albuquerque, J. M. Camboim de Sá, A. J. A. Padula, and M. R. Montenegro, "The best of two worlds: forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression," *Expert Systems with Applications*, vol. 97, pp. 177–192, 2018.

[18] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, "Forecasting volatility with a stacked model based on a hybridized Artificial Neural Network," *Expert Systems with Applications*, vol. 129, pp. 1–9, 2019.

[19] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[20] Y. Liu, "Novel volatility forecasting using deep learning-long short term memory recurrent neural networks," *Expert Systems with Applications*, vol. 132, pp. 99–109, 2019.

[21] Y. Hu, J. Ni, and L. Wen, "A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction," *Physica A: Statistical Mechanics and Its Applications*, vol. 557, pp. 1–14, 2020.

[22] F. Z. Xing, E. Cambria, and Y. Zhang, "Sentiment-aware volatility forecasting," *Knowledge-Based Systems*, vol. 176, pp. 68–76, 2019.

[23] A. Vidal and W. Kristjanpoller, "Gold volatility prediction using a CNN-LSTM approach," *Expert Systems with Applications*, vol. 157, pp. 1–9, 2020.

[24] S. Yu and Z. Li, "Forecasting stock price index volatility with LSTM deep neural network," in *Recent Developments in Data Science and Business Analytics*, M. Tavana et al., Ed., pp. 265–272, Springer, 2018.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Hindawi
Complexity
Volume 2021, Article ID 6616121, 17 pages
https://doi.org/10.1155/2021/6616121

Research Article

Claim Amount Forecasting and Pricing of Automobile Insurance Based on the BP Neural Network

Wenguang Yu,[1] **Guofeng Guan**,[2] **Jingchao Li**,[3] **Qi Wang**,[2] **Xiaohan Xie**,[1] **Yu Zhang**,[1] **Yujuan Huang**,[4] **Xinliang Yu**,[1] **and Chaoran Cui**[5]

[1]*School of Insurance, Shandong University of Finance and Economics, Jinan 250014, China*
[2]*School of Mathematic and Quantitative Economics, Shandong University of Finance and Economics, Jinan 250014, China*
[3]*College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China*
[4]*Office of Academic Research, Shandong Jiaotong University, Jinan 250357, China*
[5]*School of Computer Science & Technology, Shandong University of Finance and Economics, Jinan 250014, China*

Correspondence should be addressed to Wenguang Yu; yuwg@sdufe.edu.cn

Received 31 October 2020; Revised 23 December 2020; Accepted 7 January 2021; Published 20 January 2021

Academic Editor: Benjamin Miranda Tabak

The BP neural network model is a hot issue in recent academic research, and it has been successfully applied to many other fields, but few researchers apply the BP neural network model to the field of automobile insurance. The main method that has been used in the prediction of the total claim amount in automobile insurance is the generalized linear model, where the BP neural network model could provide a different approach to estimate the total claim loss. This paper uses a genetic algorithm to optimize the structure of the BP neural network at first, and the calculation speed is significantly improved. At the same time, by considering the overfitting problem, an early stop method is introduced to avoid the overfitting problem. In the model, a three-layer BP neural network model, which includes the input layer, hidden layer, and output layer, is trained. With consideration of various factors, a total claim amount prediction model is established, and the trained BP neural network model is used to predict the total claim amount of automobile insurance based on the data of the training set. The results show that the accuracy of the prediction by using the BP neural network model to both the data of Shandong Province and to the data of six cities is over 95%. Then, the predicted total claim amount is used to calculate premiums for five cities in Shandong Province according to credibility theory. The results show that the average premium of the five cities is slightly higher than the actual claim amount of the city. The combination of BP neural network and credibility theory can perform accurate claim amount estimation and pricing for automobile insurance, which can effectively improve the current situation of the automobile insurance business and promote the development of insurance industry.

1. Introduction

Insurance industry in China has made great progress along with the continuous development of Chinese economy, and insurance plays an increasingly important role in people's daily life. Therefore, a fair and comprehensive pricing system is essential to the development of insurance industry, which can effectively avoid the adverse selection problem, can maintain the insurance industry in a healthy competition, and can promote the development of insurance industry. According to the China Statistical Yearbook in 2015, the premium income of automobile insurance was 619.9 billion yuan, accounting for 73.59% of the premium income of property insurance; in 2016, the premium income of automobile insurance was 683.42 billion yuan, accounting for 73.76% of the premium income of property insurance; in 2017, the premium income of automobile insurance was 752.11 billion yuan, accounting for 71.35% of the premium income of property insurance. It can be seen that the premium income of automobile insurance is steadily increasing, and the proportion of the premium income to the property insurance premium income is maintained at more than 70%.

Hence, the profitability of automobile insurance plays a decisive role in the operating efficiency of property insurance companies. However, the operating condition of the automobile insurance business in China is generally poor. According to the data of China Insurance Regulatory Commission in 2018, only seven property and automobile insurance companies made profit, where 48 out of 55 unlisted property and casualty insurance companies engaged in the automobile insurance business suffered losses to varying degrees. The total loss of automobile insurance is about 8.68 billion yuan, and the loss ratio is too high in most years.

The main problem of excessive high claims in automobile insurance is inadequate premium ratemaking. Inadequate premium ratemaking does not simply mean that the premium rates are too high or too low, but means that the premiums among different risks are not differentiated or the distinctions are inappropriate, which leads to a large number of adverse selection and causes poor business quality so that the premium income does not match with the risk the company takes.

The study on premium ratemaking of automobile insurance has been attracting many scholars' attention. Bailey and Simon first proposed the idea of classification pricing, which classified insurance policies according to a certain characteristic of the risk and priced each type of insurance policies separately [1]. Denneberg first proposed the Poisson-gamma model to study the frequency of nonhomogeneous insurance policy claims and obtained good fitting results in empirical research on auto insurance [2].

The generalized linear model (GLM) is a widely accepted model for premium ratemaking of automobile insurance in recent decades. In the last century, Nelder and Wedderburn first proposed the GLM, which has been widely used as once proposed [3]. Samson and Thomas used the GLM to perform pricing for the premium rate based on the data of a third-liability insurance from an insurance company in UK, and they found that no claim discount, automobile type, region, and age class of the automobile owner have significant impact on both the claim amount and claim frequency [4]. Smyth introduced the maximum likelihood estimation of the DGLM, considered the situation when the population obeys the normal and inverse Gaussian distribution, and analyzed the selection of the initial value of the iteration [5]. Stroiński and Currie evaluated the risk through the GLM based on the data of a third-liability insurance from an insurance company in UK and proved that the GLM plays an important role in premium ratemaking of automobile insurance [6]. Meng briefly analyzed the shortcomings of traditional nonlife insurance product rate determination methods, such as the single analysis method and minimum bias procedure. It also proved that the GLM can be applied to determine premiums for automobile products by using a group of automobile insurance data [7]. Draper showed that the GLM fitted better than the traditional model based on automobile insurance data of an insurance company in France by using SAS software [8]. Zhao and Chen applied the dual-generalized linear model to price the automobile insurance premium rate. Based on an empirical study of a group of automobile insurance loss data, they found that the dual-generalized linear model is more reasonable to determine the premium rate compared with the GLM [9].

With the development of research on GLMs, scholars have found limitations of the GLM. Initially, the GLM only builds the regression relationship between the expected value of the response variable and explanatory variables and assumed that the dispersion parameter is constant. Although this assumption simplifies the model, it does not hold for some cases. Smyth and Jørgensen applied the DGLM to the premium ratemaking of auto insurance and directly predicted the premium rate of auto insurance. However, the regional factors were excluded in the empirical study, and the obtained rate structure did not reflect the regional differences [10]. Antonio and Beirlant combined the generalized linear-mixed model and Bayesian method to determine premium rates and found that this combination performs well [11]. According to the nonlife insurance loss data, Frees et al. firstly analyzed the claim frequency, claim type, and claim intensity under the framework of the hierarchical model, then applied the Bayesian method to determine the joint probability distribution among variables, and finally predicted the total claim loss in the future. Finally, they used the simulation method to predict the premium under the policy limit [12]. Wang et al. considered that the fat tail of automobile insurance loss data has significant impact on premium ratemaking; hence, they introduced the density function to describe the fat tail distribution, and based on it, they constructed the GAMLESS model under the type-two generalized beta distribution, which improved the limitations of assumptions and parameter modeling in the traditional GLM and increased the accuracy of the prediction for automobile insurance loss [13]. In the past, it was assumed that the total claim distribution was compound Poisson-gamma distributed. The GLM was used for the claim frequency and claim intensity, respectively, and then, the expected total claim was set to be equal to the expected value of claim frequency times the expected value of claim intensity. Zhang and Xie assumed that the total claim amount followed the Tweetie distribution, then directly established a GLM for the total claim amount, and obtained the average value of the total claim amount for each risk. Through the empirical analysis of the data, the above two methods are compared, and results showed that the data fitting degree based on Tweetie distribution was better [14].

With the development of science and technology, more and more attention has been paid to the driving behavior in premium ratemaking of automobile insurance. Ayuso et al. demonstrated how automobile insurance can be improved by incorporating mileage and driver behavior data. The key idea is that telemetry should facilitate the inclusion within insurance pricing of those factors that traffic authorities identify as being associated with risky drivers [15]. Huang and Meng studied the use of a wide range of driving behavior variables to predict the risk probability and claim frequency of an insured vehicle. The advantage of the model is that it can improve the interpretability and predictive accuracy of the model at the same time, thus providing a new solution for the classification pricing of UBI products [16].

Artificial neural network is a new model. It is a kind of computational model which imitates the structure of biological network after human beings have fully studied the structure of animals.

1. BP neural network model is applied to automobile insurance.
2. Genetic algorithm is used to optimize the structure of BP neural network.
3. An early stop method is introduced to avoid the overfitting problem.

The neural network is composed of multiple neurons, and each neuron is connected to each other to form a network. The network transmits and processes information and imitates the human brain structure. It has strong adaptability and can process linear and nonlinear data. The BP neural network algorithm based on backpropagation is one of the most mature and widely used neural network algorithms.

Many scholars have introduced neural network algorithms into the insurance industry. Brockett et al. applied Kohonen's self-organizing competitive networks to identify fraud problems in personal injury insurance [17]. Liu et al. compared the multiclass AdaBoost tree with the generalized linear model, two-layer BP (backpropagation) neural network, and support vector machine (SVM) to predict the effect of claim intensity, and they found that the AdaBoost method has the best prediction accuracy and relatively small variance [18]. Mzhavia applied neural networks to the risk classification of car drivers and found a set of neural networks with the best classification effect. The number of neurons in the input layer, hidden layer, and output layer of the network was 11, 12, and 2. The inspirit function was a hyperbolic tangent function [19]. Under the assumption of Poisson distribution, Wüthrich used the speed-acceleration data recorded by the Internet of vehicles to extract the driving behavior factor through the Bottleneck neural network learning algorithm and established a generalized additive model to predict the frequency of claims [20]. Zhang and Wang applied SOM to the claim prediction of automobile insurance, which provided a new way of premium ratemaking of automobile insurance [21]. In addition, the application of the neural network in other fields can be seen in Lin et al's research [22–25].

In summary, it can be found that there are abundant research studies on automobile insurance pricing and premium ratemaking, and scholars have been seeking new methods to price for automobile insurance more accurately. At the early stage, scholars mainly studied generalized linear models and continuously improved the generalized linear models so that the generalized linear models could be better applied to the premium ratemaking. However, the generalized linear models still have some shortcomings. The BP neural network has strong fault tolerance and high accuracy when fitting data. Aiming at the characteristics of the BP neural network, this paper tries to apply the BP neural network to price for automobile insurance rates and verifies the model with real data from insurance companies. The accuracy of the model is expected to provide new ideas for the premium ratemaking in the insurance industry.

The outline of the paper is organized as follows. In Section 2, we construct the BP neural network model. In Section 3, empirical analysis is carried out on the model. In Section 4, according to the characteristics of the data, we extend the model and verify that the model is also applicable to all regions of the country. In Section 5, we study the application of the model in automobile insurance ratemaking. Finally, conclusions and policy recommendations are given in Section 6.

2. The BP Neural Network Model and Optimization

2.1. The BP Neural Network Model. The BP neural network is currently the most widely used neural network. The learning rule is to adopt the algorithm of backpropagation, using the steepest descent method to adjust the coefficient in reverse according to the error between the actual output value and expected output value, until the coefficient is optimized to make the error within the acceptable range. The BP neural network can learn fixed patterns, use some data to determine the corresponding parameters, and then make predictions based on these parameters.

A three-layer BP neural network model is used in this paper, which is composed of an input layer, an output layer, and a hidden layer. Its structure is shown in Figure 1.

We assume that there are n neurons in the input layer, five neurons in the hidden layer, and two neurons in the output layer. $X_k = (x_{1k}, x_{2k}, \ldots, x_{nk})$ which are input values, $k = 1, 2, \ldots, m$. α_{ij} are the weights connecting the input layer and the hidden layer and β_{jl} are the weights connecting the hidden layer and the output layer value, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, 5$, and $l = 1, 2$. Neurons in the same layer are not connected to each other, and each neuron between the input layer and hidden layer and hidden layer and output layer are connected.

The specific algorithm of the BP neural network model based on Figure 1 is as follows: assuming that the stimulus function uses the sigmoid function, sequentially input m sample data $X_1, X_2, \ldots, X_m$ and then randomly select the kth input sample $X_k = (x_{1k}, x_{2k}, \ldots, x_{nk})$; the hidden layer input vector is $Y_k = (y_{1k}, y_{2k}, \ldots, y_{5k})$, the hidden layer output vector is $Z_k = (z_{1k}, z_{2k}, \ldots, z_{5k})$, the input vector of the output layer is $\widetilde{Y}_k = (\widetilde{y}_{1k}, \widetilde{y}_{2k})$, $\widetilde{Z}_k = (\widetilde{z}_{1k}, \widetilde{z}_{2k})$ is the output vector of the output layer, the expected output vector is $R_k = (r_{1k}, r_{2k})$, the threshold of each neuron in the hidden layer is denoted as a_j, the threshold of each neuron in the output layer is denoted as b_l, the inspirit function is $f(\cdot)$, the learning parameter is μ, and $E = (1/2)\sum_{l=1}^{2}(r_{lk} - \widetilde{z}_{lk})^2$ is the error function.

The input and output of each neuron in the hidden layer and output layer can be calculated as follows:

$$y_{jk} = \sum_{i=1}^{n} \alpha_{ij} x_{ik} - a_j,$$
$$z_{jk} = f\left(y_{jk}\right),$$

$$\widetilde{y}_{lk} = \sum_{j=1}^{5} \beta_{jl} z_{jk} - b_l,$$

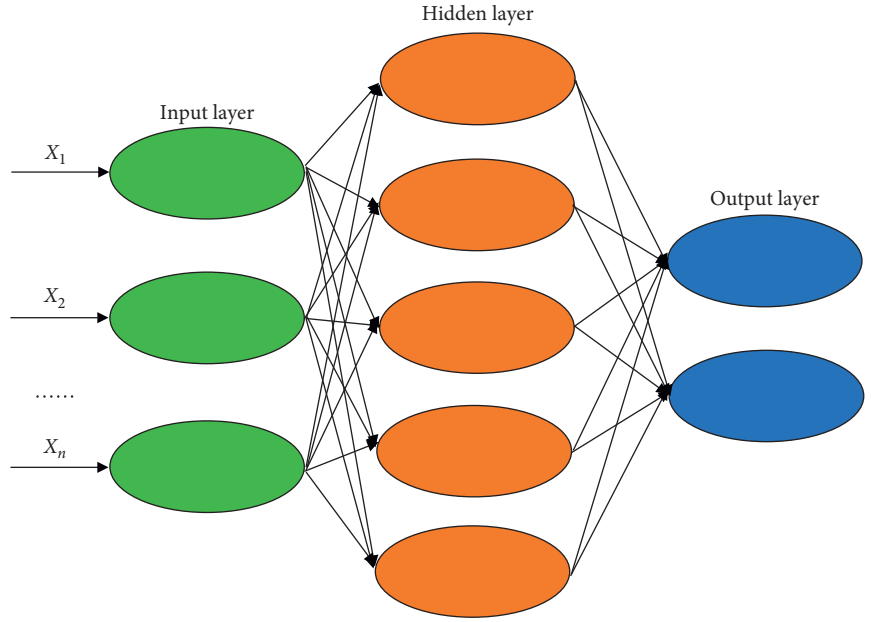


Figure 1: The three-layer BP neural network model.

$$\tilde{z}_{lk} = f(\tilde{y}_{lk}). \tag{1}$$

By using the expected output and actual output of the network, the partial derivative of the error function for each neuron in the output layer is as follows:

$$\frac{\partial E}{\partial \tilde{y}_{lk}} = \frac{\partial\left[(1/2)\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})^2\right]}{\partial \tilde{y}_{lk}} = \frac{\partial\left[(1/2)\sum_{l=1}^{2}(r_{lk}-f(\tilde{y}_{lk}))^2\right]}{\partial \tilde{y}_{lk}} = -\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})f'(\tilde{y}_{lk}) \triangleq -\delta_{lk}. \tag{2}$$

By using the connection weight from the hidden layer to the output layer, the output layer and output of the hidden layer to calculate the partial derivative of the error function for each neuron of the hidden layer is given as follows:

$$\begin{aligned}
\frac{\partial E}{\partial y_{jk}} &= \frac{\partial\left[(1/2)\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})^2\right]}{\partial y_{jk}}, \\
&= \frac{\partial\left[(1/2)\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})^2\right]}{\partial z_{jk}} \cdot \frac{\partial z_{jk}}{\partial y_{jk}}, \\
&= \frac{\partial\left[(1/2)\sum_{l=1}^{2}\left(r_{lk}-f\left(\sum_{j=1}^{5}\beta_{jl}z_{jk}-b_l\right)\right)^2\right]}{\partial z_{jk}} \cdot f'\left(y_{jk}\right), \\
&= -\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})f'(\tilde{y}_{lk})\beta_{jl} \cdot f'\left(y_{jk}\right), \\
&= -\left(\sum_{l=1}^{2}\delta_{lk}\beta_{jl}\right)f'\left(y_{jk}\right), \qquad \triangleq -\rho_{jk}.
\end{aligned} \tag{3}$$

Through the above two formulas, we can get the change of weight value β_{jl} in each adjustment as follows:

$$\Delta\beta_{jl} = -\mu\frac{\partial E}{\partial \beta_{jl}} = -\mu\frac{\partial E}{\partial \tilde{y}_{lk}} \cdot \frac{\partial \tilde{y}_{lk}}{\partial \beta_{jl}} = \mu\delta_{lk}z_{jk}. \tag{4}$$

After N adjustments, the $(N+1)$th value is as follows:

$$\beta_{jl}^{N+1} = \beta_{jl}^{N} + \Delta\beta_{jl}. \tag{5}$$

Similarly, we can get the change of weight value α_{ij} in each adjustment and the $(N+1)$th value after N adjustments as follows:

$$\begin{aligned}
\Delta\alpha_{ij} &= -\mu\frac{\partial E}{\partial \alpha_{ij}} = \mu x_{ik}\rho_{jk}, \\
\alpha_{ij}^{N+1} &= \alpha_{ij}^{N} + \Delta\alpha_{ij},
\end{aligned} \tag{6}$$

and the global error can be calculated as follows:

$$E = \frac{1}{2m}\sum_{k=1}^{m}\sum_{l=1}^{2}(r_{lk}-\tilde{z}_{lk})^2. \tag{7}$$

Finally, compare the size of the global error with the setting error. If the global error exceeds the setting error, keep adjusting the weights until the setting error is met.

The calculation process of the BP neural network model can be presented graphically, as shown in Figure 2.

2.2. Optimization and Control Problem of Overfitting Issue of the BP Neural Network. The BP neural network performs local search according to the gradient descent method, and the weight adjustment in the network is realized by the local adjustment. However, the BP neural network has the following problems. Firstly, adjusting the weight locally makes

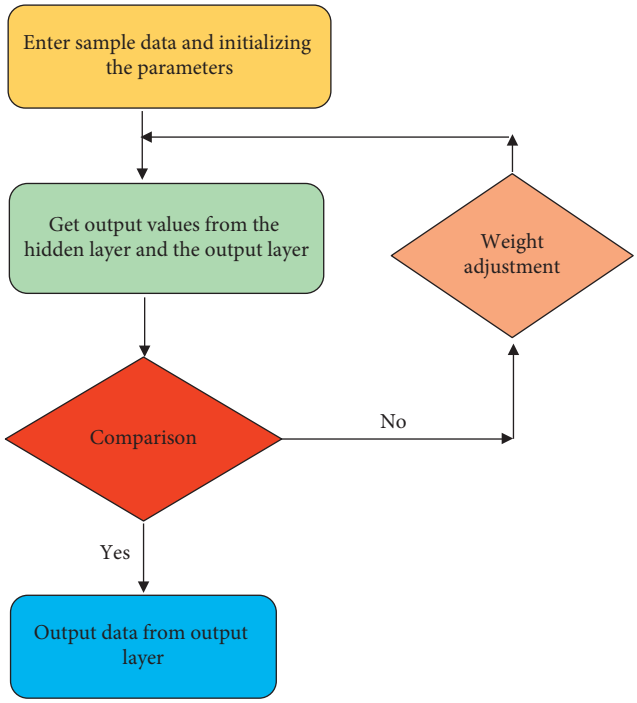


Figure 2: Back propagation flow chart of the BP neural network.

the weights fall into the local optimum rather than the global optimum. Secondly, when using the gradient descent method for optimization, there will be a flat area when the output neuron result is close to 0 or 1, where the error changes slightly with the weights in the flat zone, which causes the training process to be extremely slow and even be judged by the network that the global optimal solution has been found, and the network training will stop.

