Hindawi
Complexity
Volume 2021, Article ID 6633906, 12 pages
https://doi.org/10.1155/2021/6633906

Research Article

Applying a Probabilistic Network Method to Solve Business-Related Few-Shot Classification Problems

Lang Wu [1] and Menggang Li [2]

[1]*School of Applied Science, Beijing Information Science and Technology University, Beijing, China*
[2]*School of Economics and Management, Beijing Jiaotong University, Beijing, China*

Correspondence should be addressed to Lang Wu; wulang_0306@163.com

Received 26 November 2020; Revised 9 December 2020; Accepted 11 December 2020; Published 5 January 2021

Academic Editor: Abd E.I.-Baset Hassanien

Copyright © 2021 Lang Wu and Menggang Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It can be challenging to learn algorithms due to the research of business-related few-shot classification problems. Therefore, in this paper, we evaluate the classification of few-shot learning in the commercial field. To accurately identify the categories of few-shot learning problems, we proposed a probabilistic network (PN) method based on few-shot and one-shot learning problems. The enhancement of the original data was followed by the subsequent development of the PN method based on feature extraction, category comparison, and loss function analysis. The effectiveness of the method was validated using two examples (absenteeism at work and Las Vegas Strip hotels). Experimental results demonstrate the ability of the PN method to effectively identify the categories of commercial few-shot learning problems. Therefore, the proposed method can be applied to business-related few-shot classification problems.

1. Introduction

The artificial intelligence witnesses rapid development and yields successes in many fields due to recent explosive growth of data volume and increasing computer processing power [1–5]. With minimal or few pictures, humans can recognize objects based on previous experience and knowledge. It is attributed to the ability of learning from prior knowledge accumulated through years of experience. For example, users can quickly learn how to use smart phones due to their previous knowledge of using Nokia cellphones. Therefore, skills should be trained early by constructing an intelligent system to master multiple skills and to adapt to various environments. And new tasks should be learned based on previous experience.

Few-shot learning can be employed for this purpose. It refers to the ability of generalizing a model with a limited number of labeled samples, particularly "drawing inferences from one example." As an important component in intelligent systems, few-shot learning experienced an extensive development.

Research studies on few-shot learning began in the 1990s [6–8], coinciding with the application of sparse representation. Early studies focused on the exploratory work and proved to be successful. Several influential studies were published from 2003 to 2015, including the work on Bayesian programming learning [9–13]. Li and Fergus et al. proposed the concept of one-shot learning for the first time and demonstrated its application in classifications based on the Bayesian framework [9, 10]. The major characteristic of the Bayesian learning stage is that it can handle small samples and be well integrated into the prior knowledge because the methods adopted by the model are often based on Bayesian theory. However, one problem exists, that is, the generality of the model is often insufficient. In other words, the model at this stage is often designed for a specific problem. When the problem changes, consequently the model is no longer valid. Lake [12, 13] combined the Bayesian framework with prior knowledge to validate concept learning, proving that machine can use background knowledge through complete probability estimation based on few-shot learning. Such methods are highly dependent on selecting prior knowledge of human. The methods adopted by the Bayesian learning stage are often based on Bayesian theory because they can deal with the few-shot learning problem and to integrate prior knowledge. However, the model generality at this stage is typically insufficient. The model is usually designed for a specific problem. When the problem changes, the model is no longer valid.

Since 2015, most of the research studies on few-shot learning have focused on neural networks [14–18]. For example, the same model can be used for cat and dog recognition and face recognition under the small sample. Except for the adjusted super parameters, the model itself is unchanged, resulting in the analysis and modeling of few-shot learning problems from various angles. Currently, few-shot learning can be divided into the model-, optimization-, and metric-based methods.

(i) The model-based approach [19, 20] understands the task through a model and stores the knowledge acquired. When decisions are required, they can be made by learning effective models. As early as 2001, the memory-based neural network method was proved to be applicable to meta-learning [19], whereby the bias and output were adjusted by updating weights and learning to quickly cache expressions into the memory, respectively. The authors used long short-term memory (LSTM) and other recurrent neural network (RNN) to treat the model data as a sequence for training and inputted new class samples for classification during testing. Meta networks [20] were designed for the rapid generalization between tasks by introducing the concept of fast weight. In particular, the gradient descent algorithm is used to optimize the weights in the neural networks, which is typically a slow process. It can be accelerated with the use of an additional neural network to predict the weights, denoted as fast weight. And the weight optimized by the general gradient descent algorithm is denoted as the slow weight. In the meta-network, the lost gradient is provided as meta-information to learn the fast weight model. The slow and fast weights are subsequently combined for the final output.

(ii) The ability that fulfills the few-shot learning is employed in the optimization-based few-shot learning models. For example, Finn et al. proposed the model-agnostic meta-learning (MAML) in 2017 [21]. MAML is a general optimization algorithm that expands the differential calculation process through the computational diagrams of the gradient descent method and learns a model that includes tasks but not samples. Nichol et al. proposed Reptile [22], a simple meta-learning optimization algorithm that is similar to MAML. For example, Reptile and MAML are gradient-based meta-optimizations and are model-independent. The optimizer performs a multistep gradient descent algorithm on each training task and updates the model with the results of the last step.

(iii) Metric learning tasks learn a function that measures the distance between different objects. Measurement learning can be combined with distance and similarity-based methods, such as K-mean clustering [23], the K-nearest neighbor method [24], support vector machine [25], and other algorithms that require a given measure to reflect the important relationship among data. In the Siamese network, two networks with the same parameters are used to extract features of the two samples. Then, the extracted features are inputted into the discriminator to determine whether the two samples belong to the same object class [26]. The matching network builds encoders for the supporting and query sets, and the output of the final classifier is the weighted sum of the predicted values between the supporting and query set samples [27]. The prototype network maps the sample data in each category to a given space and extracts their "mean" and Euclidean distance to represent the class prototype and distance measurement, respectively. Thus, the training data and class prototype exhibit the closest distance compared to other prototypes [28].

The above research studies on few-shot learning generally focus on images, while studies on the practical application of few-shot learning problems are limited. Thus, in the current paper, the problem of few shot learning in the commercial field is investigated because it remains a challenge. The work is performed in terms of two aspects. On the one hand, we proposed the probabilistic network (PN) method for the business field by adopting data enhancement to process the original dataset so as to improve the sensitivity of the model to business data. On the other hand, we ensure the effective application of the PN and experimental results of absenteeism at work. Las Vegas Strip hotels can valid that the PN method could identify the categories of commercial few-shot learning problems.

The following parts are structured as follows. The applications of the PN method for few-shot and one-shot learning are described in Section 2. The experimental results are presented in Section 3. Section 4 discusses the significance of business-related few-shot classification problems. The conclusion is drawn in Section 5.

2. Methodology

Few-shot learning is an integral component for research on the exploration of intelligent systems. Current smart systems are often based on large amounts of data analysis, while few-shot learning exhibits the learning ability of extracting inferences from a single example.

Unmarked samples can be encountered extensively in real-life applications, particularly in business field. The inaccurate prediction or data classification will directly affect the business development. However, traditional forecasting methods cannot always accurately foresee business problems. Therefore, more effective indicators and models must be built in order to improve the forecasting accuracy, understand the development trends of business problems, and promote economic growth.

2.1. Data Augmentation. Data enhancement allows a limited dataset to generate more data, increases the number and diversity of training samples (noise data), and improves the robustness of the model. Therefore, in order to effectively solve the problem of few-shot learning prediction for business applications, we expand the original dataset to increase the difference between the data, thus improving the final prediction. In particular, we collect and learn the mathematical distribution of real sample data X and add noise data λ to generate new sample data X^*:

$$X^* = (X, \lambda)^T (X, \lambda). \tag{1}$$

New sample data X^* can be described by n samples as follows:

$$X^* = (x_1, x_2, \ldots, x_n), \tag{2}$$

thus reflecting the diversity of the data.

2.2. Problem Definition. In the current paper, we consider the one- and few-shot classifier learning tasks. Each task is made up of four datasets, namely, the training, test, support, and query sets. The support and query sets share the same tag space. The goal of the task is to decide which class the query belongs to.

2.2.1. Training. The episode training strategy is adopted in the training phase of the proposed method. This strategy breaks the training process into multiple episodes and subsequently performs the classification on the task for each episode. The few-shot problem training set contains several categories, each of which has many samples. In the training stage, Q categories will be randomly selected from the training set, with K samples in each category (total $Q * K$ data) to construct a subtract task. The subtract task will then be inputted as the support set (SS) of the model:

$$\mathrm{SS} = (x_i, y_i), \quad i = 1, 2, \ldots, Q * K, y_i \in 1, 2, \ldots, Q. \tag{3}$$

A sample of the remaining data from the Q classes is then extracted as the model's query set (QS):

$$\mathrm{QS} = (x_i, y_i), \quad i = 1, 2, \ldots, Q * (m - K), y_i \in 1, 2, \ldots, Q, \tag{4}$$

where m is the total number of sample of Q classes. Table 1 reports the rules used for the division of X^* to determine SS and QS.