Considering the above shortcomings of the BP neural network, this paper chooses a genetic algorithm to improve the structure of the BP neural network. The genetic algorithm has the following characteristics:

(1) The genetic algorithm starts searching from multiple starting points instead of starting from one single point so that the search range is larger when searching for the best, which is more conducive to search for the global optimal solution.

(2) The genetic algorithm searches for the target solution in an adaptive manner, in which a series of operations such as selection, crossover, and mutation is operated based on a certain probability; hence, there are great flexibility and nondirectionality in the search process.

(3) The genetic algorithm does not rely on search space knowledge and other auxiliary information. It only calculates the degree of individual pros and cons based on the fitness function and performs subsequent operations on this basis.

(4) The genetic algorithm takes the coding of the required target variable as the operation target.

Based on the above characteristics, this method can use the parallel property of the global search to find the optimal connection weight set of the neural network, and the use of gradientless optimization and random operators helps to evolve the initial weights of the artificial neural network so that the probability of the BP algorithm falling into a local minimum is minimized.

Therefore, this paper is based on the genetic algorithm to search the optimal value for the initial weight and threshold in the neural network to optimize the BP neural network. The process is set as follows. At the first stage, the genetic algorithm is used to train the neural network. The genetic algorithm is used to evolve the optimal initial weight and threshold set of neural network training. This is achieved by the genetic algorithm simultaneously searching in all possible directions in the search space and narrowing it down to the area where the best possible weight and deviation can be found. In the second stage, the neural network is trained using the BP algorithm. The training starts by initializing the BP algorithm and using the genetic algorithm to assist in training the initial weights and thresholds of the evolution. The neural network with optimal weights and thresholds is initialized by using the BP algorithm, and the global optimal solution started by the genetic algorithm is searched continuously by adjusting the weights and thresholds of the neural network.

Although the genetic algorithm is used to optimize the structure of the BP neural network, the network may still have an overfitting problem. The overfitting problem means that the accuracy of the network on the training set is very high, but the accuracy on the test set is relatively low, which affects the generalization of the model.

In order to avoid overfitting, the early stop method is applied in this paper. This method divides the sample into a confirmed subset and uses this subset to test the network error during the training process. In the initial stage of training, the network error will be reduced; but, when the network begins to overfit, the network error will rise; when the network error rises in a certain number of iterations, the network stops training. At this time, the weight and bias value of the network can be obtained when the network error is the smallest.

In addition, this paper also uses 26,100 pieces of data to train each parameter by increasing the amount of data as much as possible. The feature diversity of the data helps to make full use of all training parameters to result estimation and simulation more accurate.

Finally, this paper also reduces the number of BP neural network layers and the input layer and hidden layer neurons to prevent overfitting when building the model.

Figure 3 shows the error reduction graph in network training. The errors of the training set, prediction set, and confirmation subset have been decreasing until they become stable, indicating that the network has no overfitting problems. At the same time, the prediction accuracy of each prediction set below is also very high which verifies the conclusion.

3. Empirical Analysis of the BP Neural Network Model

3.1. Data Source Analysis. The data in this paper comes from real claim data of an insurance company in Shandong Province. The data contains eight columns, which are owner's age, owner's gender, number of seats, vehicle age,

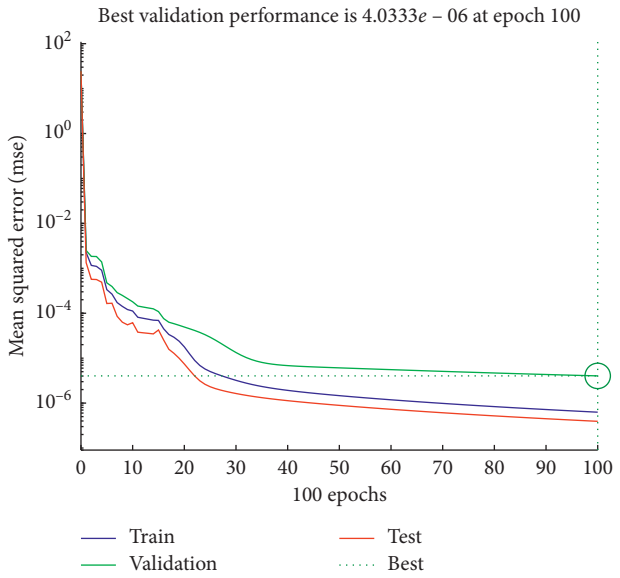


Figure 3: The error reduction graph under the early stop method.

purchase price, vehicle weight, NCD coefficient, and aggregate claim amounts, totaling 30,000 data. After removing outliers, there are 29,000 valid data remaining. The data covers 17 prefecture-level cities in Shandong Province, but the amount of data is not uniformly distributed in each prefecture-level city. Among them, the data of five prefecture-level cities such as Weifang, Binzhou, Jining, Yantai, and Weihai exceeds two thousand data, of which there are 6500 data in Yantai city, and the data of the remaining prefecture-level cities are less than one thousand. Some data are shown in Table 1.

3.2. Overall Modeling and Result Analysis. This paper takes the owner's age, owner's gender, number of seats, vehicle age, purchase price, vehicle weight, and NCD coefficient as input variables and the aggregate claim amount as the output variable. In addition, the data required by the BP neural network model is divided into a training set and a test set. The training set is used to train the network, adjust the parameters, and control the error within our goals, where the test set uses the trained network to predict the aggregate claim amount. In order to make the network training more accurate and to take the recognition degree of the output result graph into account, we select 90% of the data as a training set of the neural network, and the remaining 2,900 data are used as a test set to test prediction performance of the neural network.

Since there is a big difference in the magnitude of the data used in this paper, it will have a negative impact on the training of the network, reducing the learning ability of the network, and may even fail to reach the training goal. Therefore, all the data will be normalized at first, where all the data are mapped in the range of 0-1 to facilitate network training.

In this paper, the number of nodes in the input layer is set to 7, corresponding to 7 input variables. The number of nodes in the output layer is 1, which corresponds to the aggregate claim amount. The number of nodes in the hidden layer has a significant impact on the performance of the network, and this paper adopts an empirical algorithm to determine the number of nodes in the hidden layer, that is, $\sqrt{m+n}+a$, where m and n are the number of nodes in the input layer and output layer, respectively, and a is a random number, and after many tests and adjustments, it is taken 1 in this paper. The inspirit function between the input layer and hidden layer adopts a tan sig function, which is a hyperbolic tangent inspirit function. The activation function between the hidden layer and output layer adopts a purelin inspirit function, which is a linear inspirit function. The training function uses a trainlm function. The network parameters in this paper are set as follows: the learning rate is 0.1, and the target accuracy is set to 0.00001. We use prediction accuracy to measure the results of network prediction, which is defined as follows:

$$\text{prediction accuracy} = 1 - \frac{\text{prediction value} - \text{real value}}{\text{real value}}. \tag{8}$$

We first let the network randomly select the training set and use the selected training set to train the network. The analysis of the training network result is shown in Figure 3.

Figure 4 is a state diagram of using the training set to complete the training for the network, and it indicates that the error between the output value of the training set and the actual output value is 0.000000628 after the iteration; hence, the network can accurately fit this type of data. This also shows that the BP neural network model can be used to fit and predict the total claim amount of automobile insurance, and the trained network is a network with superior performance.

Figure 5 shows the goodness of the fit, and it indicates that global goodness of fit R reaches 0.99952, and the coefficient is very close to 1, which strongly indicates that the input neuron variable has a strong explanatory effect on the output neuron variable. The BP neural network model is very suitable for the fitting of automobile insurance claim data, and it can be used to fit and predict the aggregate claim amount.

Figure 6 is an effect diagram of using the trained network to fit the training set data. Since the individuals with no claims in the data set account for the majority, the bottom of the graph is denser, and the actual value and fitted value are stacked together, indicating that the fitting effect for zero claim amount is very good. For some very small claims, although the plots are relatively dense, it can be seen that they can be fitted well too. For individuals with relatively large claims, the network can also be accurately fitted. Hence, the network fits the training set well, and it fits most individuals accurately.

Given that the BP neural network model can accurately fit the training set, the trained network model is used to predict the data of the test set. Figure 7 shows the comparison between predicted values and actual values of the test set. The test data set also accounts for the majority of individuals with no claim, so the phenomenon of

Table 1: Claim data.

Owner's age	Owner's gender	Number of seats	Vehicle age	Purchase price (yuan)	Vehicle weight (kg)	NCD coefficient	Aggregate claim amounts
48	1	5	4	62,900.0	1039	−0.4	0.0
31	1	5	3	61,900.0	1265	−0.3	0.0
50	1	8	5	30,000.0	975	−1.3	0.0
47	1	5	9	114,800.0	1255	−1.3	0.0
33	1	5	13	263,800.0	1580	−0.15	0.0
31	2	7	5	38,800.0	1205	−0.3	0.0
40	1	5	6	37,900.0	1210	−1.3	0.0
26	2	5	3	44,900.0	1020	−1.2	0.0
33	1	7	8	278,800.0	1855	−1	0.0
37	1	5	9	37,900.0	895	−1	0.0
49	1	5	3	119,800.0	1501	0.7	0.0
40	2	8	2	52,300.0	1305	0.15	0.0
49	1	5	17	55,500.0	1050	−0.4	1,200.0

Note. In owner's gender, 1 indicates male and 2 indicates female.

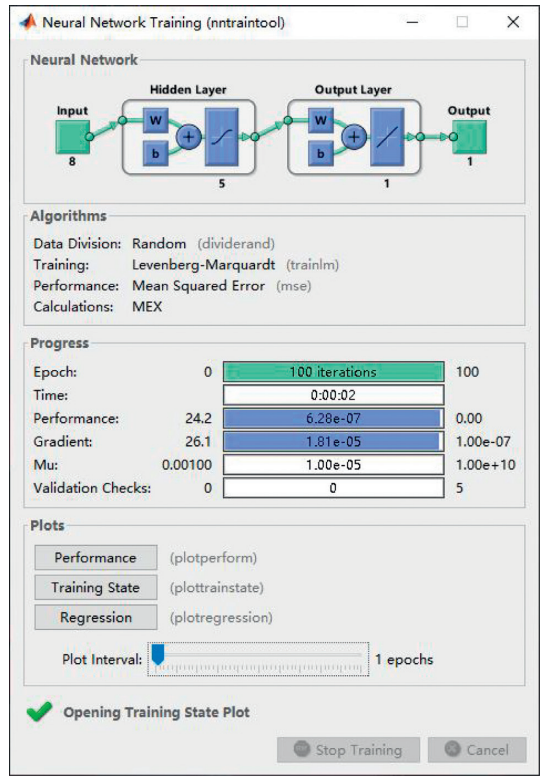


Figure 4: The overall modeling network state diagram.

accumulation appears at the bottom of the graph, which also shows that the network can accurately predict individuals with zero claims. In addition, the prediction of individuals whose total claim amount is not zero is also very accurate. The output shows that the prediction accuracy of the network on prediction data set is 99.68%, which proves the accuracy of the overall prediction results of the network, indicating that the BP neural network is very suitable for predicting the aggregate claim amount of automobile insurance.

4. Generalization of the Model Based on Data

As the automobile claim data is confidential for each insurance company, it is difficult to obtain comprehensive data. The data used in this paper only includes the claim data of various cities in Shandong Province, so the data has certain geographical limitations. Since each place has its own unique characteristics in terms of topography, weather, and economy, the data of each prefecture-level city has its own uniqueness. It can be considered that the data characteristics

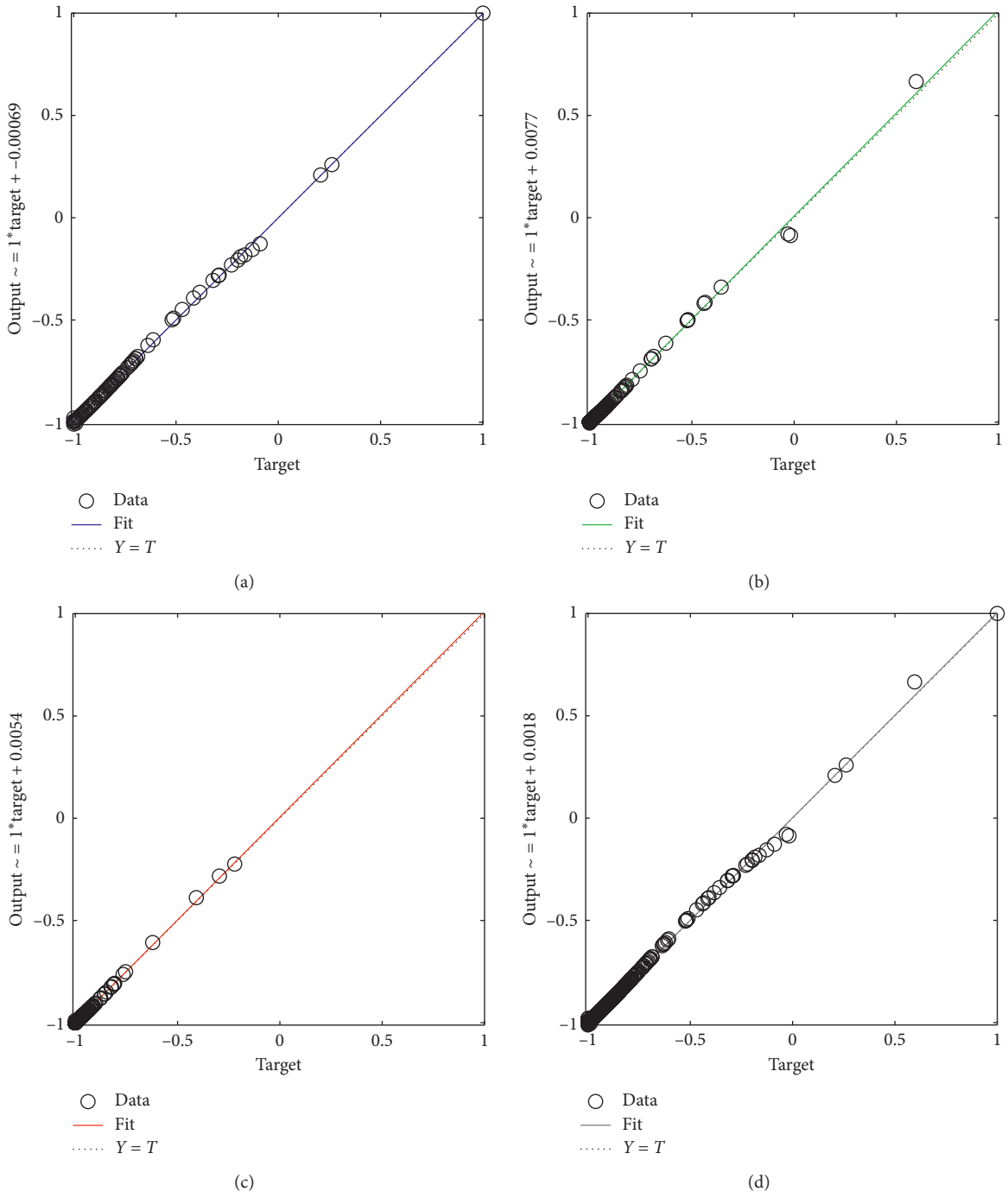


Figure 5: Overall goodness of the fit diagram: (a) training: $R = 0.99972$; (b) validation: $R = 0.99897$; (c) test: $R = 0.99967$; (d) all: $R = 0.99952$.

of each prefecture-level city are different; hence, the accuracy and adaptability of the model will be tested and verified based on the data at the prefecture-level city level. If it can be verified that the BP neural network model can accurately fit and predict the data of each prefecture-level city, then the model is applicable to data national wide. Because some prefecture-level cities contain relatively few data and are not representative, this paper only uses prefecture-level cities with more than two thousand data to fit and to predict, such as Binzhou City, Jining City, Weihai City, Weifang City, Jinan City, and Laiwu City.

When fitting and predicting the data of each prefecture-level city, this paper randomly divides the data into a training set and a prediction set by considering that a certain amount of data is required to train the network. The training set accounts for 90%, and the test set accounts for 10%. The target accuracy is set to 0.00001. The following describes the fitting and forecasting for the six prefecture-level cities in detail.

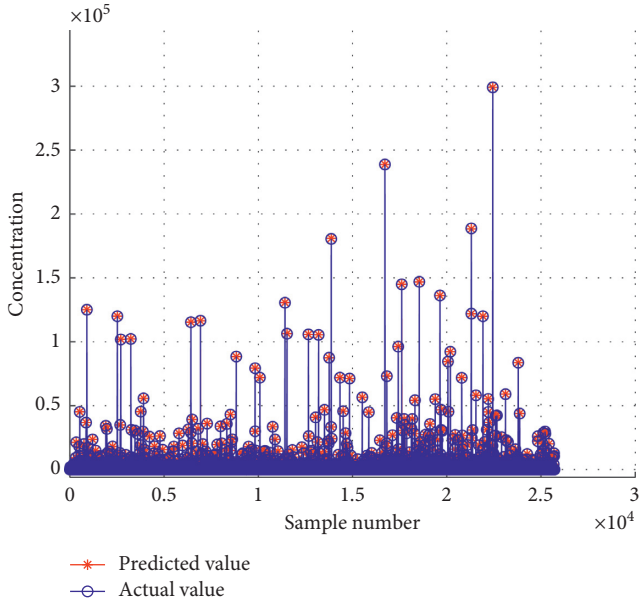


Figure 6: The overall modeling training set data fitting effect diagram.

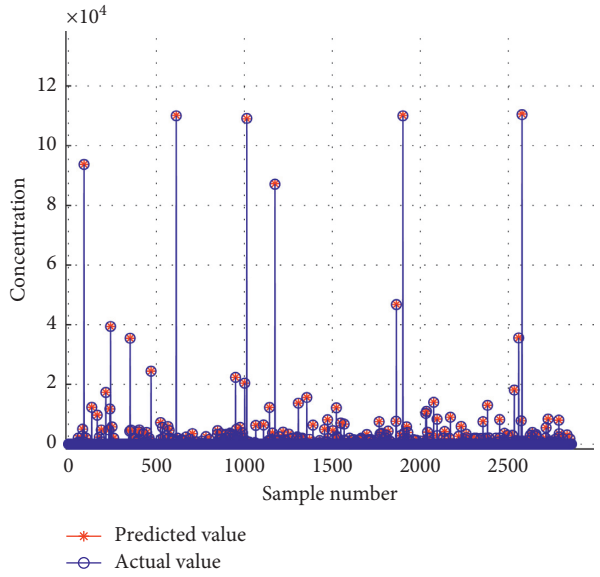


Figure 7: The prediction effect diagram of overall modeling test set data.

4.1. Binzhou City. Binzhou City has only 2300 pieces of data. By fitting and forecasting the data of Binzhou City, the following training results and prediction results are obtained, as shown in Figures 8 and 9.

Figure 8 shows the effect of fitting the data of the training set from Jinzhou City. Individuals with a claim amount of zero are accurately fitted and stacked at the bottom of the graph, and both small claims and large claims are also accurately fitted. From the above figure, it can be seen that the network accurately fits the Binzhou training set data as a whole.

Figure 9 gives the comparison between predicted values and actual values. It shows that the predicted values of the aggregate claim amount of zero is exactly the same as the true values, and small aggregate claim amounts can almost be predicted correctly. There is only a small error in the predicted value of the high claim data. From the output of the network, the prediction accuracy of the network on the prediction data set of Jinan City is 99.06%, which indicates that the network has made accurate predictions on the prediction data set of Binzhou City.

4.2. Jining City. There are only 2130 pieces of data in Jining City. By fitting and forecasting the data of Jining City, the following training result figure and prediction result figure are obtained, as shown in Figures 10 and 11.

Since this data set contains a large number of individuals with zero claims, there is a stacking phenomenon at the bottom of Figure 10, which also shows that the individuals with zero claim amount are accurately fitted. Figure 10 shows that the network accurately fits both small claims and large claims, meaning that the network fits well in general.

Figure 11 gives the comparison between the actual values and predicted values obtained by predicting the input data of the prediction set. It can be seen from Figure 11 that the predicted values of the small aggregate claim amounts are completely consistent with the actual values. The prediction accuracy of the network for the prediction set data is 99.49%, which shows that the network can also be applied to fit Jining automobile insurance claim data and accurately predict the accumulated claim amount.

4.3. Weihai City. There are 2020 pieces of data in Weihai City. By fitting and forecasting the data of Weihai City, the following training result figure and prediction result figure are obtained, as shown in Figures 12 and 13.

Figure 12 is a comparison diagram of the fitted values and actual values of the training set after the network is trained through the Weihai city training data set. It can be seen that there are many claims in the training set of this city, and the network accurately fits all the data with claims.

Figure 13 indicates that the network accurately predicts the data in the prediction data set with claims, and the predicted values are consistent with the actual values. The predicted and actual values of the data without claims are stacked at the bottom of the image, and the prediction accuracy of the prediction set of Weihai City by the network is 99.76%; hence, the prediction outcome of the network is very good.

4.4. Weifang City. There are 3000 pieces of data in Weifang City. By fitting and forecasting the data of Weifang City, the following training result figure and prediction result figure are obtained, as shown in Figures 14 and 15.

The accumulation phenomenon appears at the bottom of Figure 14, and the trained network accurately fits individuals

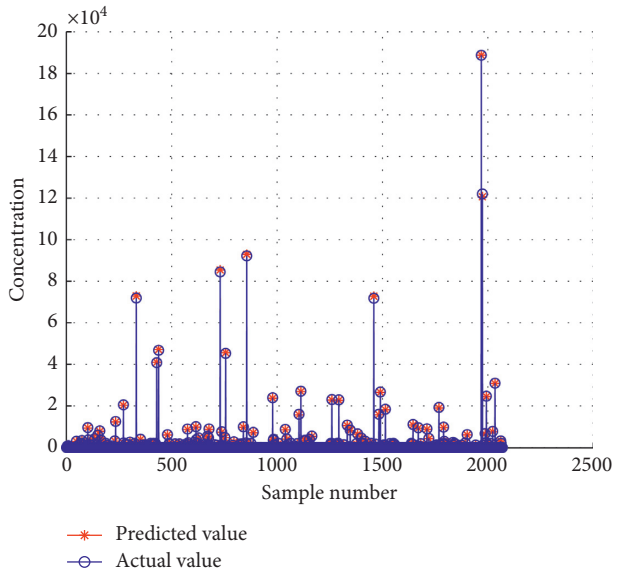


Figure 8: The fitting effect diagram of Binzhou training set data.

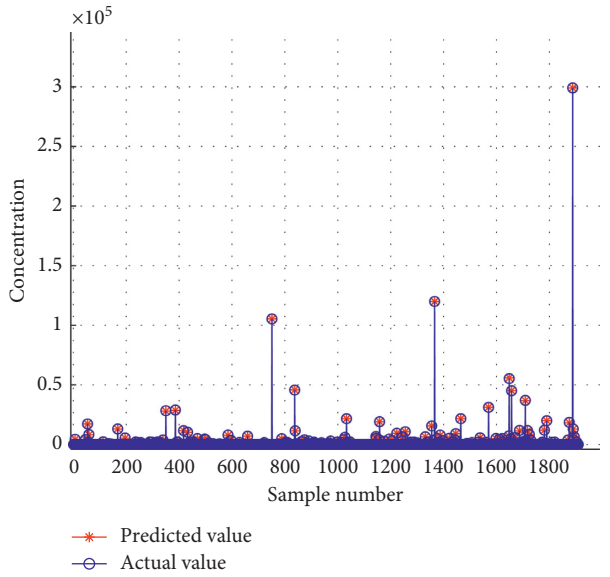


Figure 10: The fitting effect diagram of the Jining training set data.

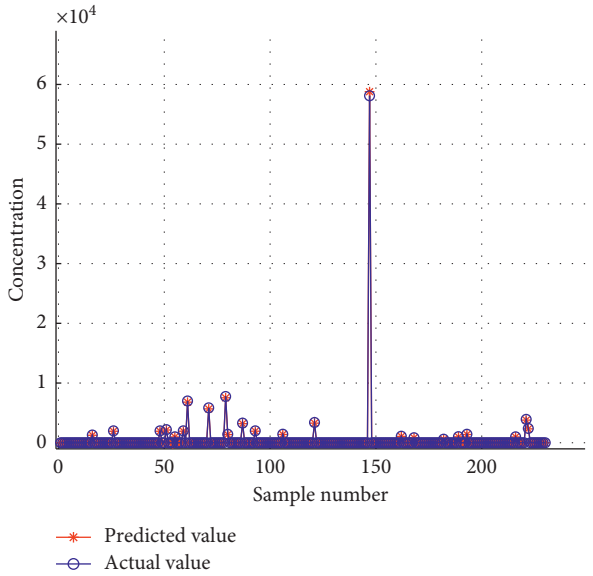


Figure 9: The Binzhou test set data prediction effect diagram.

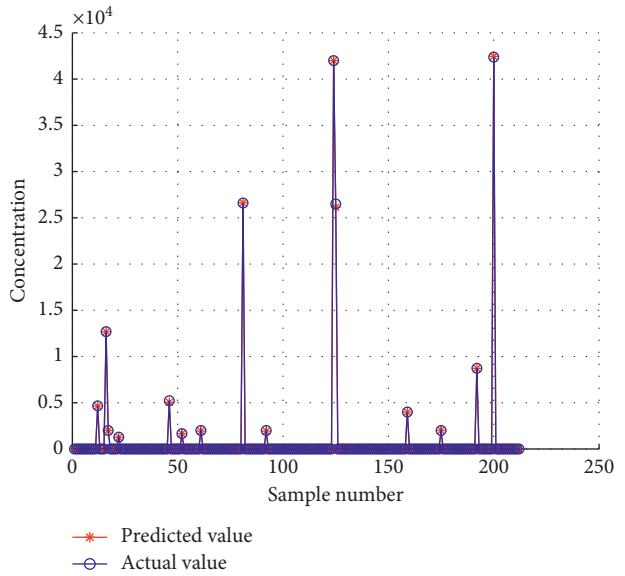


Figure 11: The Jining test set data prediction effect diagram.

with zero claims. Most of the data for claims in the training set of this city are small claim amounts and larger claim amounts. The fitting values of the network to claim data are consistent with the actual values, indicating that the network has a good fitting effect.

It can be seen from Figure 15 that claim amounts in Weihai's prediction set vary. The predicted value of the network for each piece of claim data is consistent with the actual value. Again, the predicted values and actual values of the data for which no claim has occurred are stacked at the bottom of the image. The overall prediction accuracy of the network for the prediction set data is 98.83%, and the prediction performance is good.

4.5. Jinan City. There are 4370 pieces of data in Jinan City. By fitting and forecasting the data of Jinan City, the following training result figure and prediction result figure are obtained, as shown in Figures 16 and 17.

Figure 16 shows the comparison between fitted values and actual values of the trained network on the input data of the training set. Since there are more data in this prefecture-level city, the proportion of data without claims is larger, so the lower part of the graph appears to have a heavy accumulation phenomenon, and at the same time, fitted values of the claim data are consistent with actual values, and the network fits well. Besides, there are many claims in this data set, and the network still has a good fit for all claim data.

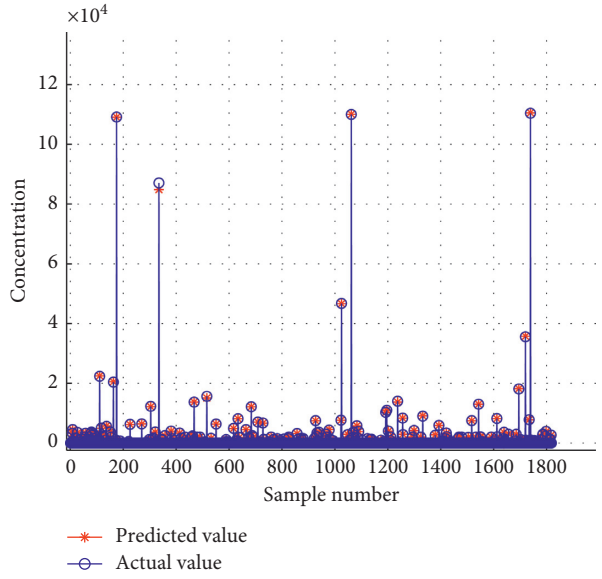


Figure 12: The fitting effect diagram of Weihai training set data.

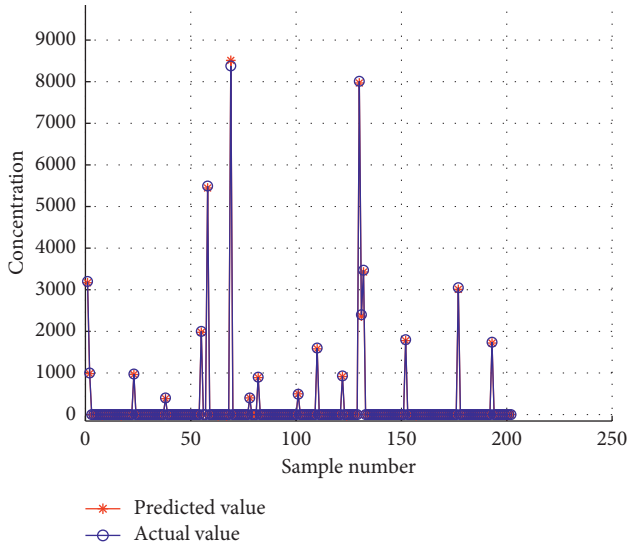


Figure 13: The Weihai test set data prediction effect diagram.

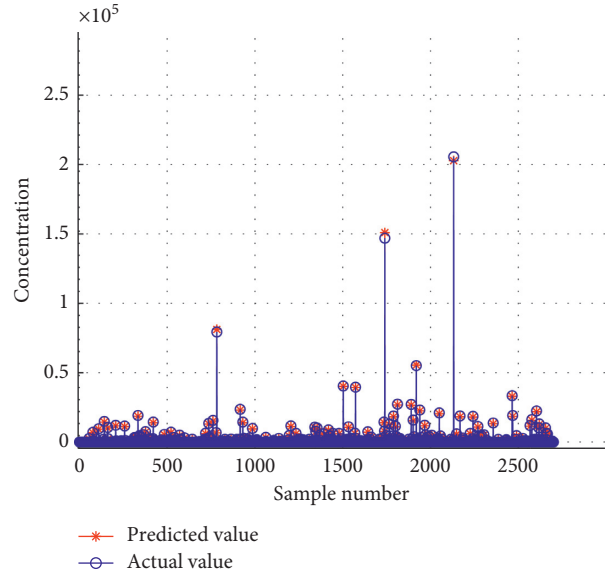


Figure 14: The fitting effect diagram of Weifang training set data.

Figure 17 shows the comparison between predicted values and actual values of the input data of the test set by the network. This prefecture-level city has more centralized claim data, and the network has made accurate predictions for most of the claim data. The prediction for zero claims is accumulated at the bottom of the graph. The overall prediction accuracy of the network for the prediction set data is 99.96%, and the prediction effect is very good.

4.6. Laiwu City. There are 2600 pieces of data in Laiwu City. By fitting and forecasting the data of Laiwu City, the following training result figure and prediction result figure are obtained, as shown in Figures 18 and 19.

Figure 18 shows the fitting effect of the training set in Laiwu City. The fitted value of the network for each piece of claim data can accurately correspond to the actual value. The predicted and actual values of the zero claim data are stacked at the bottom of the graph, and the network has a good fitting effect.

Figure 19 is the prediction output of the trained BP neural network on the Laiwu City prediction data set. It shows that the network can accurately predict the claim data in the prediction data set. The overall prediction accuracy of the network for this prediction set data is 99.98%, and the overall prediction performance of the network is good.

In summary, the BP neural network model not only can accurately fit and predict the claim data of the entire

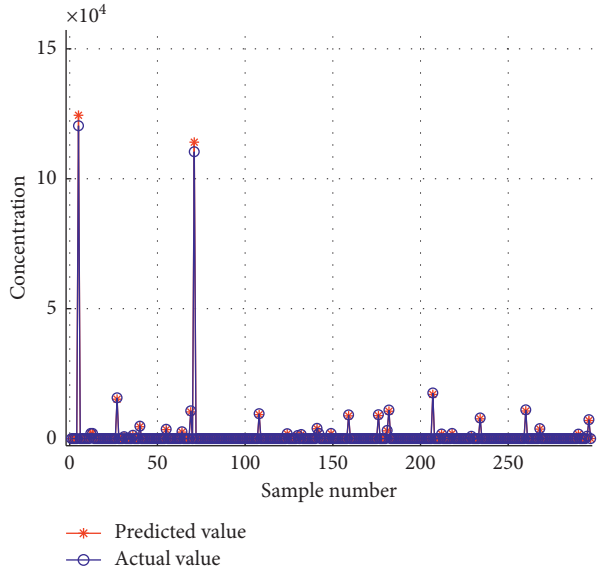


Figure 15: The Weifang test set data prediction effect diagram.

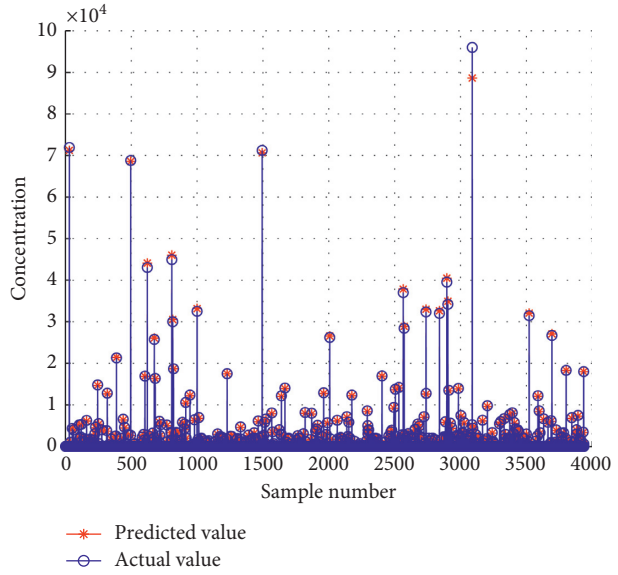


Figure 16: The fitting effect diagram of Jinan training set data.

Shandong Province but also can accurately fit and predict the data of six prefecture-level cities. Besides, the trained BP network has a prediction accuracy of over 95% for each prefecture-level city, indicating that it is reasonable and accurate to use the BP neural network model to fit and predict the aggregate claim amount of automobile insurance. Moreover, the BP neural network fits and predicts the data of six prefecture-level cities with different data characteristics well, indicating that the BP neural network can adapt to data with different characteristics in different regions. As the BP neural network has a strong tolerance, the network can be used to fit and predict data nationwide.

5. Premium Ratemaking

5.1. Model Introduction. Credibility theory is the study of how to reasonably use prior information and individual claim experience to estimate, predict, and formulate posterior insurance premiums. The posterior premium estimate is calculated as follows:

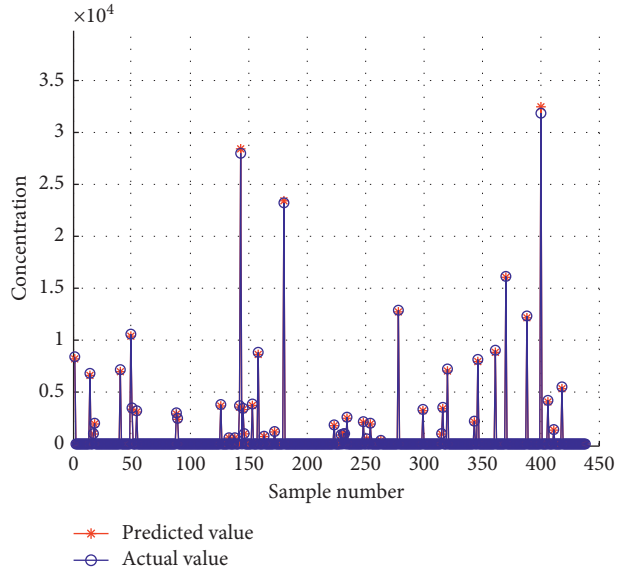


Figure 17: The Jinan test set data prediction effect diagram.

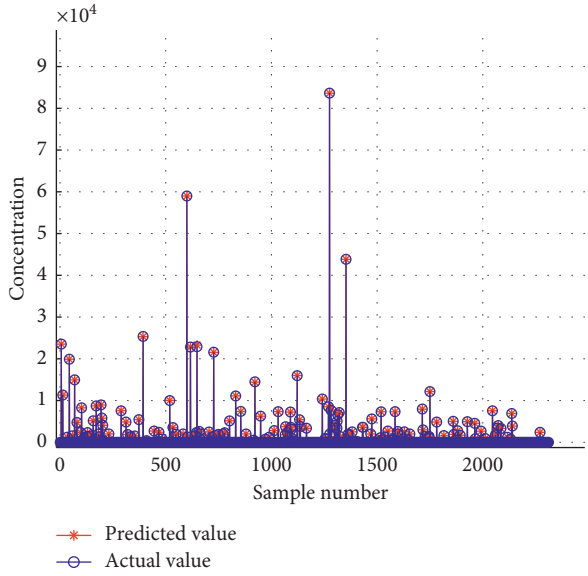


Figure 18: The fitting effect diagram of Laiwu training set data.

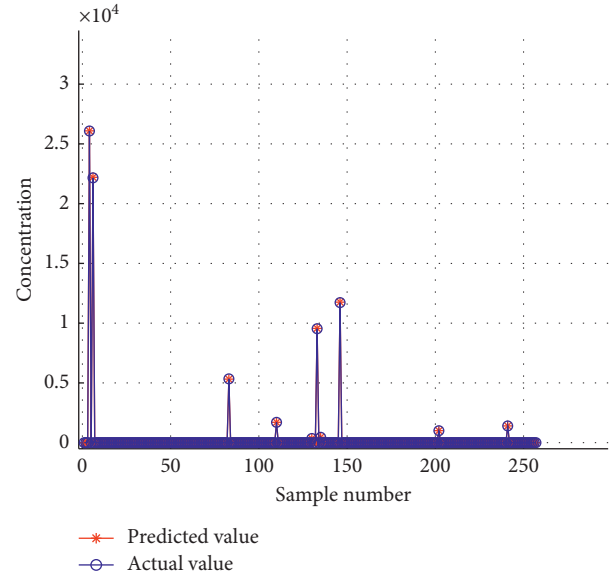


Figure 19: The Laiwu test set data prediction effect diagram.

$$\text{Estimation for posterior premium} = Z * \text{Experience value} + (1 - Z) * \text{Prior value}, \tag{9}$$

where Z $(0 < Z < 1)$ is the reliability factor. The posterior premium estimate is called the reliability estimate. Only by choosing the reliability factor correctly, the adjusted insurance premium can then be closed to its actual risk level. In addition to insurance premiums, the credibility theory can also be used to estimate the number of claims, total claim amounts, loss ratio, and relative number of levels. There are two types of credibility models: the classical reliability model and most accurate reliability model.

The classical reliability model attempts to limit the influence of random fluctuations in the observed data on the estimated value, which is also called the limited-fluctuation reliability theory. In the classical reliability model, it is necessary to determine when the individual risk reaches a certain scale, and the reliability factor = 1, that is, the empirical data is fully credible. This scale becomes the "full credibility standard," and the reliability factor less than 1 is called partial credibility. Suppose the data volume of the individual risk is n; then, n_f is the full credibility standard. If $n > n_f$, then the reliability factor $Z = 1$. If $n < n_f$, then $Z = \sqrt{n/n_f}$.

The most accurate reliability model is the so-called least squares reliability model. This model determines the reliability factor by minimizing the sum of squared errors

between the estimated value and actual value and emphasizes the accuracy of the estimated results, mainly including the Buhlmann reliability model and Buhlmann–Straub reliability model. The Buhlmann reliability model assumes that the scale of the individual risk remains the same. If n represents the empirical period, that is, the number of years of observation of empirical data, the reliability factor Z is given as $Z = n/(n + k)$, and k is called the Buhlmann parameter, which is the ratio of the mean of the process variance (EPV) to the variance of the hypothetical mean (vhm). In the Buhlmann–Straub reliability model, the scale of the individual risk can be changed. The Buhlmann reliability model is a simplified form of it. In the most accurate reliability model, empirical data and prior data both have significant influence on the reliability factor. For equally important influences, the reliability factor can only be found when empirical data and prior data are determined.

5.2. The Buhlmann Model in Nonparametric Estimation. Since nonparametric estimation does not require the use of overall information (the overall distribution and some parameter characteristics of the population), the distribution type of the population does not need to be assumed, and we can directly perform statistical testing on the distribution of the population, so, in this paper, we use the Buhlmann model from nonparametric estimation to calculate pure premiums.

Since the BP neural network can accurately predict the claim information for each of the insured and in the previous empirical analysis, we have proved that the BP neural network can accurately predict the total claim amount of the insured. Therefore, we use the obtained claim prediction value as empirical data to determine individual premiums. The following is based on the actual claim data and predicted data of the six prefecture-level cities to calculate the premium.