In Table 1, $\mathrm{random\,sample}(\mathrm{SS}, N)$ takes a random sample N without returning it back to SS; X^*/SS denotes the elements that are in collection X^* but not in collection SS.

In the business field, a Q-way K-shot problem denotes the model used to learn how to distinguish the Q categories from the $Q * K$ data and is considered as a business task. Thus, each episode will have distinct business tasks throughout the entire training process. We denote the entire training process as T_{train}, while each trained business task is defined as $T_{\mathrm{train}}(\tau)$, $\tau = 1, 2, \ldots, E$, where E is the number of business tasks. The business tasks satisfy $T_{\mathrm{train}}(\tau) \in T_{\mathrm{train}}$, $\tau = 1, 2, \ldots, E$. $T_{\mathrm{train}}(\tau)$ can be described as follows:

$$\begin{aligned} T_{\mathrm{train}}(\tau) &= \mathrm{SS}_\tau \bigcup \mathrm{QS}_\tau \\ &= \left\{ (x_1^\tau, y_i^\tau), (x_2^\tau, y_2^\tau), \ldots, (x_{QK}^\tau, y_{QK}^\tau) \right\} \bigcup \left\{ (x_1^\tau, y_i^\tau), (x_2^\tau, y_2^\tau), \ldots, (x_{Q(m-K)}^\tau, y_{Q(m-K)}^\tau) \right\}. \end{aligned} \tag{5}$$

Next, we employ the training set to constantly adjust the parameters and determine the final training model.

2.2.2. Test. The evaluation stage adopts the same framework as the training stage, but with data taken from the evaluation dataset. The test process is defined as $T_{\mathrm{test}}(\tau)$, $\tau = 1, 2, \ldots, E$ and also contains the test set (TS), which described as follows:

$$\mathrm{TS} = (x_i, y_i), \quad i = 1, 2, \ldots, Q * K, y_i \in 1, 2, \ldots, Q. \tag{6}$$

The datasets are then used to evaluate the generalization ability of the final model.

2.2.3. Task. The category labels for the sample in the query set are determined from the sample in the support set. Hence, the training and testing processes make up the entire learning task. For this task, we trained the model, adjusted the parameters, and predicted the data in order to solve the one- and few-shot problems. The learning task is defined as T_τ and satisfies the following formula:

$$\begin{aligned} & T_\tau := \left(T_{\mathrm{train}}(\tau) \bigcup T_{\mathrm{test}}(\tau), P(\mathrm{Q}|x_i), (x_i, y_i) \right), \\ & \tau = 1, 2, \ldots, E, \\ & i = 1, 2, \ldots, n, \\ & y_i \in \{1, 2, \ldots, q\}, \\ & Q \le q, \end{aligned} \tag{7}$$

where the dataset contains class q. We then divided the problem into two categories depending on category Q:

Problem 1. Few-shot business problem: it is defined by $k > 1$ and $\mathrm{SS} = QK$. The sample number of each class in the support set is greater than one, and the number of query sets is unconstrained. This problem is denoted as the few-shot problem of finance.

Table 1: Sampling rules of the dataset.

```
Rule: data sampling
Input: dataset $X^*$
Output: **support** set (SS), **query set** (QS)
1: $V \longleftarrow$ Random Sample ($\{1, 2, \ldots, n\}, Q$)
2: for $i$ in $\{1, 2, \ldots, Q\}$:
3:     $SS_i \longleftarrow$ Random Sample ($X^*_{V_i}, K$)
4:     $QS_i \longleftarrow$ Random Sample ($X^*_{V_i}/SS_i, t(m - K)$)
5: return SS, QS
```

Problem 2. One-shot business problem: this is defined by $k = 1$, where the sample numbers of each class is one in the support set, and the number of query sets is unconstrained. This problem is denoted as the one-shot problem of finance.

2.3. PN Category Method for the Few-Shot Problem.

2.3. PN Category Method for the Few-Shot Problem. In this section, the classification of the few-shot problem is described.