First of all, we use the average actual claim amounts μ of the six prefecture-level cities as prior data and the claims of different individuals predicted by the BP neural network X_{ij} as empirical data, and by applying the formula $P_i = Z_i \times \overline{X}_i + (1 - Z_i)\mu$, the average net premiums of the corresponding six prefecture-level cities P_i are obtained, where i represents the i^{th} prefecture-level city, Z_i is the reliability factor of the i^{th} prefecture-level city, $\overline{X}_i$ is the average predicted claim amount of the i^{th} prefecture-level city, X_{ij} is the predicted claim amount of the j^{th} insured from the i^{th} prefecture-level city, Y_{ij} indicates the actual claim amount of the j^{th} insured from the i^{th} prefecture-level city, and n_i indicates the number of people in the training set of the i^{th} prefecture-level city. $i = 1, 2, \ldots, 6$; $j = 1, 2, \ldots, n_i$.

The average value of predicted claims for the i^{th} prefecture-level city in Shandong Province is calculated as $\overline{X}_i = (9/n_i) \sum_{j=1}^{(n_i/9)} X_{ij}$.

The average value of predicted claims of six prefecture-level cities in Shandong Province is $\overline{X} = (1/6) \sum_{i=1}^{6} \overline{X}_i$.

The actual mean value of claims in the training set of the i^{th} prefecture-level city in Shandong Province is $\mu_i = E(\overline{Y}_i) = E((1/n_i) \sum_{j=1}^{n_i} Y_{ij})$.

The average value of actual claims in the training set of six prefecture-level cities in Shandong Province is $\mu = E(\overline{Y}) = (1/6) \sum_{i=1}^{6} E(\overline{Y}_i) = (1/6) \sum_{i=1}^{6} \mu_i$.

The variance of the predicted claim amount for the i^{th} prefecture-level city is $v_i = \text{Var}(X_{ij})$.

The variance of the forecast claim amount in Shandong Province is as follows:

$$v = \frac{1}{6} \sum_{i=1}^{6} v_i = \frac{1}{6} \sum_{i=1}^{6} \text{Var}\left(X_{ij}\right), \tag{10}$$

and we have

$$\text{Var}\left[\overline{X}_i\right] = E\left[\text{Var}\left(\overline{X}_i | \Theta_i\right)\right] + \text{Var}\left[E\left(\overline{X}_i | \Theta_i\right)\right] = \frac{v}{n} + a, \tag{11}$$

where the estimated value for the structural parameter is μ_i, size of risk X_i is measured by Θ_i, and v and a are calculated as follows:

(1) $\widehat{\mu}_i = \overline{Y}_i = (1/n_i) \sum_{j=1}^{n_i} Y_{ij}$ is the estimated mean value of actual claims in the i^{th} prefecture-level city

(2) $\widehat{v}_i = v_i = \text{Var}(X_{ij})$ is the variance of the predicted claim amount for the i^{th} prefecture-level city

(3) $\widehat{v} = (1/6) \sum_{i=1}^{6} \widehat{v}_i$ is the variance of the forecast claim amount of six prefecture-level cities in Shandong Province

$$\widehat{a} = \frac{1}{5} \sum_{i=1}^{6} \left(\overline{X}_i - \overline{X}\right)^2 - \frac{\widehat{v}}{n}. \tag{12}$$

The pure premium of each prefecture-level city is calculated as follows:

$$P_i = \widehat{Z}_i \overline{X}_i + \left(1 - \widehat{Z}_i\right)\mu, \quad i = 1, 2, \ldots, 6, \tag{13}$$

where

$$\widehat{Z}_i = \frac{n_i}{n_i + \widehat{k}}, \qquad \widehat{k} = \frac{\widehat{v}}{\widehat{a}}. \tag{14}$$

5.3. Empirical Analysis of Premium Ratemaking. Through the Buhlmann model, we find the parameter values for each prefecture-level city and calculate the pure premium and corresponding aggregated pure premium for each prefecture-level city according to the corresponding parameters.

From Table 2, it can be seen that the training of the BP neural network requires a certain amount of data. Generally, the larger the amount of data, the better the performance of the trained network and the higher the accuracy of the predicted data. The Z value increases with the increase of the amount of data, which means that, as the amount of data increases, the predicted value obtained through the BP neural network weighs more in the determination of the

Table 2: The comparison between the predicted pure premium and actual total claim.

	Jinan City	Binzhou City	Jining City	Laiwu City	Weihai City	Weifang City
Z value	0.4627	0.2382	0.2245	0.2589	0.2486	0.2897
Size of prediction set	438	230	213	260	202	300
Average pure premium	521.14	630.85	629.44	602.09	599.8	574.78
Average claim amount in prediction set	434.19	633.87	527.16	531.09	480.37	468.83
Profit percentage	0.2	−0.004	0.19	0.13	0.24	0.22

Table 3: Pure premiums for some individuals.

Predicted claim amount X_{ij}	Premium calculated P_{ij}
0	449.462
0	449.462
6162	1917.2504
0	449.462
0	449.462
3533	1291.0226
0	449.462
0	449.462
1653	843.2066
.....	.....

premium, which makes the premium ratemaking more reasonable for individuals with different risks.

Besides, the average pure premium of each prefecture-level city calculated by the above model can ensure that insurance companies do not lose money in the automobile insurance business. The average net premiums calculated by Jinan and Binzhou are almost the same as the actual average claims, while the average net premiums calculated by Jining, Laiwu, Weihai, and Weifang are slightly higher than the actual average claims. This shows that our premium ratemaking model is reasonable and can be used as a new idea for automobile insurance companies to determine premium rates.

Based on the parameters obtained from the above model of prefecture-level cities, we can find the pure premiums that each individual should pay in the six prefecture-level city prediction sets. The calculation formula is as follows:

$$P_{ij} = \widehat{Z}_i \times X_{ij} + \left(1 - \widehat{Z}_i\right)\overline{X}_i, \tag{15}$$

where P_{ij} is the pure premium for the j^{th} individual in the i^{th} prefecture-level city, $\widehat{Z}_i$ is the reliability factor of the i^{th} prefecture-level city, and X_{ij} is the predicted claim amount of the j^{th} individual in the i^{th} prefecture-level city. Laiwu City has been taken as an example to find the pure premiums for individuals. Since the insured usually takes into account no claims preferential treatment rules in real life, this paper also takes this situation into account, and then, when the individual's predicted claim amount is less than 200, it is recorded as 0. The calculated results are shown in Table 3.

Table 3 shows the premium calculated for selected individuals with different risks. From Table 3, we can see that, for individuals with lower predicted claims, the calculated pure premiums are relatively low, while for individuals with higher predicted claims, the premiums are relatively high. This model can effectively identify risks and helps to charge the insured with premiums that are compatible with their risks, making premium ratemaking more reasonable and fair.

6. Conclusion

A brand new method and idea to price for automobile insurance is attempted in this paper. Different from the traditional model, we did not separately model the number of claims and the individual claim amount, but directly modeled the aggregate claim amount. This idea is simple and straightforward, and the results are clear. Based on the BP neural network model in Matlab, this paper fits and predicts 29,000 valid data in Shandong Province, which shows that the BP neural network can predict the aggregate amount of automobile insurance claims very accurately. In addition, in order to break the regional limitations of the data, we fitted and predicted the data of each prefecture-level city separately with the prefecture-level city as a unit. It shows that the network is very inclusive of data and can break geographical restrictions; hence, it can be applied to nationwide data.

Given that the fitted and predicted aggregate claim amounts are accurate, this paper uses the credibility theory to calculate the average pure premiums for six prefecture-level cities. By using the average aggregate claim amount of the training set data of the entire Shandong Province, the average pure premium of each prefecture-level city is adjusted so that the overall automobile insurance compensation situation of Shandong Province can be considered, and the individual compensation situation of each prefecture-level city can be highlighted. The result shows that the pure premium calculated by this method can improve the profitability of the automobile insurance business.

Taking into account the different risks and aggregate claim amount of each of the insured, this paper aims to find appropriate net premiums for each of the insured corresponding to their risks. Using the reliability factor of individual's location-level city and the average claim amount of the prefecture-level city to calculate each individual's pure premium and trying to personalize the premium rate, through this calculation method, individual risks can be identified to a certain extent, and the premium rates can be differentiated to make the determination of car insurance rates more reasonable and fair.

The BP neural network is introduced and applied to the field of automobile insurance, which has the important application value for pricing of automobile insurance rates. The combination of the BP neural network and Buhlmann model can calculate the pure premium that matches the liability of the insurance company, which can effectively

improve the current situation of the loss of the insurance company's automobile insurance business, thereby boosting the enthusiasm and creativity of the insurance company to develop more and better products to promote the soundness of the country's automobile insurance industry.

To avoid the shortcomings of the BP neural network such as slow convergence speed and easy to fall into a local minimum, the genetic algorithm is selected to optimize the BP neural network in this paper. The genetic algorithm has a strong adaptive and optimal ability, which can effectively improve the convergence speed of the BP neural network and prevent the network from falling into a local minimum. At the same time, by considering the overfitting problem of the BP neural network, this paper chooses to use the early stop method to make the network error continue to decrease and stabilize, effectively avoiding the overfitting problem.

Data Availability

The data used to support the findings of this work 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 supported by the National Natural Science Foundation of China (no. 11301303), National Social Science Foundation of China (no. 15BJY007), Taishan Scholars Program of Shandong Province (no. tsqn20161041), Humanities and Social Sciences Project of the Ministry Education of China (no. 19YJA910002), Natural Science Foundation of Shandong Province (no. ZR2018MG002), Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions (no. 1716009), Shandong Provincial Social Science Project Planning Research Project (no. 19CQXJ08), Risk Management and Insurance Research Team of Shandong University of Finance and Economics, Excellent Talents Project of Shandong University of Finance and Economics, Collaborative Innovation Center Project of the Transformation of New and Old Kinetic Energy and Government Financial Allocation, Shenzhen Peacock Program (no. 000417), Shandong Jiaotong University "Climbing" Research Innovation Team Program, and 1251 Talent Cultivation Project of Shandong Jiaotong University.

References

[1] R. A. Bailey and L. J. Simon, "Two studies in automobile insurance ratemaking," *ASTIN Bulletin*, vol. 1, no. 4, pp. 192–217, 1960.

[2] D. Denneberg, "Premium calculation: why standard deviation should be replaced by absolute deviation," *ASTIN Bulletin*, vol. 20, no. 2, pp. 181–190, 1990.

[3] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[4] D. Samson and H. Thomas, "Linear models as aids in insurance decision making: the estimation of automobile Insurance Claims," *Journal of Business Research*, vol. 15, no. 3, pp. 247–256, 1987.

[5] G. K. Smyth, "Generalized linear models with varying dispersion," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 51, no. 1, pp. 47–60, 1989.

[6] K. J. Stroiński and I. D. Currie, "Selection of variables for automobile insurance rating," *Insurance Mathematics and Economics*, vol. 8, no. 1, pp. 35–46, 1989.

[7] S. W. Meng, "An application of generalized linear model to automotor insurance pricing," *Application of Statistics and Management*, vol. 26, no. 1, pp. 24–29, 2007.

[8] N. R. Draper, "Generalized linear models for insurance data by Piet de Jong, Gillian Z. Heller," *International Statistical Review*, vol. 76, no. 2, p. 315, 2008.

[9] M. Q. Zhao and Y. P. Chen, "The auto insurance ratemaking based on double generalized liner models and the comparison with generalized linear models," *Insurance Studies*, vol. 10, pp. 32–41, 2016.

[10] G. K. Smyth and B. Jørgensen, "Fitting tweedie's compound Poisson model to insurance claims data: dispersion modelling," *ASTIN Bulletin*, vol. 32, no. 1, pp. 143–157, 2002.

[11] K. Antonio and J. Beirlant, "Actuarial statistics with generalized linear mixed models," *Insurance: Mathematics and Economics*, vol. 40, no. 1, pp. 58–76, 2007.

[12] E. W. Frees, P. Shi, and E. A. Valdez, "Actuarial applications of a hierarchical insurance claims model," *ASTIN Bulletin*, vol. 39, no. 1, pp. 165–197, 2009.

[13] X. H. Wang, S. W. Meng, and Y. S. Wang, "Automobile insurance pricing models based on heavy-tailed loss distribution and its application," *Insurance Studies*, no. 4, pp. 67–78, 2017.

[14] L. Z. Zhang and H. Y. Xie, "The application of Tweedie distribution in auto insurance ratemaking," *Insurance Studies*, no. 1, pp. 80–90, 2017.

[15] M. Ayuso, M. Guillén, and J. P. Nielsen, "Improving automobile insurance ratemaking using telematics: incorporating mileage and driver behaviour data," *Transportation*, vol. 46, no. 3, pp. 735–752, 2019.

[16] Y. F. Huang and S. W. Meng, "Automobile insurance classification ratemaking based on telematics driving data," *Decision Support Systems*, vol. 127, Article ID 113156, 2019.

[17] P. L. Brockett, X. Xia, and R. A. Derrig, "Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud," *The Journal of Risk and Insurance*, vol. 65, no. 2, pp. 245–274, 1998.

[18] Y. Liu, B. J. Wang, and S. G. Lv, "Using multi-class adaboost tree for prediction frequency of auto insurance," *Journal of Applied Finance and Banking*, vol. 4, no. 5, pp. 45–53, 2014.

[19] T. Mzhavia, *Vehicle Insurance Claim Data Study and Forecasting Model Using Artificial Neural Networks*, Tallinn University of Technology, Tallinn, Estonia, 2016.

[20] M. V. Wüthrich, "Covariate selection from telematics car driving data," *European Actuarial Journal*, vol. 7, no. 1, pp. 89–108, 2017.

[21] L. Z. Zhang and D. Wang, "A research on the rate making of automobile insurance with big data—modeling of automobile insurance claim severity based on SOM neural network," *Insurance Studies*, no. 9, pp. 56–65, 2018.

[22] Y. C. Lin, J. Li, M.-S. Chen, Y.-X. Liu, and Y.-J. Liang, "A deep belief network to predict the hot deformation behavior of a ni-based superalloy," *Neural Computing and Applications*, vol. 29, no. 11, pp. 1015–1023, 2016.

[23] Y. C. Lin, Y. J. Liang, M. S. Chen, and X. M. Chen, "A comparative study on phenomenon and deep belief network models for hot deformation behavior of an Al–Zn–Mg–Cu Alloy," *Applied Physics A*, vol. 123, p. 68, 2016.

[24] Y. C. Lin, J. Huang, H.-B. Li, and D.-D. Chen, "Phase transformation and constitutive models of a hot compressed TC18 titanium alloy in the $\alpha + \beta$ regime," *Vacuum*, vol. 157, pp. 83–91, 2018.

[25] D.-D. Chen, Y. C. Lin, and F. Wu, "A design framework for optimizing forming processing parameters based on matrix cellular automaton and neural network-based model predictive control methods," *Applied Mathematical Modelling*, vol. 76, pp. 918–937, 2019.

Hindawi
Complexity
Volume 2020, Article ID 8858258, 20 pages
https://doi.org/10.1155/2020/8858258

Research Article

Bidders Recommender for Public Procurement Auctions Using Machine Learning: Data Analysis, Algorithm, and Case Study with Tenders from Spain

Manuel J. García Rodríguez, Vicente Rodríguez Montequín, Francisco Ortega Fernández, and Joaquín M. Villanueva Balsera

Project Engineering Area, University of Oviedo, Oviedo 33012, Spain

Correspondence should be addressed to Vicente Rodríguez Montequín; montequi@uniovi.es

Received 14 September 2020; Revised 9 November 2020; Accepted 11 November 2020; Published 25 November 2020

Academic Editor: Thiago Christiano Silva

Recommending the identity of bidders in public procurement auctions (tenders) has a significant impact in many areas of public procurement, but it has not yet been studied in depth. A bidders recommender would be a very beneficial tool because a supplier (company) can search appropriate tenders and, vice versa, a public procurement agency can discover automatically unknown companies which are suitable for its tender. This paper develops a pioneering algorithm to recommend potential bidders using a machine learning method, particularly a random forest classifier. The bidders recommender is described theoretically, so it can be implemented or adapted to any particular situation. It has been successfully validated with a case study: an actual Spanish tender dataset (free public information) which has 102,087 tenders from 2014 to 2020 and a company dataset (nonfree public information) which has 1,353,213 Spanish companies. Quantitative, graphical, and statistical descriptions of both datasets are presented. The results of the case study were satisfactory: the winning bidding company is within the recommended companies group, from 24% to 38% of the tenders, according to different test conditions and scenarios.

1. Introduction

The largest adjudicators of a country, by number of projects and by cost, are public procurement agencies. For example, public authorities in the European Union spend around 14% of GDP (around €2 trillion) on public procurement [1] every year. The definition of public procurement is the purchase of goods, works, or services by a public agency. Public procurement is clearly important to politicians, citizens, researchers, and companies because of its size. On the other hand, the European open data market size (products and services enabled by open data) was €184.45 billion in 2019, according to the official European Data Portal [2]. High growth is expected in the near future. The availability of open data in public procurement announcements (also known as tenders) enables the building of a bidders recommender.

The bidders recommender may be a strategic tool for improving the efficiency and competitiveness of organisations and is particularly suitable for the two main stakeholders: suppliers and public procurement agencies. On the one hand, it is useful to the supplier because it assists in identifying the most suited tenders, i.e., those that they should prioritise. On the other hand, the contracting agency could automatically search companies with a compatible profile for the tender's announcement, e.g., selective tendering where suppliers are only allowed by invitation. Thus, it could be called a "bidders search engine" or a "bidders recommender."

Many public agencies do not easily obtain competitive offers when they publish public procurement announcements. It is a serious problem with negative consequences for the project in terms of cost, quality, lifetime,

sustainability, etc. A bidders recommender would produce significant benefits as follows:

(i) Tenders with more bidders have lower award prices and, consequently, the public agencies will reduce costs. This relationship is quantitatively demonstrated for Spanish tenders in this paper, but there are more empirical studies, e.g., in Italy [3] and the Czech Republic [4, 5].

(ii) This new tool will provide support to small- and medium-sized enterprises (SMEs), which play a crucial role in most economies. It will make it easier and more efficient for SMEs to access procurement auctions, promote inclusive growth, and support principles such as equal treatment, open access, and effective competition [6].

(iii) In scenarios of high participation, it is more difficult to generate corruption or collusive tendering (where the bidders do not compete honestly).

The main objective of this paper is to propose an algorithm to search for suppliers (companies) to invite to tender. Discovering the number and identity of bidders is challenging, since there does not exist a suitable quantitative model to forecast the identities of a single or a group of specific key competitors likely to submit a future tender [7]. So, the input parameters of the bidders recommender have to have the tender's announcement but also be a generic algorithm that can be implemented or adapted to any particular situation. The main issue is to get information about bidders and the rest of the companies in the market because in many countries, the information is not public or free.

Some papers have proposed similar tools, but only the tenders are characterised or analysed, not the bidders, e.g., a product search service [8] or a similar tenders engine search (comparison of one tender to all other tenders according to specific criteria) [9]. Our work is based on the profile of the winning companies rather than the characteristics of the tender. Thus, this paper is a novel study which brings a new and modern perspective to gathering tenders and bidders. The bidders recommender has used tenders that have been published in Spain. In particular, the tender dataset has 102,087 Spanish tenders from 2014 to 2020. All types of works are included, not only construction auctions (which are the favourite subjects in the public procurement literature, for several reasons). The company dataset has 1,353,213 Spanish companies to search suitable bidders. In [10, 11], the Spanish public procurement system as well as the European and national legislation is described, and they have also analysed Spanish tenders for other purposes.

The application of this pioneering bidders recommender by public procurement agencies or potential bidders is summarised in Figure 1. It has three sequential steps or phases, and the input is obviously a new public procurement announcement, also known as a tender notice. Initially, it is based on forecasting the winning company of the tender thanks to a machine learning method called a random forest classifier model. This classification model has previously been trained with lots of tenders and their respective winning companies. The second phase is to add the business information of the forecast winning company for creating a profile of a winning company. The business information is in the company dataset (data from the Business Register). Finally, similar or compatible companies are searched, according to their profile, where the search criteria are filters or fixed rules.

The paper is structured as follows. Section 2 summarises the literature review associated with the bidders recommender in public procurement auctions. Section 3 presents the fields of the dataset and the machine learning algorithm (called random forest classifier) which will be used in the recommender. Furthermore, the bidders recommender is explained in detail (Section 3.5) and some evaluation metrics are defined to measure the accuracy of detecting the winning company of the tender within the group of bidders. Section 4 quantitatively describes the datasets for the real case study from Spain to test the bidders recommender. It is tested under different scenarios, and the results are presented in Section 4.3. In Section 5, the recommender is discussed from a general perspective to be applied to other countries or datasets. Finally, some concluding remarks, limitations, and avenues for future research are presented in Section 6.

2. Literature Review

This paper involves (either directly or indirectly) diverse topics such as open government data, public procurement and its regulation, machine learning, tender evaluation, prediction techniques, business registers, and so on. The bidders recommender has a multidisciplinary nature which fills a gap in the literature. Nevertheless, the key components have an extensive literature which will be summarised in the following paragraphs.