2.3.1. Feature Embedding. The neural network is widely used to construct new feature spaces, as it is considered as a successful technique in feature embedding based on their strong learning ability. Convolutional neural networks are locally connected neural networks with advantages in mapping, classification, prediction, and learning speed. The feature embedding based on the convolutional neural network is performed under the following steps:

*S*1 (input layer): input sample x_i, and rewrite it as $\hat{x}_i$;

*S*2 (convolution layer): data $\hat{x}_i$ are scanned using the deep convolutional neural network $g(\hat{x}_i; \omega, b)$, where convolution kernel $\omega \in R$ is a learnable weight vector and $b \in R$ is the learnable bias, to obtain a feature map;

*S*3 (batch norm layer): batch norm standardization is adopted to ensure the equal distribution of the layer inputs. Thus, the network has the following form:

$$g(\hat{x}_i, t(\hat{x}_i, tEn(\hat{x}_i)qvarh(\hat{x}_i))n; q\omega h,_b), \tag{8}$$

where $E(\hat{x}_i)$ and $\text{var}(\hat{x}_i)$ are the mean and variance of word vectors $\hat{x}_i$, respectively.

*S*4 (rectified linear unit (ReLU) nonlinearity layer): in order to avoid linearization, the following activation function is used:

$$f(g(\hat{x}_i; \omega, b)) = f_g(\hat{x}_i), \tag{9}$$

where f is the ReLU activation function.

The parameters of the feature extraction network are maintained consistent for each task across the support and query sets. Figure 1 summarizes the feature extraction framework.

The convolutional layer aims to extract local features of the feature matrix, in order to reduce the computational complexity of the network and to obtain more representative features. Moreover, the batch norm has a fundamental influence on the training process by smoothing the solution space of the optimization problems. This smoothness ensures that the gradient is both predictable and stable, allowing for a wider range of learning rates and a faster network convergence. The batch norm is applied to the sensitive areas of the activation function, corresponding to the frontal section of the function in this paper. Training difficulties often occur during the training of deep networks, which is attributed to variations in the output data of the upper layer network, following each parameter iteration and update. Such variations are denoted as internal covariate shifts and complicate the network learning of the next layer. In order to reduce the internal covariate shift, each layer of the neural network is normalized.

2.3.2. Category Contrast. Once the feature map is obtained by randomly extracted samples from the support set $\hat{x}_i$and query set $\hat{x}_j$, a key problem arises related to the classes learning from a small set. In our method, the final decision is made by combining the query set sample with the information for each category via the category determination system.

The network processing of two feature maps $f_g(\hat{x}_i)$ and $f_g(\hat{x}_j)$ is compared, and the corresponding scores are calculated. Each score denotes the probability of each classification of the query set sample $\hat{x}_j$. It is noted that for $Q > 1$, we determine the Q scores. The feature map outputs are then summed element-by-element. Furthermore, the target function can be expressed as follows:

$$\begin{aligned}
&\max P(q|\hat{x}_j; \varphi(\text{combinate}(f_g(\hat{x}_i), f_g(\hat{x}_j)),\\
&s.t. \left|\varphi(\text{combinate}(f_g(\hat{x}_i), f_g(\hat{x}_j))\right| < \varepsilon, \hat{x}_i, \hat{x}_j \text{ belong to the same category},\\
&\left|\varphi(\text{combinate}(f_g(\hat{x}_i), f_g(\hat{x}_j))\right| > \varepsilon, \text{otherwise},
\end{aligned} \tag{10}$$

where $q = 1, 2, \ldots, Q$, $0 < \varepsilon < 1$. Probability P is determined using the probability network (PN) method, described in the following, where $\varphi(x)$ represents the network structure.

(1) Convolution layer: the convolution module contains a convolutional layer, a batch normalization layer, and a maximum pooling layer

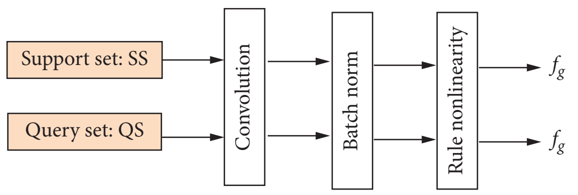


FIGURE 1: Feature embedding framework of the proposed method.