In this article, we used open data and, especially, Open Government Data (OGD). The OGD initiatives have grown very strongly in the academic field [12–14]. That is to say, open data are produced by governmental entities in order to promote government transparency and accountability. Hence, there are different stakeholders, user groups, and perspectives [15, 16]. The OGD is a part of the public value of e-government [17], and it is a new and important resource with economic value [18, 19]. For example, data.europe.au and *data.gov* are online portals that provide open access datasets in a machine-readable format [20] and are generated by the European Union and the United States of America public agencies, respectively. However, there are challenges and risks in dealing with the data quality of open datasets (quality over quantity) [21] and this article suffers from these too. It is very important to measure the transparency and the metadata quality in the open government data portals [22–24].

Other public procurement fields that have recently sparked the interest of governments, policy makers, and researchers are Big and Open Linked Data (BOLD) [25], the growing awareness of public procurement as an innovation policy tool [26], and the role of e-government in sustainable public procurement [27].

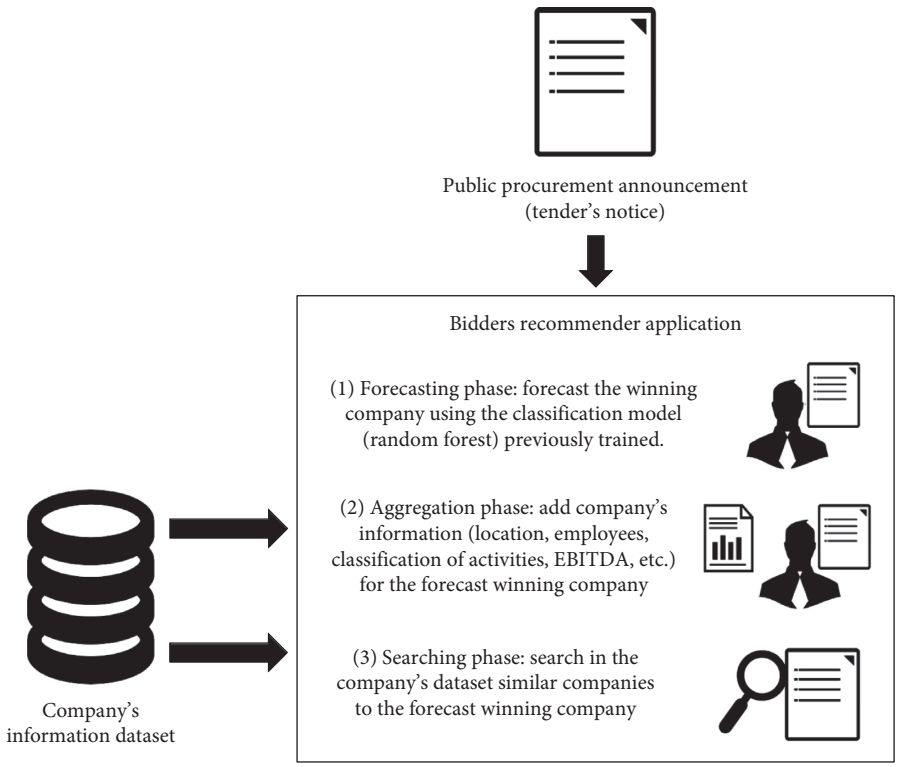


Figure 1: Flowchart of the application of the bidders recommender for a new tender.

This article uses a machine learning algorithm. The big data and machine learning technologies can be used for econometrics [28, 29], enterprises [30], tender evaluation [31], or analysis of public procurement notices [32]. Therefore, this paper follows the trends in the literature.

There is extensive literature about tender evaluation (also called bidding selection methods) for the selection of the optimal supplier in public procurement [33] with different techniques such as the economic scoring formulas [34], data envelopment analysis [35] or multicriteria decision making [36, 37], and where multiple bidders are evaluated on the basis of price and quality [38]. In particular, the most studied public procurement auctions are related to construction, i.e., distribution of bids [39], bidding competitiveness and position performance [40], strategic bidding [41], tender evaluation and contractor selection [42, 43], and empirical analysis in countries such as Slovakia [44]. There are almost no studies which include all kinds of business sectors and a large volume of tenders. However, this article has a holistic approach due to the large tender dataset of all sectors.

Another relevant subject in the public procurement literature is the detection of collusive tendering or bid rigging [45] with case studies in Spain [46], India [47], and Hungary [48]. This occurs when businesses that would otherwise be expected to compete secretly conspire to raise prices or lower the quality of goods or services for purchasers in a public procurement auction (this is called a cartel). In addition, public procurement contracts have other issues such as optimal quality [49], too many regulations [50], systemic risk [51], or corruption [52–54]. Corruption is a form of dishonesty undertaken by a person or organisation with the authority to acquire illicit benefit. There are empirical studies to detect corruption by analysing public tenders in many countries, for example, in China [55], Russia [56], the Czech Republic [57], and Hungary [58]. The application of algorithms by governments or enterprises to detect collusion or corruption [59], especially using machine learning methods [60–62], has become an almost inevitable topic and the subject of numerous studies. Indirectly, this article could create a useful tool for these topics since it is able to forecast the most probable winning bidders and, therefore, the detection of unlikely winners too.

Forecasting and prediction techniques are widely studied and applied in the academic field of public procurement auctions. In [63], the mathematical relationship between scoring parameters in tendering is studied because, among other reasons, it is useful for the bid tender forecasting model [64]. There are some notable key parameters which have been analysed in the forecasting literature, especially for construction auctions, from traditional techniques to new machine learning methods, for example, the probability of bidder participation [7], an award price estimator [10, 65, 66], or cost estimator [67, 68]. However, as far as we know, this article is the first attempt to forecast the winning company for all tenders in a country.

In conclusion, this paper creates a smart search engine to recommend a group of companies for each tender, according to the forecast winning company. This means they have a similar business, technical, and economic profiles. Therefore, it is necessary to find these profiles in the Business Registers [69, 70] or other databases where the company's annual accounts are available. For instance, it is even possible to forecast

the corporate distress using machine learning in such reports [71]. The analysis of a company's profile has the same basis as the academic topic called bankruptcy prediction. This is the measurement of corporate solvency and the creation of prediction models [72] to forecast the company failure or distress. It has been intensively discussed over the past decades [73], using traditional statistical techniques [74–76] or machine learning methods, such as gradient boosting [72], neural networks [77], support vector machine [78], or the comparison of different methods [79, 80].

3. Materials and Methods

This section describes the necessary components to create the bidders recommender proposed in this article. It is described theoretically so that it can be implemented in any country, not only in Spain. Section 3.1 presents the origin of the tender dataset and describes its fields, and, analogously, the company dataset is presented in Section 3.2. Section 3.3 explains the random forest classifier which is used in the first phase of the bidders recommender method. In Section 3.4, the evaluation metrics are defined to measure the recommender's accuracy. Finally, the bidders recommender algorithm is described in detail in Section 3.5.

3.1. Tender Dataset. The European and Spanish legislation on public procurement and on the reuse of public information is extensively detailed in [11]. The official website of the Public Sector Contracting Platform (P.S.C.P.) of Spain publishes the public procurement notices and their resolutions of all contracting agencies belonging to the Spanish Public Sector.

The P.S.C.P. has an open data section for the reuse of this information which will be used in this article to generate the tender dataset. The information is provided by the Ministry of Finance (the link is given in the Data Availability section) and has been published as open data since 2012. The fields, their descriptions, and the process to obtain the dataset are the same as discussed in [10]. However, these fields are shown in Table 1 for the convenience of the reader. A remarkable limitation is that only the identity of the winning company is known, not the rest of the bidders, and this will be a constraint for the recommendation system.

3.2. Company Dataset. In general, to obtain business information (companies' annual accounts) over several years is not easy or free. In Europe, Business Registers offer a range of services, which may vary from one country to another. However, the core services provided by all registers are to examine and store company information and to make this information available to the public [69]. *European Regulation 2015/884* [81] interconnects the Business Registers of the EU countries. The *European Business Registry Association* [82] has a list of Business Registers from around the world, for more information.

The authors have collected a dataset of annual accounts from Spanish companies, based on the information available in the Spanish Business Register. It is a public institution, but access is not free of charge. It is the main legal instrument for recording business activity: the company documents and submission of the annual accounts. The companies become a legal entity through their registration on the Business Register.

The fields of the company dataset are explained in Table 2. They can be divided into 5 headings: general information, human resources, location, accounting measures (operating income, EBIT, and EBITDA), and different systems for classifying industries or economic activities (CNAE, NACE2, IAE, US SIC, and NAICS). It should be noted that the company's annual accounts have more fields, but the authors have not been able to access and collect them. The fields of this dataset try to characterise the company from different points of view: main business activities (CNAE, NACE2, IAE, US SIC, and NAICS), nearby market (location), work capacity (employees), size (operating income), financial performance (EBITDA), etc. Not all of the fields have been used because they are not relevant to the analysis in this paper.

3.3. Random Forest Classifier. Random forest (RF), introduced by Breiman [83] in 2001, is an ensemble learning method for classification or regression that operates by constructing a multitude of decision trees at training times and outputting the class, which is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is a popular learning algorithm that offers excellent performance [84], no overfitting [85, 86], a versatility of applicability to large-scale problems and in handling different types of data [85, 87]. Particularly, Random Forest has been applied with remarkable success in tender datasets, for example in [10]. It provides its own internal generalisation error estimate, called the out-of-bag (OOB) error. Simplified algorithm of RF for classification [88] is summarized in Algorithm 1.

At each split in each tree, the improvement in the split criterion is the measure of the importance attributed to the splitting variable and is accumulated over all the trees in the forest separately for each variable. This is called "variable importance" [83].

3.4. Evaluation Metrics. It is necessary to define some error metrics to compare similar variables of the datasets and calculate the prediction error of the bidders recommender. The use of metrics based on medians and relative percentage is useful because the dataset has outliers of great weight, and the use of such metrics helps us to counteract the effect of these outliers. To compare variables of the dataset, the median absolute percentage error (MdAPE) was used, as defined in the following equation:

$$\text{MdAPE}\ (\%) = \frac{100}{\text{n}}\ \text{median}\left(\left|\frac{A_1 - F_1}{A_1}\right|, \left|\frac{A_2 - F_2}{A_2}\right|, \ldots, \left|\frac{A_\text{n} - F_\text{n}}{A_\text{n}}\right|\right), \tag{1}$$

where A_t is the actual value for period t, F_t is the expected value for period t, and n is the number of periods.

The following error metrics are to measure the prediction error of the RF classifier method for multiclass classification on imbalanced datasets [89]. Multiclass

Table 1: Most relevant data fields in the Spanish Public Procurement Notices (tenders) used in the dataset.

Name	Description	Name column dataset
Tender status	Status of the tender during the development of the procedure: prior notice, in time, pending adjudication, awarded, resolved, or cancelled	Not used (similar to Result_code)
Contract file number	Unique identifier for a contract file	Not used
Object of the contract	Summary description of the contract	Not used (unstructured textual information)
Public procurement agency	Public procurement agency that made the tender: name, identifier (NIF or DIR3), website, address, postal code, city, country, contact name, telephone, fax, e-mail, etc. CCAA is the Autonomous Community which is a first-level division in Spain. Latitude and longitude have been calculated from postal code, and they are not official fields in the notice.	Name_Organisation Postalzone CCAA Province Municipality Latitude Longitude
Tender price	Amount of bidding budgeted (taxes included)	Tender_Price
Duration	Time (days) to execute the contract	Duration
CPV classification	CPV (Common Procurement Vocabulary) is a European system for classifying the type of work in public contracts defined in the Commission Regulation (EC) No 213/2008: http://data.europa.eu/eli/reg/2008/213/oj The numerical code consists of 8 digits, subdivided into divisions (first 2 digits of the code), groups (first 3 digits), classes (first 4 digits), and categories (first 5 digits)	CPV CPV_Aggregated (first 2 digits of the CPV number)
Contract type	Type of contract defined by legislation (Law 9/2017): works, services, supplies, public works concession, works concession, public services management, services concession, public sector and private sector collaboration, special administrative, private, patrimonial, or others	Type_code
Contract subtype	Code to indicate a subtype of contract. If it is a type of service contract: based upon the 2004/18/CE Directive, Annex II. If it is a type of works contract: works contract codes defined by the Spanish DGPE.	Subtype_code
Contract execution place	Contract's execution has a place through the Nomenclature of Statistical Territorial Units (NUTS), created by Eurostat [47]	Not used (assumed equal to postalzone)
Type of procedure	Procedure by which the contracts was awarded: open, restricted, negotiated with advertising, negotiated without publicity, competitive dialogue, internal rules, derived from framework agreement, project contest, simplified open, association for innovation, derivative of association for innovation, based on a system dynamic acquisition, bidding with negotiation, or others	Procedure_code
Contracting system	The contracting system indicates whether it is a contract itself or a framework agreement or dynamic acquisition system	Not used
Type of processing	Type of processing: ordinary, urgent, or emergency	Urgency_code
Award result	Type of results: awarded, formalised, desert, resignation, and withdrawal	Result_code
Winner identifier (CIF)	Identifier of the winning bidder (called CIF in Spain) and its province (region)	CIF_Winner Winner_Province
Award price	Amount offered by the winning bidder of the contract (taxes included)	Award_Price
Date	Date of agreement in the award of the contract	Date
Number of received offers	Number of received offers (bidders participating) in each tender	Received_Offers

classification occurs when the input is classified into one, and only one, nonoverlapping class. An imbalanced dataset occurs when there is a disproportionate ratio of observations in each class.

Let $\widehat{y}_i$ be the predicted value of the $i-$th sample $(1 \le i \le n)$, y_i be the corresponding true value, ω_i be the corresponding sample weight, and L be the set of classes $(1 \le l \le L)$. Accuracy (2) is the proportion of correct predictions over n samples:

$$\text{accuracy} = \frac{1}{\text{n}} \sum_{i=1}^{n} 1\left(\widehat{y}_i = y_i\right), \tag{2}$$

where $1\left(\widehat{y}_i\right)$ is the indicator function. The equation returns a 1 if the classes match and 0 otherwise.

Balanced accuracy (3) avoids inflated performance estimates on imbalanced datasets:

$$\text{balanced accuracy} = \frac{1}{\sum_{i=1}^{n} \widehat{\omega}_i} \sum_{i=1}^{n} 1\left(\widehat{y}_i = y_i\right).\omega_i, \tag{3}$$

where $1\left(\widehat{y}_i\right)$ is the indicator function and $\widehat{\omega}_i = \omega_i / \sum_{j=1}^{n} 1\left(\widehat{y}_j = y_i\right).\omega_j$.

Let y_l be the subset of true values with class l. The precision (average macro) is calculated as follows:

$$\text{precision} = \frac{1}{L} \sum_{l=1}^{L} \frac{\left|y_l \cap \widehat{y}_l\right|}{\left|y_l\right|}. \tag{4}$$

Finally, the out-of-bag (OOB) is a method of measuring the prediction error in RF and other machine learning

Table 2: Data fields in the company's information database.

Name	Description	Name column dataset
Name company	Name of the company	Not used
CIF	CIF (for the Spanish term Certificado de Identificación Fiscal) is the company registration number. This identifier provides formal registration on the company tax system in Spain. In many countries, a company would be issued with a separate VAT number, while in Spain, the CIF also forms the VAT number.	CIF
Establishment date	It is the date on which the company starts its activities	Establishment_Date
Legal form	It is the entity type of company defined in the Spanish legal system. Mainly, there are two types: public limited company (PLC) and private company limited by shares (Ltd.)	Legal_Form
Last available year info	Last available year with economic information (operating income, EBIT, and EBITDA) of the company	Last_Available_Year_Info
Social capital	Minimum capital required to register the company in the legal system	Not used
Status company	Opened company (active) or closed company (inactive)	Status_Company
City, province, and country	City, province, and country of the company	City_Company Province_Company
Latitude and longitude	It represents the coordinates at geographic coordinate system of the company's location	Latitude_Company Longitude_Company
Web	Website of the company	Not used
President and CEO	President and Chief Executive Officer (CEO) of the company	Not used
Employees	Number of employees	Employees
Number group companies	Number of companies controlled (owned) by the company	Not used
Number investee companies	Number of companies in which the investor (company) makes a direct investment	Not used
Operating income	It measures the amount of profit realised from a business's operations, after deducting operating expenses (cost of goods sold, wages, depreciation, etc.). Value per year. Operating income = gross income − operating expenses = net profit + interest + taxes	Operating_Income
EBIT	Earnings before interest and taxes (EBIT) is a company's net income before interest and income tax expenses have been deducted. It is an indicator of a company's profitability. EBIT can be calculated as revenue minus expenses excluding tax and interest. The most important difference between operating income and EBIT is that EBIT includes any nonoperating income the company generates. Value per year. EBIT = net income + interest + tax	EBIT
EBITDA	Earnings before interest, taxes, depreciation, and amortization (EBITDA) is a measure of a company's overall financial performance. Value per year. EBITDA = net income + interest + taxes + depreciation + amortization = operating income + depreciation + amortization	EBITDA
Activity description	Textual description of the main business activities of the company	Not used
CNAE	CNAE (for the Spanish term Clasificación Nacional de Actividades Económicas) is the national classification of economic activities from Spain for statistical purposes. The last version of the CNAE has been adopted in 2009 (Royal Decree-Law 475/2007). It is equivalent to the European classification NACE2. It has primary and secondary codes.	CNAE_Primary CNAE_Secondary
NACE2	NACE2 (for the French term Nomenclature statistique des Activités Économiques dans la Communauté Européenne) is the statistical classification of economic activities in the European Community. The current version is revision 2 and was established by Regulation (EC) No 1893/2006. It is the European implementation of the United Nations (UN) classification ISIC (revision 4). There is a correspondence between NACE and ISIC. It has primary and secondary codes.	NACE2_Primary NACE2_Secondary
IAE	IAE (for the Spanish term Impuestos de Actividades Económicas) is the classification of economic activities in the Spanish Tax Agency for tax purposes. It has primary and secondary codes.	IAE_Primary IAE_Secondary
US SIC	The Standard Industrial Classification (SIC) is a system for classifying industries established in the United States (US) but also used by agencies in other countries. In the US, the SIC has been replaced by NAICS but some US government departments and agencies continued to use SIC codes. It has primary and secondary codes.	SIC_Primary SIC_Secondary
NAICS	The North American Industry Classification System (NAICS2017) is a classification of business establishments by type of economic activity (process of production). It has largely replaced the older SIC. It has primary and secondary codes.	NAICS_Primary NAICS_Secondary

(1) For $b = 1$ to B (number of trees):
(a) Draw a bootstrap sample $\mathbb{Z}^*$ of size N from the training data.
(b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{\min}$ is reached.
(i) Select m variables at random from the p variables.
(ii) Pick the best variable/split point among the m.
(iii) Split the node into two daughter nodes.
(2) Output the ensemble of trees $\{T_b\}_1^B$.
To make a prediction at a new point x, let $\widehat{C}_b(x)$ be the class prediction of the $b-$th random forest tree. Then, $\widehat{C}_{rf}^B(x) = \text{majority vote}\left\{\widehat{C}_b(x)\right\}_1^B$.

Algorithm 1: Simplified algorithm of random forest for classification.

models. The RF classifier is trained using bootstrap aggregation, where each new tree is fitted from a bootstrap sample of the training observations $z_i = (y_i, \hat{y}_i)$. The OOB error is the average error for each z_i calculated using predictions from the trees that do not contain z_i in their respective bootstrap sample. This allows the RF classifier to be fitted and validated while being trained [88].

3.5. Bidders Recommender Algorithm

3.5.1. Creation of the Bidders Recommender Algorithm. The flowchart for the creation of the bidders recommender is summarised in Figure 2. The two data sources and the steps for its development are illustrated. It is important to note that the application of the bidders recommender is one thing (see Figure 1), but its creation and setting is another. The steps are quite similar, but they are not the same.

The creation of the bidders recommender has the following four sequential steps. It is based on initially training the classification model, then forecasting the winning company, and aggregating its business information. Finally, it requires searching for similar companies, according to the profile where the search criteria are filters or fixed rules.

(1) Training and Forecasting Phase. Train the classification model (random forest classifier) over the tender dataset. Typically, 80% of the data is for the training subset and 20% is for the testing subset. Then, forecast the winning company for each tender of the testing subset by applying the previous classification model. The following input and output variables (described in Table 1) have been used by the random forest classifier:

(1) Input variables: Procedure_code, Subtype_code, Name_Organisation, Date, CCAA, Province, Municipality, Latitude, Longitude, Tender_Price, CPV, and Duration.
(2) Output variables (forecast): N winning companies (variable called CIF_Winner) for each tender. Typically, $N = 1$ but it is also possible to predict the N most probable companies to win the tender.

At this point, the accuracy$_{n=N}$ of the testing subset can be calculated. It will be the minimum accuracy of the bidders recommender because these N forecast winning companies will be inserted into the recommended companies group.

(2) Aggregation Phase. Add the business fields from the company dataset (described in Table 2) to the forecast winning company estimated in the previous step. The business fields are

(1) General information: *CIF*, *Last_Available_Year_Info*, *Status_Company*, and *Employees*.
(2) Location: *Latitude* and *Longitude*.
(3) Economic indicators per year: *Operating_Income*, *EBIT*, and *EBITDA*.
(4) Systems of classification of economic activities: NACE2, IAE, SIC, and NAICS.

(3) Searching Phase. In the company dataset, search for similar companies to the forecast winning company. Hence, it will create a recommended companies group for each tender. The search criteria (filters) are a basic mechanism to modulate the number of recommended companies, and they are described below. Each filter has a constant factor (numeric value from 0 to infinite) to increase or decrease the size of the search.

(a) $\text{OperatingIncome}_{\text{co.}} \geq F_{\text{OI}} \cdot \text{OperatingIncome}_{\text{forecastco.}}$.
(b) $\text{EBIT}_{\text{co.}} \geq F_{\text{EBIT}} \cdot \text{EBIT}_{\text{forecast co.}}$.
(c) $\text{EBITDA}_{\text{co.}} \geq F_{\text{EBITDA}} \cdot \text{EBITDA}_{\text{forecast co.}}$.
(d) $\text{Employees}_{\text{co.}} \geq F_E \cdot \text{Employees}_{\text{forecast co.}}$.
(e) $\sum_{i=1}^{C} 1[\{\text{Code}\}_{\text{co.}} = \{\text{Code}\}_{\text{forecast co.}}] \geq F_{CEA} \cdot C$ where $1[\text{Code}]$ is the indicator function (returns 1 if the codes match and 0 otherwise), C is the total number of codes of the forecast company, and {Code} is the identification number of the different systems of classifications of economic activities registered by the forecast company:

$$\text{Code} = \{\text{NACE2, IAE, SIC and NAICS}\}.$$

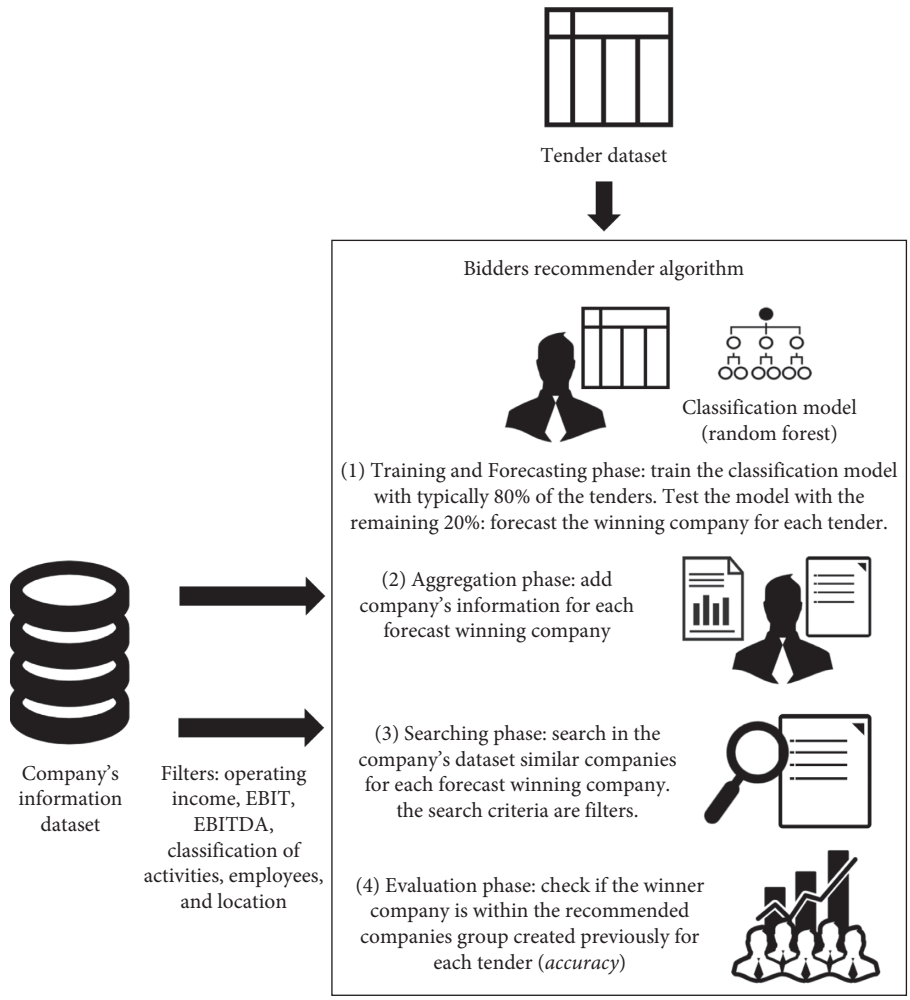


Figure 2: Flowchart of the creation of the bidders recommender.

(f) $\text{Distance}_{\text{tender}-\text{co.}} \leq F_{\text{D}} \cdot \text{Distance}_{\text{tender}-\text{forecast co.}}$.

Therefore, it is necessary to set up the bidders recommender by assigning numeric values to the previous six factors: F_{OI}, F_{EBIT}, F_{EBITDA}, F_E, F_{CEA}, and F_D. The three economic filters (operating income, EBIT, and EBITDA) are annual values. The minimum annual value for $\text{Operating\_Income}_{\text{forecast co.}}$, $\text{EBIT}_{\text{forecast co.}}$, and $\text{EBITDA}_{\text{forecast co.}}$ for the last available 5 years were selected. For searching companies, the $\text{Operating\_Income}_{\text{co.}}$, $\text{EBIT}_{\text{co.}}$, and $\text{EBITDA}_{\text{co.}}$ of the tender's year date were selected.

(4) Evaluation Phase. Check if the real winner company is within the recommended companies group created for each tender (phase 3). This evaluation metric is called $\text{accuracy}_{n=M}$. Logically, $\text{accuracy}_{n=M} \geq \text{accuracy}_{n=N}$ because the N forecast winning companies (phase 1) are automatically within the recommended companies group. Furthermore, the mean and median number of the recommended companies of each tender is calculated. Large groups are more likely to contain the real winner company but, obviously, the smart search engine is less useful because it recommends too many companies.

Therefore, the bidders recommender selects winning companies from the tender dataset but also incorporates new companies available in the market (company dataset) that have a similar profile to the forecast winning company. Creating this profile to search similar companies is a very complex issue, which has been simplified. For this reason, the searching phase (3) has basic filters or rules. Moreover, it is possible to modify or add other filters according to the available company dataset used in the aggregation phase. The fields available in the company dataset (filters) will strongly depend on the country. In our case study, the filters are the following:

(i) Economic resources to finance the project: $\text{Operating\_Income}_{\text{co.}}$, $\text{EBIT}_{\text{co.}}$, and $\text{EBITDA}_{\text{co.}}$.

(ii) Human resources to do the work: $\text{Employees}_{\text{co.}}$.

(iii) Kind of specialised work which the company can do: NACE2, IAE, SIC, and NAICS.

(iv) Geographical distance between the company's location and the tender's location: $\text{Distance}_{\text{tender}-\text{co.}}$. It will be shown that it is a fundamental parameter. Intuitively, the proximity has business benefits such as lower costs.

3.5.2. Application of the Bidders Recommender. The application of the bidders recommender (see Figure 1) by public agencies or potential bidders for a new tender was summarised in Section 1. It has three phases, which is very similar to its creation. The first phase (forecasting) is to predict the most probable company to win the tender using the model, already trained by the random forest classifier. The second phase (aggregation) is exactly the same: add the business fields from the company to the forecast winning company. Finally, the third phase (searching) is simply applying the filters (numeric factors) that were previously fixed in the creation, in order to search the recommended companies.

4. Experimental Analysis

A real case study from Spain is presented to evaluate the bidders recommender. Section 4.1 summarises the preprocessing of the two data sources: tender dataset and company dataset. Section 4.2 provides a quantitative description of both datasets and their relationship such as the correlation. In Section 4.3, the bidders recommender is applied under two different scenarios with five different settings in each one. Finally, the results are presented and analysed for these ten different tests.

4.1. Data Preprocessing. Data preprocessing of the tender dataset is necessary due to the fact that information has not been verified automatically to correct human errors, such as incorrect formatting, wrong values, empty fields, and so on. Data preprocessing can be divided into the following 5 consecutive tasks: extraction, reduction, cleaning, transformation, and filtering. They are described in detail in [10] because the data source and the data preprocessing are the same in both articles. At first, there were 612,090 tenders. After data preprocessing, there were 110,987 tenders.

Data preprocessing of the company dataset is a simple task since the data source is already a database. Therefore, it is not necessary to verify or check the data. The company dataset has 1,353,213 Spanish companies listed.

Finally, the tender dataset has been merged with the company dataset. This relationship is possible thanks to the CIF field (ID company number) which both datasets have. The merged dataset has 102,087 tenders and their respective winner companies. About 8,900 tenders have been lost because the winning company's CIF has not been found for some reason. The possible reasons include foreign company, wrong CIF value, winning company's CIF not stored in the database, etc.

4.2. Statistical Analysis of the Datasets. Firstly, the most relevant information of the tender dataset will be explained, quantitatively. Secondly, the company dataset will also be explained, and, finally, the correlations between both datasets will be analysed.

Table 3 shows the quantitative description of the tender dataset: total numbers, means, medians, maximum, percentages, etc. The dataset has 19 fields or variables: 15 announcement fields and 4 award fields. There are 102,087 tenders from 2014 to 2020 spread across Spain, and any CPV code is possible. Therefore, there are a wide number of heterogeneous tenders which will be used in the bidders recommender.

Looking at Table 3, the following issues are observed:

(i) There are a lot of winning companies and tendering organisations. On average, each public procurement agency creates 17.72 tenders and each company wins 4.80 tenders.

(ii) There is a great dispersion of prices (for both Tender_Price and Award_Price) considering the median, the mean, and the maximum. Furthermore, there is a remarkable difference between Tender_Price and Award_Price, looking at the differences between their medians (€12,535.60) and their means (€93,177.42).

(iii) The 5 types of CPV with greater weight add up to 51.16% of the total number of tenders.

(iv) With every passing year, a greater number of tenders are recorded in the Spanish Public Procurement System without wrong values or incomplete data.

(v) The Spanish capital (Madrid) accounts for 37.50% of the tenders. The 5 Provinces with greater weight add up to 56.21% of the total number of tenders (Spain has 50 provinces).

(vi) 32.43% of Spanish auctions have only one bidder. A large number of tenders with only one bidder could be a sign of anomaly (collusion, corruption, economical disorder, or others). However, according to the European public reports [90], this ratio is similar to other countries, like, for example, Poland (37.5%), Romania (34%), or Czech Republic (26.6%).

Table 4 shows the quantitative description of the company dataset. There are 1,353,213 companies, and 61.44% of them are active. The dataset has 23 fields (see the description in Table 2): general information of the company, location, employees, 3 economic indicators (operating income, EBIT, and EBITDA), and 5 systems of classification of economic activities (CNAE, NACE2, IAE, SIC, and NAICS).

Looking at Table 4, the following issues are discussed:

(1) The Spanish companies have a small size for 3 reasons. First of all, 91.58% are limited companies (private companies limited by shares). Secondly, the mean number of employees is 11.51 employees per company. Thirdly, in the year 2018, the median operating income was only €299,130, the median EBIT was only €10,472.40, and the median EBITDA was only €18,733.35.

(2) The highest number of economic fields (operating income, EBIT, and EBITDA) were recorded in the year 2016 (about 700,000 companies), followed by 2015 and then 2017.

(3) The 5 Provinces with greater weight add up to 45.38% of the total number of companies. So, the companies are concentrated in certain locations.

Table 3: Quantitative description of the tender dataset.

Topic	Description	Value
General values	Total number of tenders in the dataset	102,087
	Temporal range of tenders	2014/01/02–2020/03/31
	Total number of tendering organisations	5,761
	Total number of winning companies	21,268
	Mean number of offers received per tender	4.38
	Mean duration of tender's works	376.30 days
Dataset's variables	Input variables of tender's notice: Procedure_code, Urgency_code, Type_code, Subtype_code, Result_code, Name_Organisation, Postalzone, Postalzone_CCAA, Postalzone_Province, Postalzone_Municipality, Tender_Price, CPV, CPV_Aggregated, Duration, and Date	15 input variables (description in Table 1)
	Output variables of tender's resolution: Award_Price, Winner_Province, CIF_Winner, and Received_Offers	4 output variables (description in Table 1)
Tender price (taxes included)	Mean tender price	€422,293.27
	Median tender price	€78,650.00
	Maximum tender price	€3,196,970,000
	Aggregated tender price of all tenders	€43,110,653,361
Award price (taxes included)	Mean award price	€329,115.85
	Median award price	€66,114.40
	Maximum award price	€786,472,000
	Aggregated award price of all tenders	€33,598,449,589
Number of tenders by received offers (bidders)	Tenders with Received_Offers = 1 (one bidder)	33,112 (32.43%)
	Tenders with Received_Offers = 2 (two bidders)	16,302 (15.97%)
	Tenders with Received_Offers = 3 (three bidders)	13,583 (13.31%)
	Tenders with Received_Offers ≥ 4 (four or more bidders)	39,090 (38.29%)
Number of tenders by CPV	Tenders with CPV = 45 : Construction work	24,699 (24.19%)
	Tenders with CPV = 50 : Repair and maintenance services	8,692 (8.51%)
	Tenders with CPV = 79 : Business services (law, marketing, consulting, recruitment, printing and security)	6,900 (6.76%)
	Tenders with CPV = 72 : IT services (consulting, software development, internet and support)	6,444 (6.31%)
	Tenders with CPV = 34 : Transport equipment and auxiliary products to transportation	5,506 (5.39%)
Number of tenders by type code	Tenders with Type_code = 1: Goods/Supplies	31,065 (30.43%)
	Tenders with Type_code = 2: Services	46,377 (45.43%)
	Tenders with Type_code = 3: Works	24,480 (23.98%)
Number of tenders by year	Number of tenders in 2014	1,002 (0.98%)
	Number of tenders in 2015	5,165 (5.06%)
	Number of tenders in 2016	9,746 (9.55%)
	Number of tenders in 2017	15,081 (14.77%)
	Number of tenders in 2018	25,879 (25.35%)
	Number of tenders in 2019	38,571 (37.78%)
	Number of tenders in 2020 (until March inclusive)	6,643 (6.51%)
Number of tenders by location (province)	Top 1: number of tenders from Madrid	38,285 (37.50%)
	Top 2: number of tenders from Valencia	7,616 (7.46%)
	Top 3: number of tenders from Alicante	4,097 (4.01%)
	Top 4: number of tenders from Baleares	3,866 (3.79%)
	Top 5: number of tenders from Sevilla	3,526 (3.45%)

Figure 3 shows the frequency histogram of the number of tenders won by the same company. The reader must not confuse this histogram with the number of tenders by received offers (bidders) which is described in Table 3. The most frequent number of tenders won by the same company is 1. This means that about 10,000 companies have won only 1 tender. It is more or less 47% of the total number of winning companies. About 3,800 companies (18%) have won 2 tenders and so on (the trend is decreasing). Therefore, only 53% of companies have won 2 or more tenders. This distribution is important for the bidders recommender. It is more difficult to forecast the winning company successfully if a lot of companies have won only 1 tender because there are no patterns, trends, or relationships between tenders.

Figure 4 shows the relationship between the received offers of bidders for each tender and the underbid (also called discount). Actually, the underbid is the evaluation metric called MdAPE (median absolute percentage error) between

Table 4: Quantitative description of the company dataset.

Topic	Description	Value
General values	Total number of companies in the dataset	1,353,213
	Total number of opened companies (actives)	831,356 (61.44%)
	Total number of closed companies (inactives)	521,857 (38.56%)
	Temporal range of the opened companies' establishment date	1842/03/17–2019/03/25
	Mean of the opened companies' establishment date (seniority date)	2002/12/18
	Mean employees of opened companies (actives)	11.51
	Total number of the opened companies of legal entity type: private company limited by shares (Ltd.) (SL in Spanish)	761,358 (91.58%)
	Total number of the opened companies of legal entity type: public limited company (PLC) (SA in Spanish)	60,633 (7.29%)
Dataset's variables	CIF, Establishment_Date, Legal_Form, Last_Available_Year_Info, Status_Company, City_Company, Province_Company, Latitude_Company, Longitude_Company, Employees, Operating_Income, EBIT, EBITDA, CNAE_Primary, CNAE_Secondary, NACE2_Primary, NACE2_Secondary, IAE_Primary, IAE_Secondary, SIC_Primary, SIC_Secondary, NAICS_Primary, and NAICS_Secondary	23 variables (description in Table 2)
Operating income, EBIT, and EBITDA	Total number of opened companies with annual operating income available information (data from 2006 to 2018)	14,695 (2006); 22,080 (2007); 31,067; 38,120; 46,762; 85,210; 460,751; 589,239; 621,926; 659,266; 694,059; 648,598; 124,514 (2018)
	Total number of opened companies with annual EBIT available information (data from 2006 to 2018)	16,642 (2006); 24,618 (2007); 35,441; 41,558; 50,253; 89,890; 476,655; 608,397; 640,520; 677,366; 711,972; 663,761; 127,267 (2018);
	Total number of opened companies with annual EBITDA available information (data from 2006 to 2018)	16,654 (2006); 24,637 (2007); 35,452; 41,571; 50,266; 89,917; 476,719; 608,482; 640,623; 677,468; 712,085; 663,880; 127,295 (2018)
	Mean operating income of the year 2018	€4,122,727.11
	Median operating income of the year 2018	€299,130.00
	Mean EBIT of the year 2018	€397,964.64
	Median EBIT of the year 2018	€10,472.40
	Mean EBITDA of the year 2018	€542,772.79
	Median EBITDA of the year 2018	€18,733.35
Number of opened companies by location (province)	Top 1: number of opened companies from Madrid	157,705 (18.97%)
	Top 2: number of opened companies from Barcelona	114,207 (13.74%)
	Top 3: number of opened companies from Valencia	45,590 (5.48%)
	Top 4: number of opened companies from Alicante	33,386 (4.02%)
	Top 5: number of opened companies from Sevilla	26,368 (3.17%)

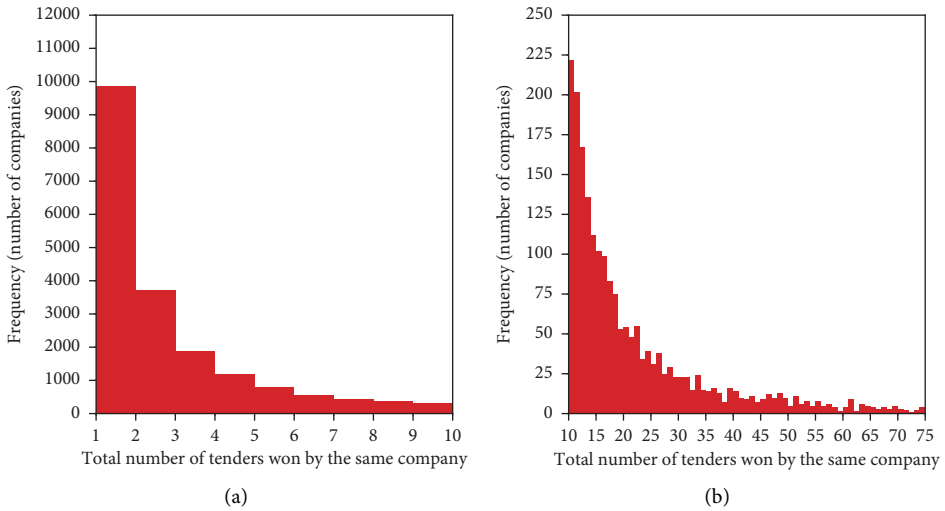


Figure 3: Histogram of frequency (number of companies) based on the total number of tenders in the dataset won by the same company (bidder). The graph is divided into two for better visualisation.