(2) Fully connected layer 1: it is a layer with an ReLU activation function and a layer that can combine all the local features into global features to calculate the final score for each category

(3) Fully connected layer 2: it is a linear layer with a Sigmod activation function

The neuron is connected by the fully connected layer to the neuron in the next layer. The fully connected layer can act as a "classifier" for the whole neural network, mapping the learned features to the sample marker space. A nonlinear activation function is introduced to the neural network to determine the categories of the query set samples.

At an input of 0, the ReLU activation function equals 0. But when output values equal the input value, the input is greater than 0. It increases the network sparsity, reduces the amount of computation, and enhances the convergence speed. What is more, it is commonly applied by current mainstream networks. When inputs are greater than 0, the slope is constant, and the gradient disappearance can be reduced by the ReLU.

The Sigmod function maps the input between [0, 1]. When the absolute value of the input is large, the output is close to 0 or 1 and output changes are minimal. The Sigmod function can be considered as a sample layer in the output data, whereby the greater output value means larger probability of sample labels. On the contrary, with smaller output data value, the sample label features less probability. Thus, as the input value increases, the effect of increasing probability is gradually diminished and vice versa, highlighting the benefits of nonlinear mapping.

The activation functions ensure that the classification probabilities of the query set sample are between 0 and 1. In theory, the sample category is determined based on the maximum output results. It effectively considers which class properties may be most relevant to the category of unknown sample. Figure 2 summarizes the category contrast.

2.3.3. Loss Function. Applying the sigmoid function as the neuron activation function, we employ the l_1 loss function to replace the variance cost function in order to avoid a slow training process. The l_1 loss function is frequently used in classification problems and represents the difference between the actual sample tag and the predicted probability. More specifically, the smaller the loss function value is, the closer the actual sample tag to the predicted probability will be.

The number of nodes in the last output layer is generally equal to the number of targets of the classification task. Let $N\,(\mathrm{QK})$ denote the final number of nodes; then, the neural network is able to determine an n-dimensional array as the output result for each sample, with each array dimension corresponding to a category. In the most ideal case, if a sample belongs to the class, then the output value of the output node corresponding to the category should be 1, while that of other nodes should be 0, namely [0,0,1,0, . . ., 0]. This array is taken as the sample label and is the most expected output result of the neural network. Thus, the l_1 loss function between the actual sample tag and the predicted model can be expressed as follows:

$$L_{\mathrm{loss}} = \frac{1}{M}\sum_{\tau=1}^{M}\left|P\left(q|\hat{x}_j;\varphi\left(\mathrm{combinate}\left(f_g\left(\hat{x}_i\right),f_g\left(\hat{x}_j\right)\right)\right)-\delta\left(\hat{x}_i,\hat{x}_j\right)\right)\right|,$$
$$\delta=\begin{cases}1, & \hat{x}_i,\hat{x}_j \text{ belong to the same category},\\ 0, & \text{otherwise}.\end{cases} \tag{11}$$

In which, M expresses the number of subtasks to learn a model. The norm in the loss function ensures a stable gradient, irrespective of the input value. Hence, this is a relatively robust solution that avoids the gradient explosion.

2.3.4. Training Strategy. The effective feature learning for a relatively small amount of labeled data proves to be a complicated task. These rare categories need to be generalized without additional training due to the associated costs and long cycles. In order to enhance the classifier generalization capability, few-shot learning generally employs episodic training. Thus, a large number of subtasks T_τ (or historical tasks) similar to the target task are taken to learn a model. A reasonable initial model value is then obtained by acting on the target task, such that the model can rapidly adapt with just a small amount of data for the target task. However, the model needs to combine previous experience with the information learned by the few-shot approach of the current new task. In addition, overfitting must be avoided. We take the entire classification task as T, which can be described in terms of subtasks T_τ:

$$T=\sum_{\tau=1}^{M}T_\tau, \tag{12}$$

where T_τ satisfies equation (7).

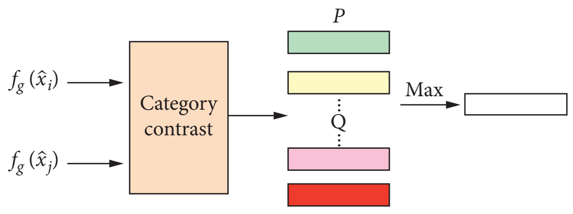


Figure 2: Schematic diagram of the category contrast process.

The model predicts the sample labels for each subtask in the query set for a given support set, where the purpose of training is to minimize the prediction error. This process can be considered as meta learning as the training process explicitly uses a given support set for learning in order to minimize the query set error. The l_1 loss function between the actual sample tag and the predicted model is adopted in this paper. And the smaller loss function can produce a robust model, which is in the following expression:

$$\min L_{\text{loss}} \varphi\Big(\text{combinate}\left(f_g(\hat{x}_i), f_g\big(\hat{x}_j\big)\right), \delta\big(\hat{x}_i, \hat{x}_j\big)\Big). \quad (13)$$

The training process described in equation (5) will output a set of parameters. When faced with new data categories, the model will exhibit a strong performance under these parameters. The sampled tasks are distinct when training is performed each time. Hence, the training includes numerous category combinations. It enables the model to learn the common components of different tasks, extract important features, and perform category comparisons. The models learned through this learning mechanism can effectively classify new tasks. Figure 3 summarizes the training process.

2.4. Category Method of the One-Shot Problem. For the one-shot problem, $K = 1$, namely, the query set only has one sample for each category. A convolutional neural network is adopted to accurately extract the sample features, followed by the implementation of the batch normalization layer and the ReLU activation function layer. Moreover, the parameters of the feature extraction network are consistent for each task across the support and query sets.

The network structure imitates that of the feature extractor to attain the final category. At this stage, we employ the convolution layer, batch normalization, pooling layer, fully connected ReLU, and the fully connected sigmoid nonlinearity layer. The single sample in the query set determines the final sample category by comparing the relationship between each category and assessing the category based on the output results.

3. Results

Few-shot learning consists of the training and testing stages. As the datasets used in the training and evaluation stages have no common category, episode training is typically employed in the training stage to ensure that the few-shot learning is strictly met. The training process is divided into multiple episodes by the strategy, whereby each episode requires the sampling of the task data for the task classification. The same episode training strategy is used for the testing stage. The only difference is that the data are sampled from the evaluation dataset. More specifically, the evaluation phase needs to sample the support and query sets from the evaluation dataset. The few-shot learning model needs to determine the category label of the samples in the query set based on the support set samples.

The same network and training hyperparameters are used for each task. The networks were trained in PyTorch by using the Adam optimizer with a learning rate of 0.001. Each model is trained for 100 or 200 epochs, with $Q = 3$ and $Q = 5$ per epoch. The data were obtained from the UCI Machine Learning Repository datasets (https://archive.ics.uci.edu). Based on the few-shot learning size of the business industry, we selected two datasets to construct the model datasets following definitions 1 and 2.

The dataset selection should follow two criteria. On the one hand, the data must be authentic, so the datasets come from a public database to reflect the persuasive of the results. On the other hand, it is difficult to obtain a large number of samples or marker samples in the dataset domain. Only in this way, can the value of proposed method be reflected in this paper.

The proposed method is compared with commonly used intelligent prediction methods including the K-nearest neighbor [29], logistic regression [30], decision tree [31], and Naive Bayes [32] algorithms. The selection of the training, validation, and test sets is consistent across methods to ensure a horizontal comparison of the differences in accuracies. However, the support, query, and test set samples are randomly selected to ensure differences in the results of each calculation.

3.1. Few-Shot Business Problem. The training set contains numerous categories, with multiple samples in each category. At the training stage, 5 categories are randomly selected from the training set, with 5 samples for each category (total of 25 samples) used to construct a support set. The model learns how to distinguish the 5 categories within the 25 samples by extracting a sample batch from the remaining class data as the query set. In addition, the proposed method also learns how to distinguish the 3 categories within the 15 samples by extracting a sample batch from the remaining class data as the query set.

The K-nearest neighbor, logistic regression, decision tree, and Naive Bayes algorithms adopt 5 categories to train the models and 5 categories within the 25 samples to test the models.

3.1.1. Example 1 (Absenteeism at Work). It is common to observe that employees sometimes are late for work, leave early, or are absent at work, owing to health problems and family crises. It occasionally causes absenteeism. However, such phenomenon rarely occurs in the company; thus, absenteeism datasets are small. Under such circumstances, the data mining process can be complicated, thus preventing an effective classification by traditional methods. In the