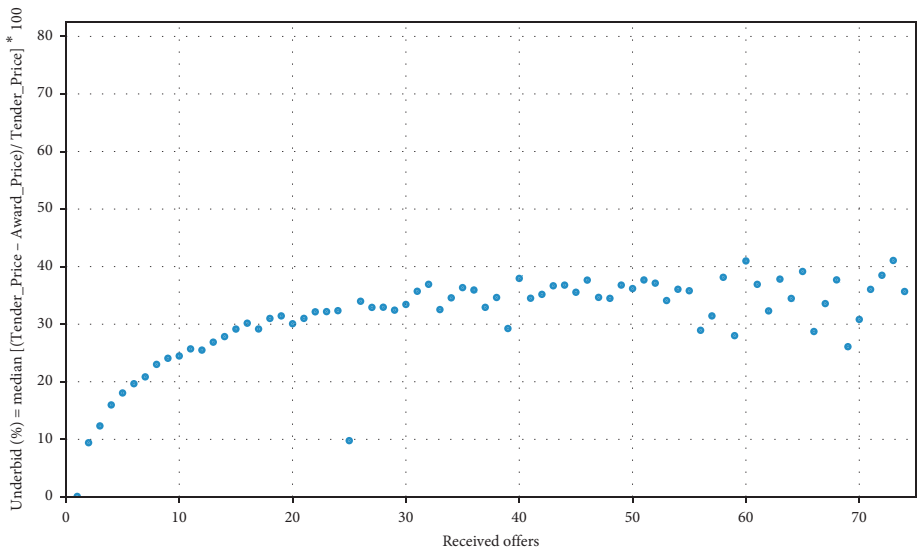


Figure 4: Relation between the received offers of bidders and the underbid (median absolute percentage error between tender price and award price).

the tender price and the award price, which is explained in Section 3.4. The trend is clear: the underbid increases until stabilising at around 35%. Hence, we have quantitatively demonstrated how the tenders with more bidders have lower award prices. In other words, the award price is lower in a tender with more competitiveness and the public procurement agencies will save money. So, the objective of the agencies should be to encourage the participation of companies to receive more offers. For this reason, the bidders recommender is a very useful tool for these agencies because they can effectively increase the number of participants in each tender.

To obtain new, relevant information through the variables in the merged dataset (the tender variables plus company variables), the Spearman correlation method was used. Figure 5 shows the Spearman correlation matrix (a symmetric matrix with respect to the diagonal). It is mathematically described in [10], and it is also used for the same purpose.

Looking at Figure 5, the most important correlations are the following:

(1) Tender_Price vs. Award_Price (0.97): this high correlation is in accordance with common sense since high bids are associated with high awards and low bids with low awards.

(2) Type_code vs. Subtype_code (0.77): each type of contract has its associated subtypes of contract.

(3) City_Tender vs. Province_Tender (0.43): the public procurement agency is in a city which belongs to a Province. So, the relationship city-province is always the same.

(4) Underbid vs. Received_Offers (0.54): the underbid (or discount) is the absolute percentage error (APE %) between Tender_Price and Award_Price. When the public procurement agency receives more offers from bidding companies, the underbid is bigger. This important correlation will be explained in detail in the following section.

(5) CPV vs. Duration (0.33): each type of work is usually associated with a temporal range (duration) for its realisation.

(6) CPV vs. CPV_Aggregated (0.99) has an obvious correlation: CPV_Aggregated is the first 2 digits of the CPV number.

(7) Latitude_Tender vs. Latitude_Company (0.57) and Longitude_Tender vs. Longitude_Company (0.55): this means that both locations (tender and company) are close and therefore the distance tender-company will be an input parameter for the bidders recommender.

(8) Employees, Operating_Income_LAY_-0, EBIT_LAY_-0, and EBITDA_LAY_-0 are strongly correlated with each other. Big companies have a lot of employees, and these companies can earn more profits.

4.3. Bidders Recommender Validation. There are two related validations: firstly, to validate the classification model (random forest) applied in phase 1 (train and forecast) of the bidders recommender and secondly, and more importantly, the validation of the bidders recommender results which is phase 4 (evaluation). This checks if the real winner company is within the recommended companies group.

For validating the classification model, Figure 6 shows three different ratios between the training and testing subsets (train : test in percentage) randomly chosen: 90 : 10, 80 : 20, and 70 : 30. Furthermore, it shows the behaviour of the error metrics (accuracy, precision, balanced accuracy, and OOB) for a different number of trees generated in the random

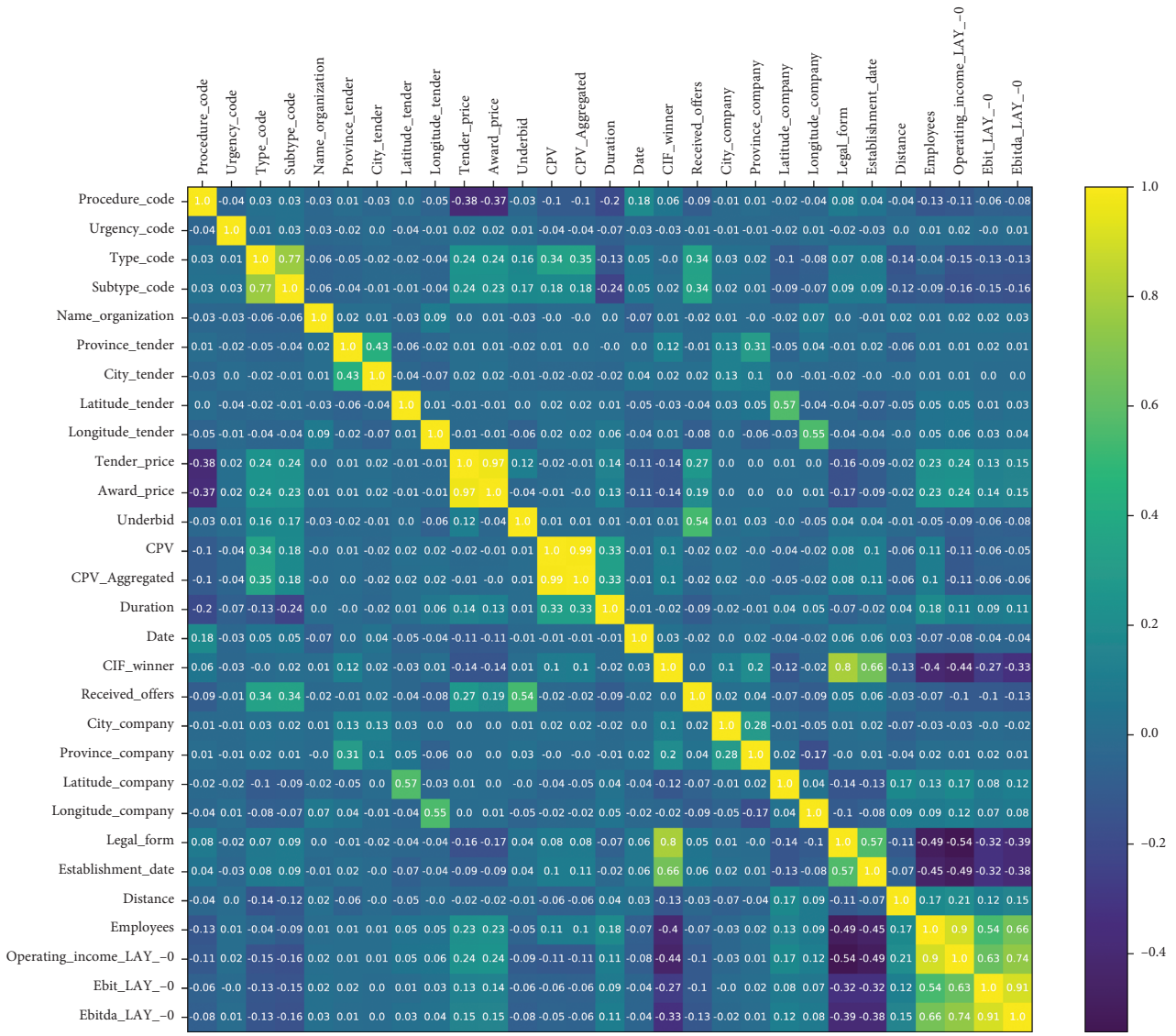


FIGURE 5: Correlation matrix between the variables of the two datasets (tenders and companies). Spearman's rank correlation coefficient is the method applied.

forest classifier. The accuracy$_{n=1}$ is the most important error for this study, and, in each graph, it is constantly of the order of 18%, 17%, and 15%, respectively. Logically, when decreasing the training data percentage, the accuracy is lower. Hence, the number of trees is not relevant and the election of the ratio also has a minimal impact. *RandomForestClassifier* from *Scikit-learn*, which is a machine learning library for the Python programming language, has 75 trees and a ratio of 80 : 20 and is the function used in this article.

Validation of the bidders recommender results was tested over two scenarios with five different setups. In the first scenario, the testing subset is 20% and is chosen randomly. In the second scenario, the dataset is ordered by tender date and the testing subset is the latest 20%, i.e., the most recent tenders. So, the second scenario is more appropriate to test a real engine search. Each scenario has the same five setups (filter settings), from very low (restrictive) filters to very high. The filters are described in detail in Section 3.5. Basically, there are six factors (F_{OI}, F_{EBIT}, F_{EBITDA}, F_E, F_{CEA}, and F_D), and it is necessary to assign numeric values. Hence, there are 10 combinations to test the bidders recommender.

The validation of the bidders recommender is shown in Table 5. The evaluation metric to measure the success of the recommender is the accuracy: the percentage of tenders where the winning company is within the recommended companies group. For scenario 1, when $N = 1$ (it is predicted that the most probable company will win the tender), the accuracy is 17.07%. When $N = 5$ (the 5 most probable companies to win the tender), the accuracy rises to 31.58%. Finally, the bidders recommender searches a group of compatible companies, automatically including the previous 5 companies, for each tender. The range of the accuracy is from 33.25% to 38.52% according to the settings applied. The

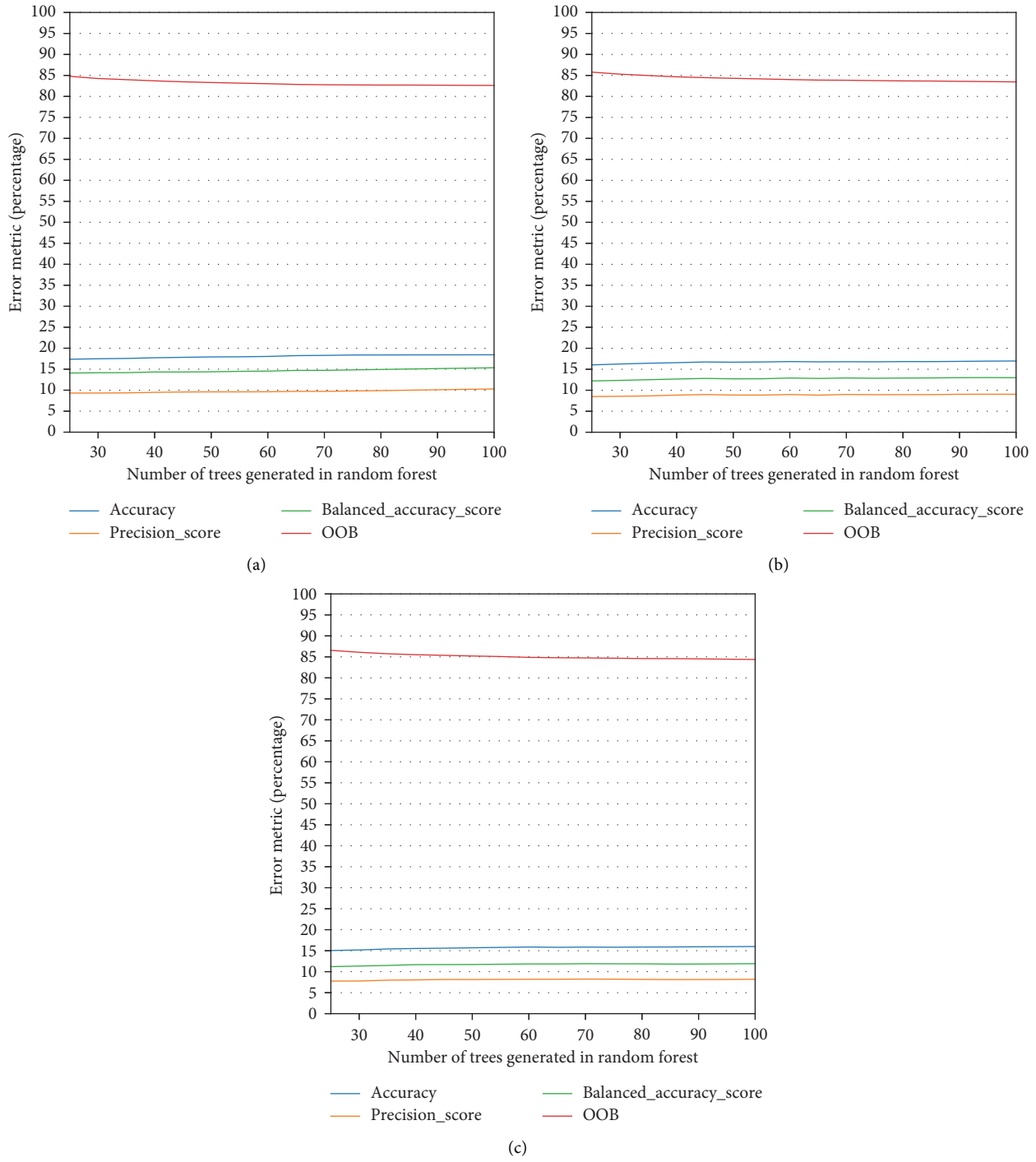


Figure 6: Relationship between trees in random forests and error metrics (accuracy, precision, balanced accuracy, and OOB) for different ratios of training and testing subsets. (a) 90 : 10. (b) 80 : 20. (c) 70 : 30.

reason to the increasing accuracy is simple: there are more recommended companies. Consequently, the mean (and median) number of recommended companies is higher.

Analogously for scenario 2, $\text{Accuracy}_{n=1} = 10.25\%$, $\text{Accuracy}_{n=5} = 23.12\%$, and $\text{Accuracy}_{n=M} = [24.79\% - 30.52\%]$. This accuracy is significantly lower than that in scenario 1, and it could be for multiple reasons. For example, recent tenders have less business information because the annual accounts of the winner company are published the following year. In particular, the company dataset does not have information about operating income, EBIT, and EBITDA in 2019 and 2020 (see Table 4). However, there are a lot of tenders in 2019 and 2020 (see Table 3).

One area of interesting analysis is the size of the companies group generated by the bidders recommender. This recommender will be more efficient if the group is small and

Table 5: Testing the bidders recommender for two scenarios: results of the accuracy and number of recommended companies per tender for five different setups.

Description		Different bidders recommender settings				
		Very low	Low	Medium	High	Very high
Bidders recommender factors for the settings	F_{OI}: operating income factor	0.25	0.5	0.65	0.75	1.0
	F_{EBIT}: EBIT factor	0.25	0.5	0.65	0.75	1.0
	F_{EBITDA}: EBITDA factor	0.25	0.5	0.65	0.75	1.0
	F_E: employees factor	0.15	0.25	0.25	0.35	0.45
	F_{CEA}: classification economic activities factor	0.125	0.15	0.14	0.175	0.2
	F_D: distance tender-company factor	1.6	1.4	1.4	1.2	1
Results of scenario 1: testing subset is the 20% of the dataset randomly chosen	$\text{Accuracy}_{n=1}$: winner company is the forecast company			17.07%		
	$\text{Accuracy}_{n=5}$: winner company is within the top 5 forecast companies			31.58%		
	$\text{Accuracy}_{n=M}$: winner company is within the recommended companies group	38.52%	36.20%	35.92%	34.04%	33.25%
	Mean and median number of the recommended companies of each tender	877.43; 86	469.69; 35	430.48; 31	226.07; 11	145.97; 9
Results of scenario 2: testing subset is the last 20% of the dataset ordered by tender's date	$\text{Accuracy}_{n=1}$: winner company is the forecast company			10.25%		
	$\text{Accuracy}_{n=5}$: winner company is within the top 5 forecast companies			23.12%		
	$\text{Accuracy}_{n=M}$: winner company is within the recommended companies group	30.52%	28.00%	27.73%	25.55%	24.79%
	Mean and median number of the recommended companies of each tender	900.64; 95	470.41; 37	430.33; 33	210.92; 11	132.10; 9

the accuracy is high. Figure 7 shows the boxplots, disaggregated by CPV, for scenarios 1 and 2 (medium setup). CPV is the system for classifying the type of work in public contracts. The total mean is very similar in both scenarios: 430.48 potential bidders (median is 31) and 430.33 potential bidders (median is 33), respectively. The median value, disaggregated by CPV, is usually below 50 companies. However, the mean value of each CPV has great variability.

5. Discussion

The main objective is to find out and recommend companies for a new tender announcement. However, it is not easy to measure the performance of the bidders recommender; each company is unique and different from the rest, so the searching, comparison, and recommendation of companies is relative (subjective evaluation). Accuracy has been selected as the evaluation metric to measure the performance: the percentage of tenders where the winning company is within the recommended companies group.

Table 5 shows the results of the bidders recommender: the accuracy, mean, and median number of recommended companies over two scenarios with five different set ups (very low, low, medium, high, and very high). The main determining factor to get a good performance is due to the top 5 forecast companies (called $\text{Accuracy}_{n=5}$). This means that the 5 most probable companies to win a tender can be incorporated to the recommender companies group (called $\text{Accuracy}_{n=M}$). For scenario 1, $\text{Accuracy}_{n=5} = 31.58\%$ and $\text{Accuracy}_{n=M} = [33.25\% - 38.52\%]$. For scenario 2, $\text{Accuracy}_{n=5} = 23.12\%$ and $\text{Accuracy}_{n=M} = [24.79\% - 30.52\%]$. The range is governed by the bidders recommender settings. Hence, the user can configure the factors for the settings (F_{OI}, F_{EBIT}, F_{EBITDA}, F_E, F_{CEA}, and F_D) to search more or less companies.

Figure 7 shows the boxplots for the size of the recommended companies group, disaggregated by the type of tender's work (CPV). There are considerable differences in the size, mean, and median values for each CPV. Other interesting analyses would be to disaggregate by geographic regions, business sectors, or markets.

As seen in this article, the bidders recommender depends strongly on the fields of public procurement announcements and the information available to characterise the bidders. Therefore, the recommender cannot be the same for each country since their public procurement systems are not unified or standardised for several reasons: regulations, laws, diverse information systems, different tender criteria, distinct levels of technological maturity in public administration, etc. However, this paper establishes the basis to create a bidders recommender which can be adapted to each country according to the two basic data sources: tender information and company information. This is because the recommender is an open frame which can easily add or modify other

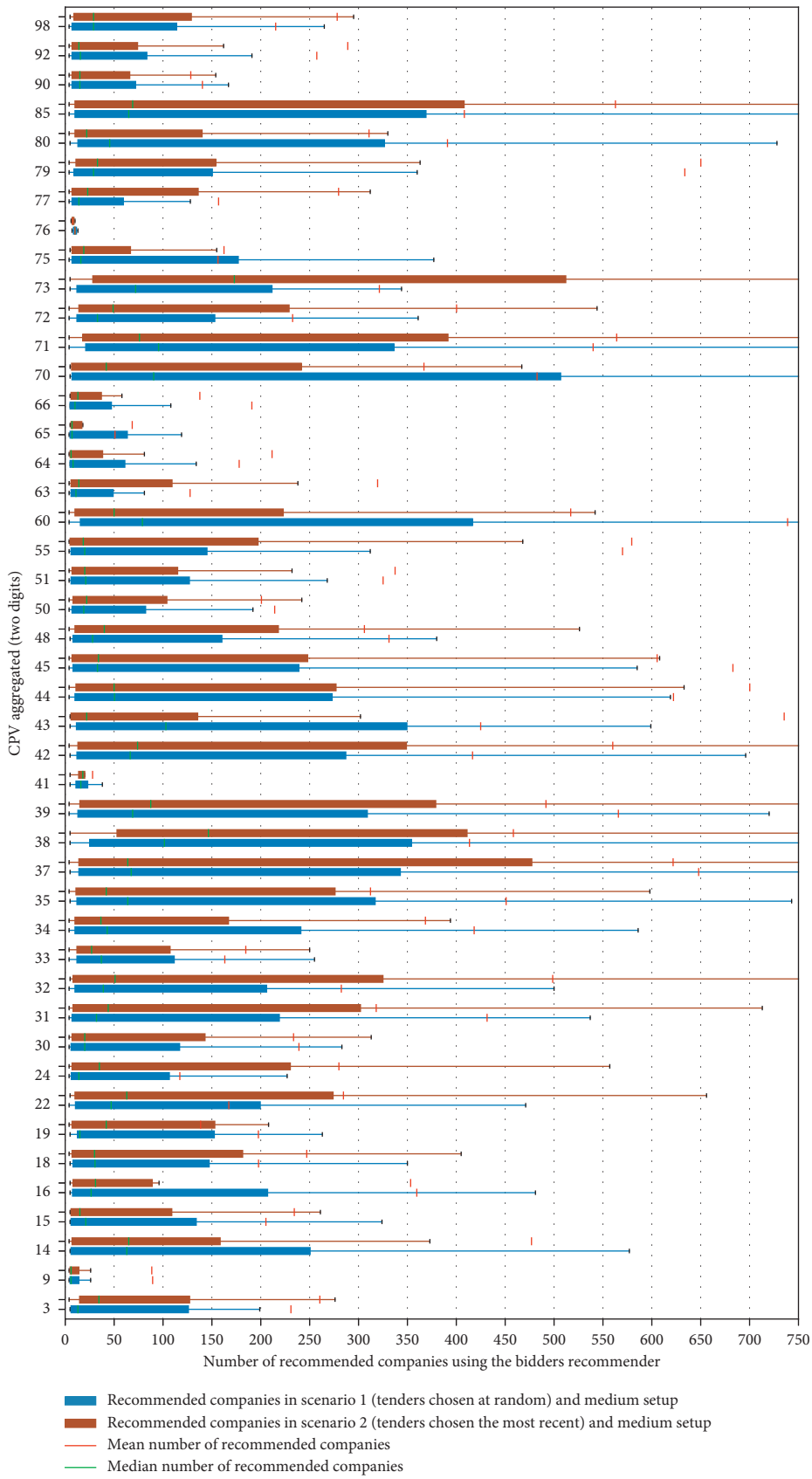


Figure 7: Boxplots for the size of the recommended companies group generated by the bidders recommender, disaggregated by CPV. Scenario 1 (blue colour) and scenario 2 (brown colour) both have a medium setup.

available fields or data sources. The selection and optimisation of the recommender's parameters can significantly improve it. It is a laborious task and particular to each country.