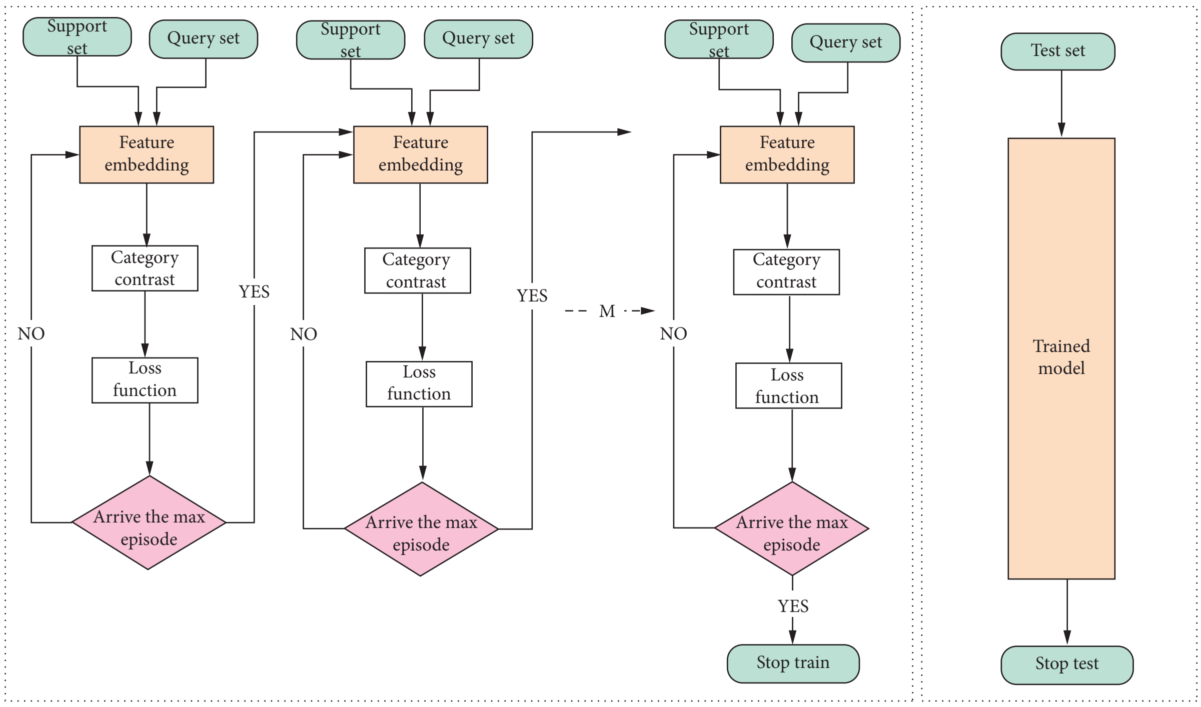


Figure 3: The application of the PN method for few-shot learning.

current paper, absenteeism records are selected from a Brazilian courier company to determine employee absenteeism. This dataset contains 740 samples and 21 variables. Variables include individual identification (ID), reason for absence, month of absence, day of the week, season, age, workload average, education, drinking status, and absenteeism time in hours.

In the simulation experiment, the convolutional neural network is employed to optimize the model and train the parameters by using Python. Table 2 indicates the accuracy of the test process. We consider both the recognition and identification accuracy. The parameter derivative typically changes to 0, preventing the parameter from updating. Figure 4 depicts the loss function changes during the training process.

The classification model proposed in this paper with other machine learning predictions (K-nearest-neighbor, logistic regression, decision tree, and Naive Bayes algorithms) is given in Table 2. The accuracy of the proposed method is higher than other machine learning prediction algorithms, which indicates our model is suitable to solve few-shot learning problems in the commercial field. In particular, our model can improve the accuracy of both the testing and classification processes.

The loss function convergence is executed with 100 iterations (Figure 4). It is close to 0.1 at iteration 100. This indicates the successful classification effect of the model as well as the ability of the proposed method to classify 3-category and 5-category problems (i.e., $Q = 3, 5$ in Problem 1).

Thus, it is demonstrated that the proposed model can not only deal with few-shot learning absenteeism effectively but also shows great advantages compared with other methods.

The working time is an important labor resource controlled by individuals. Workers achieve a given work performance by completing tasks within a limited time. With excessive absenteeism, the work will be completed with low efficiency, which can affect the performance appraisal of employees [33]. Therefore, a worker must effectively organize their working time. When the working hours are given within a reasonable range, the employees can fulfill their duties within a sufficient time frame. However, the employee will face greater time pressure if the working time is lower than the reasonable range, which consequently exerts a negative impact on work performance. Absenteeism is not only conducive to the management of employees but also affects the normal operation of the enterprise. Therefore, employee absenteeism has a far-reaching impact on business activity. It is of positive significance for employees and enterprises to effectively identify employees' absenteeism and reasonably predict the time when they are absent. It can promote the development of enterprises and bring benefits to their work performance.