In summary, the recommender is an effective tool for society because it enables and increases the bidders participation in tenders with less effort and resources. Furthermore, this will serve to modernise the public procurement systems with a new approach based on machine learning methods and data analysis. Thus, the beneficiaries are the government, the citizens, and the two main users:

(1) *Public Contracting Agencies*. When they publish a tender notice, the algorithm automatically recommends suppliers which have a suitable profile for the tender. The agencies could contact these suppliers directly and invite them to participate if they are really interested in the tender.

(2) *Potential Bidders*. They will be able to search suitable tenders effortlessly, according to the type of tender and the profile of previous winning companies.

6. Conclusions and Future Research

The public procurement systems of many countries continue to use the inefficient mechanisms and tools of the 20th century for the publication of tenders and the attraction the offers and bidders. However, more and more new technologies (open data, big data, machine learning, etc.) are emerging in the public administration sector to improve their systems, proceedings, and services. This article clearly demonstrates how it is possible to create new tools using these technologies.

Especially, this paper develops a pioneering algorithm to recommend potential bidders. It is a multidisciplinary system which fills a gap in the literature. The bidders recommender proposed here is a promising and strategic instrument for improving the efficiency of public procurement agencies and should also facilitate access to the tenders for the suppliers. The recommender brings a trendy new perspective to gathering tenders and bidders.

The bidders recommender is described theoretically and also validated experimentally, using a case study from Spain. Two datasets have been used: tender dataset (102,087 Spanish tenders from 2014 to 2020) and company dataset (1,353,213 Spanish companies). The company dataset is difficult to collect because it is nonfree public information in Spain, so it is a valuable dataset. Quantitative, graphical, and statistical descriptions of both datasets have been presented.

The results of the case study have been successful because of the accuracy; it means that the winning bidding company is within the recommended companies group (from 24% to 38% of the tenders). The accuracy range is due to the two test scenarios (either being chosen from the most recent tenders or chosen at random), and each scenario has five different settings for the bidders recommender. Hence, the recommender has been validated for over 10 combinations of testing and the results are quite successful and promising, opening the research up to other countries and datasets.

The main limitation of this research is inherent to the design of the recommender's algorithm because it necessarily assumes that winning companies will behave as they behaved in the past. Companies and the market are living entities which are continuously changing. On the other hand, only the identity of the winning company is known in the Spanish tender dataset, not the rest of the bidders. Moreover, the fields of the company's dataset are very limited. Therefore, there is little knowledge about the profile of other companies which applied for the tender. Maybe in other countries the rest of the bidders are known. It would be easy to adapt the bidder recommender to this more favourable situation.

This paper opens the door to future research for creating bidder recommendation systems. In particular, for this recommender, some research can be done to improve it, as follows:

(i) The training and forecasting phase of the algorithm (step 1) to predict the winning company is based on the random forest classifier. Alternative methods of machine learning can be studied to increase the accuracy.

(ii) The aggregation phase (step 2) can use other fields of business information to create the profile of the winning company for the tender.

(iii) The searching phase (step 3) implements basic rules or filters to search similar companies. It would be interesting to explore more sophisticated methods, for example: clustering to group similar companies.

(iv) There is no ranking of recommended companies. This means that the algorithm only recommends companies without any associated probabilities, so the user cannot choose the companies that are most likely to be recommended to win the tender. This can be solved by applying a voting system or some kind of distance in the searching phase (step 3) of the algorithm.

Data Availability

The processed data used to support the findings of this study are available from the corresponding author upon request. The raw data from Spain are available at the Ministry of Finance, Spain (open data of Spanish tenders are hosted in http://www.hacienda.gob.es/es-ES/GobiernoAbierto/Datos%20Abiertos/Paginas/licitaciones_plataforma_contratacion.aspx).

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 Pablo Ballesteros-Pérez (PhD in Project Management and expert researcher in procurement auctions from the University of Cádiz (Spain)), for his very

valuable comments and suggestions to improve this article. This study was supported by the Plan of Science, Technology and Innovation of the Principality of Asturias (Ref: FC-GRUPIN-IDI/2018/000225).

References

[1] European Commission, "Public procurement," 2017.
[2] E. Huyer and L. van Knippenberg, "The economic impact of open data opportunities for value creation in europe," 2020.
[3] S. Curto, S. Ghislandi, K. Van de Vooren, S. Duranti, and L. Garattini, "Regional tenders on biosimilars in Italy: an empirical analysis of awarded prices," *Health Policy*, vol. 116, no. 2-3, pp. 182–187, 2014.
[4] T. Hanák and P. Muchová, "Impact of competition on prices in public sector procurement," *Procedia Computer Science*, vol. 64, pp. 729–735, 2015.
[5] J. Soudek and J. Skuhrovec, "Procurement procedure, competition and final unit price: the case of commodities," *Journal of Public Procedure*, vol. 16, no. 1, pp. 1–21, 2016.
[6] OECD Public Governance Reviews, *SMEs in Public Procurement: Practices and Strategies for Shared Benefits*, OECD Publishing, Paris, 2018.
[7] P. Ballesteros-Pérez, M. Skitmore, E. Pellicer, and J. H. Gutiérrez-Bahamondes, "Improving the estimation of probability of bidder participation in procurement auctions," *International Journal of Project Management*, vol. 34, no. 2, pp. 158–172, 2016.
[8] A. Mehrbod and A. Grilo, "Advanced Engineering Informatics Tender calls search using a procurement product named entity recogniser," *Advanced Engineering Informatics*, vol. 36, 2018.
[9] M. Nečaský, J. Klímek, J. Mynarz, T. Knap, V. Svátek, and J. Stárka, "Linked data support for filing public contracts," *Complexity*, vol. 65, no. 5, pp. 862–877, 2014.
[10] M. J. García Rodríguez, V. Rodríguez Montequín, F. Ortega Fernández, and J. M. Villanueva Balsera, "Public procurement announcements in spain: regulations, data analysis, and award price estimator using machine learning," *Complexity*, vol. 2019, 2019.
[11] M. J. García Rodríguez, V. R. Montequín, F. O. Fernández, and J. V. Balsera, "Spanish Public Procurement: legislation, open data source and extracting valuable information of procurement announcements," *Procedia Computer Science*, vol. 164, pp. 441–448, 2019.
[12] D. Corrales-Garay, M. Ortiz-de-Urbina-Criado, and E. M. Mora-Valentín, "Knowledge areas, themes and future research on open data: a co-word analysis," *Government Information Quarterly*, vol. 36, no. 1, pp. 77–87, 2018.
[13] J. Attard, F. Orlandi, S. Scerri, and S. Auer, "A systematic review of open government data initiatives," *Government Information Quarterly*, vol. 32, no. 4, pp. 399–418, 2015.
[14] E. Afful-Dadzie and A. Afful-Dadzie, "Liberation of public data: exploring central themes in open government data and freedom of information research," *International Journal of Information Management*, vol. 37, no. 6, pp. 664–672, 2017.
[15] J. Lassinantti, A. Ståhlbröst, and M. Runardotter, "Relevant social groups for open data use and engagement," *Government Information Quarterly*, vol. 36, no. 1, pp. 98–111, 2018.
[16] F. Gonzalez-Zapata and R. Heeks, "The multiple meanings of open government data: understanding different stakeholders and their perspectives," *Government Information Quarterly*, vol. 32, no. 4, pp. 441–452, 2015.
[17] J. D. Twizeyimana and A. Andersson, "The public value of E-Government-a literature review," *Government Information Quarterly*, vol. 36, no. 2, pp. 167–178, 2019.
[18] F. Ahmadi Zeleti, A. Ojo, and E. Curry, "Exploring the economic value of open government data," *Government Information Quarterly*, vol. 33, no. 3, pp. 535–551, 2016.
[19] G. Magalhaes and C. Roseira, "Open government data and the private sector: an empirical view on business models and value creation," *Government Information Quarterly*, vol. 23, pp. 1–10, 2017.
[20] R. Krishnamurthy and Y. Awazu, "Liberating data for public value: the case of Data.gov," *International Journal of Information Management*, vol. 36, no. 4, pp. 668–672, 2016.
[21] S. Sadiq and M. Indulska, "Open data: quality over quantity," *International Journal of Information Management*, vol. 37, no. 3, pp. 150–154, 2017.
[22] S. Kubler, J. Robert, S. Neumaier, J. Umbrich, and Y. Le Traon, "Comparison of metadata quality in open data portals using the Analytic Hierarchy Process," *Government Information Quarterly*, vol. 35, no. 1, pp. 13–29, 2018.
[23] R. P. Lourenço, "An analysis of open government portals: a perspective of transparency for accountability," *Government Information Quarterly*, vol. 32, no. 3, pp. 323–332, 2015.
[24] N. Veljković, S. Bogdanović-Dinić, and L. Stoimenov, "Benchmarking open government: an open data perspective," *Government Information Quarterly*, vol. 31, no. 2, pp. 278–290, 2014.
[25] M. Lnenicka and J. Komarkova, "Big and open linked data analytics ecosystem: theoretical background and essential elements," *Government Information Quarterly*, vol. 36, no. 1, pp. 129–144, 2018.
[26] N. Obwegeser and S. D. Müller, "Innovation and public procurement: terminology, concepts, and applications," *Technovation*, vol. 74, 2018.
[27] P. Adjei-bamfo, T. Maloreh-nyamekye, and A. Ahenkan, "The role of e-government in sustainable public procurement in developing countries : a systematic literature review," *Government Information Quarterly*, vol. 142, 2018.
[28] S. Mullainathan and J. Spiess, "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 87–106, 2017.
[29] H. R. Varian, "Big data: new tricks for econometrics," *Journal of Economic Perspectives*, vol. 28, no. 2, pp. 3–28, 2014.
[30] I. Lee and Y. J. Shin, "Machine learning for enterprises: applications, algorithm selection, and challenges," *Business Horizons*, vol. 63, no. 2, pp. 157–170, 2020.
[31] M. Bilal and L. O. Oyedele, "Big Data with deep learning for benchmarking profitability performance in project tendering," *Expert Systems with Applications*, vol. 147, 2020.
[32] J. J. Grandia, "Assessing the implementation of sustainable public procurement using quantitative text-analysis tools: a large-scale analysis of Belgian public procurement notices," *Journal of Purchasing and Supply Management*, vol. 19, 2020.
[33] M. A. Bergman and S. Lundberg, "Tender evaluation and supplier selection methods in public procurement," *Journal of Purchasing and Supply Management*, vol. 19, no. 2, pp. 73–83, 2013.
[34] P. Ballesteros-Pérez, M. C. González-Cruz, and A. Cañavate-Grimal, "On competitive bidding: scoring and position probability graphs," *International Journal of Project Management*, vol. 31, no. 3, pp. 434–448, 2013.
[35] M. Falagario, F. Sciancalepore, N. Costantino, and R. Pietroforte, "Using a DEA-cross efficiency approach in

public procurement tenders," *European Journal of Operational Research*, vol. 218, no. 2, pp. 523–529, 2012.
[36] M. Dotoli, N. Epicoco, and M. Falagario, "Multi-Criteria Decision Making techniques for the management of public procurement tenders: a case study," *European Journal of Operational Research*, vol. 88, 2020.
[37] Y. Wang, C. Xi, S. Zhang, D. Yu, W. Zhang, and Y. Li, "A combination of extended fuzzy AHP and Fuzzy GRA for government e-tendering in hybrid fuzzy environment," *European Journal of Operational Research*, vol. 2014, 2014.
[38] P. L. Lorentziadis, "Competitive bidding in asymmetric multidimensional public procurement," *European Journal of Operational Research*, vol. 282, no. 1, pp. 211–220, 2020.
[39] P. Ballesteros-Pérez and M. Skitmore, "On the distribution of bids for construction contract auctions," *Construction Management and Economics*, vol. 35, no. 3, pp. 106–121, 2017.
[40] P. Ballesteros-Pérez, M. L. del Campo-Hitschfeld, D. Mora-Melià, and D. Domínguez, "Modeling bidding competitiveness and position performance in multi-attribute construction auctions," *Operations Research Perspectives*, vol. 2, pp. 24–35, 2015.
[41] H. Jung, G. Kosmopoulou, C. Lamarche, and R. Sicotte, "Strategic bidding and contract renegotiation," *International Economic Review*, vol. 60, no. 2, pp. 801–820, 2019.
[42] A. Cheaitou, R. Larbi, and B. Al Housani, "Decision making framework for tender evaluation and contractor selection in public organisations with risk considerations," *International Economic Review*, vol. 68, 2019.
[43] J. Bochenek, "The contractor selection criteria in open and restricted procedures in public sector in selected EU countries," *Procedia Engineering*, vol. 85, pp. 69–74, 2014.
[44] T. Hanák and C. Serrat, "Analysis of construction auctions data in Slovak public procurement," *Advances in Civil Engineering*, vol. 2018, 2018.
[45] D. Imhof, *Empirical Methods for Detecting Bid-Rigging Cartels*, Université Bourgogne Franche-Comté, London, UK, 2018.
[46] P. Ballesteros-Pérez, M. C. González-Cruz, A. Cañavate-Grimal, and E. Pellicer, "Detecting abnormal and collusive bids in capped tendering," *Automation in Construction*, vol. 31, pp. 215–229, 2013.
[47] S. S. Padhi, S. M. Wagner, and P. K. J. Mohapatra, "Design of auction parameters to reduce the effect of collusion," *Decision Sciences*, vol. 47, no. 6, pp. 1016–1047, 2016.
[48] B. Tóth, M. Fazekas, and T. István János, "Toolkit for detecting collusive bidding in public procurement with examples from hungary," 2015.
[49] G. L. Albano, B. Cesi, and A. Iozzi, "Public procurement with unverifiable quality: the case for discriminatory competitive procedures," *Journal of Public Economics*, vol. 145, pp. 14–26, 2017.
[50] S. Tadelis, "Public procurement design: lessons from the private sector," *International Journal of Industrial Organization*, vol. 30, no. 3, pp. 297–302, 2012.
[51] K. Bloomfield, T. Williams, C. Bovis, and Y. Merali, "Systemic risk in major public contracts," *International Journal of Forecasting*, vol. 35, no. 2, pp. 667–676, 2019.
[52] G. Locatelli, G. Mariani, T. Sainati, and M. Greco, "Corruption in public projects and megaprojects: there is an elephant in the room!," *International Journal of Project Management*, vol. 35, no. 3, pp. 252–268, 2017.
[53] K. G. Dastidar and D. Mukherjee, "Corruption in delegated public procurement auctions," *European Journal of Political Economy*, vol. 35, pp. 122–127, 2014.
[54] A. Estache and R. Foucart, "The scope and limits of accounting and judicial courts intervention in inefficient public procurement," *European Journal of Political Economy*, vol. 157, 2018.
[55] Y. Huang, "An empirical study of scoring auctions and quality manipulation corruption," *European Economic Review*, vol. 120, 2019.
[56] P. Detkova, E. Podkolzina, and A. Tkachenko, "Corruption, centralization and competition: evidence from Russian public procurement," *International Journal of Public Administration*, vol. 41, no. 5-6, pp. 414–434, 2018.
[57] V. Titl and B. Geys, "Political donations and the allocation of public procurement contracts," *European Economic Review*, vol. 111, pp. 443–458, 2019.
[58] I. J. Tóth and M. Hajdu, "Cronyism in Hungary An empirical analysis of public tenders 2010-2016," 2018.
[59] OCDE, "Algorithms and collusion," 2017.
[60] M. Huber and D. Imhof, "Machine learning with screens for detecting bid-rigging cartels," *International Journal of Industrial Organization*, vol. 65, pp. 277–301, Jul. 2019.
[61] K. Rabuzin and N. Modrušan, *Prediction of Public Procurement Corruption Indices Using Machine Learning Methods*, Knowledge Engineering and Knowledge Management, New York, NY, USA, 2019.
[62] T. Sun and L. J. Sales, "Predicting public procurement irregularity: an application of neural networks," *Journal of Emerging Technologies in Accounting*, vol. 15, no. 1, pp. 141–154, 2018.
[63] P. Ballesteros-Pérez, M. C. González-Cruz, and A. Cañavate-Grimal, "Mathematical relationships between scoring parameters in capped tendering," *International Journal of Project Management*, vol. 30, no. 7, pp. 850–862, 2012.
[64] P. Ballesteros-Pérez, M. C. González-Cruz, M. Fernández-Diego, and E. Pellicer, "Estimating future bidding performance of competitor bidders in capped tenders," *Journal of Civil Engineering and Management*, vol. 20, no. 5, pp. 702–713, 2014.
[65] J.-S. Chou, C.-W. Lin, A.-D. Pham, and J.-Y. Shao, "Optimized artificial intelligence models for predicting project award price," *Automation in Construction*, vol. 54, pp. 106–115, 2015.
[66] J.-M. Kim and H. Jung, "Predicting bid prices by using machine learning methods," *Applied Economics*, vol. 51, no. 19, p. 2011, 2018.
[67] T. D. Fry, R. A. Leitch, P. R. Philipoom, and Y. Tian, "Empirical analysis of cost estimation accuracy in procurement auctions," *International Journal of Business and Management*, vol. 11, no. 3, p. 1, 2016.
[68] R. M. Skitmore and S. T. Ng, "Forecast models for actual construction time and cost," *International Journal of Business and Management*, vol. 38, no. 8, pp. 1075–1083, 2003.
[69] Official Website of the European Union, "European e-justice portal," 2003.
[70] D. Goens, "The exploitation of Business Register data from a public sector information and data protection perspective: a case study," *Computer Law & Security Review*, vol. 26, no. 4, pp. 398–405, 2010.
[71] R. Matin, C. Hansen, C. Hansen, and P. Mølgaard, "Predicting distresses using deep learning of text segments in annual reports," *Expert Systems with Applications*, vol. 132, pp. 199–208, 2019.
[72] S. Jones and T. Wang, "Predicting private company failure: a multi-class analysis," *Journal of International Financial Markets, Institutions and Money*, vol. 61, pp. 161–188, 2019.

[73] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968.
[74] S. Chava and R. A. Jarrow, *Bankruptcy Prediction with Industry Effects*, World Scientific, Berlin, Germany, 2008.
[75] D. Duffie, L. Saita, and K. Wang, "Multi-period corporate default prediction with stochastic covariates," *The Journal of Finance*, vol. 83, no. 3, pp. 635–665, 2007.
[76] E. Altman, G. Sabato, and N. Wilson, "The value of non-financial information in small and medium-sized enterprise risk management," *The Journal of Finance*, vol. 6, no. 2, pp. 1–33, 2010.
[77] Q. Yu, Y. Miche, E. Séverin, and A. Lendasse, "Bankruptcy prediction using Extreme Learning Machine and financial expertise," *Neurocomputing*, vol. 128, pp. 296–302, 2014.
[78] J. Min and Y. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Systems with Applications*, vol. 28, no. 4, pp. 603–614, 2005.
[79] H. Son, C. Hyun, D. Phan, and H. J. Hwang, "Data analytic approach for bankruptcy prediction," *The Journal of Finance*, vol. 28, 2019.
[80] C.-F. Tsai, "Combining cluster analysis with classifier ensembles to predict financial distress," *Information Fusion*, vol. 16, no. 1, pp. 46–58, 2014.
[81] The European Commission, "Regulation (EU) 2015/884 establishing technical specifications and procedures required for the system of interconnection of registers established by Directive 2009/101/EC," 2015.
[82] European Business Registry Association, https://ebra.be.
[83] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[84] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: a survey and results of new tests," *Pattern Recognition*, vol. 44, no. 2, pp. 330–349, 2011.
[85] M. R. Segal, "Machine learning benchmarks and random forest regression," *Pattern Recognition*, vol. 44, 2004.
[86] M. Fernández-Delgado, M. S. Sirsat, E. Cernadas, S. Alawadi, S. Barro, and M. Febrero-Bande, "An extensive experimental survey of regression methods," *Neural Networks*, vol. 111, pp. 11–34, 2019.
[87] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
[88] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Berlin, Germany, 2008.
[89] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
[90] M. Fazekas, "Single bidding and non- competitive tendering procedures in EU co-funded projects," 2019, https://ec.europa.eu/regional_policy/en/information/publications/reports/2019/single-bidding-and-non-competitive-tendering.