3.1.2. Example 2 (Las Vegas Strip). With the development of the Internet and e-commerce, online commenting has gained wide popularity. Online user reviews serve as a vital information resource for consumers, playing a crucial role in the decision-making of potential customers. More than 80% of online consumers will refer to the comments made by other consumers before making purchasing decisions. It is believed that such information is more authentic than the information provided by sellers. Research shows that consumer reviews have a significant impact on product sales.

Table 2: Comparison of classification results on absenteeism with $k > 1$.

Method	$\mathbf{Q} = 3$	$\mathbf{Q} = 5$
K-nearest neighbor	–	42%
Logistic regressive	–	43%
Decision tree	–	45%
Bayes	–	39%
PN	94%	98%

The online ecosystem of the hotel and tourism industry is both complex and diverse. In the current paper, we focus on the lack of online reviews in this industry, taking 21 hotels in the Las Vegas Strip as an example. The dataset contains 504 records and 20 tuned features, including traveler type, tennis court, casino, free Internet, hotel name, hotel stars, no. of rooms, user continent, member years, and review month.

We follow the same classification methodology as the previous example in Section 4.1.1. Table 3 compares the accuracy of the results by using the methods. The accuracy of proposed is higher than K-nearest neighbor, logistic regression, decision tree, and Naive Bayes algorithms. Therefore, it is proved that the PN method is more applicable for few-shot learning.

The loss function is taken as the benchmark to determine the weight parameter minimized in equation (5). Figure 5 presents the variations in the loss function curve of the training samples, and it is close to 0.1 at iteration 200. It is observed that the PN method can effective treatment the Las Vegas Strip dataset.

Our method outperforms the traditional machine-learning in terms of accuracy (Table 3). It verifies the effectiveness of our method in dealing with problems in the business. Compared to the traditional approaches, our method obtains improved weight parameters from the training data, with stable variations in loss function (Figure 5). Thus, it is demonstrated that the proposed model can not only effectively deal with few-shot learning Las Vegas Strip but also shows great advantages compared with other methods.

Online reviews typically included reviewer information, a review rating, and review content. Consumers generally describe information related to product attributes or its performance. The review plays an important role in the purchasing decisions of potential consumers, particularly in the hospitality industry. Prior to consumption, consumers are unable to make reasonable judgments on the quality of the hotel. If a hotel is located in a city unfamiliar to consumers, online comments is of vital significance. As comments are not limited by location, users can quickly find the required information. Essentially, online reviews provide an important basis for potential consumers to make purchasing decisions and will also affect the profit of the hotel industry. Therefore, it is highly important for hotels to identify online review information. The earlier stage witnesses less the hotel online review information. It needs a small number of sample hotel online evaluation problem classification methods and accurate identification of customer evaluation. According to the classification results, the hotel can attract customers and provide better service by adjusting the business model.

3.2. One-Shot Business Problem. The training set contains numerous categories, with multiple samples in each category. During the training stage, 5 categories are randomly selected from the training set, with one sample for each category (a total of 5 samples) used to construct a support set. The model learns how to distinguish the 5 categories from the 5 samples by extracting a sample batch from the remaining data of the 5 classes as the query set. In addition, the proposed method also learns how to distinguish the 3 categories from the 3 samples.

The K-nearest neighbor, logistic regression, decision tree, and Naive Bayes algorithms also adopt the 5 categories to train the models.

3.2.1. Example 3 (Absenteeism at Work). In order to further verify the PN method in dealing with a one-sample problem, we continue with the problem in example 1 and take a single sample from each sample. The single sample is used to train the model, update the parameters, and determine the final classification model.

The performance of the PN method surpasses that of the other methods (Table 4). In particular, the accuracy of the PN method is 92% with $Q = 5$, exceeding the traditional approaches (the K-nearest neighbor, logistic regression, decision tree, and Naive Bayes algorithms) by approximately 50%. In addition, the PN loss function stabilizes, followed with 100 training iterations, which indicates the stable change in parameters (Figure 6). Therefore, the proposed PN method can effectively deal with single-sample absenteeism problems.

3.2.2. Example 4 (Las Vegas Strip). Similarly, in order to further verify the effectiveness of the proposed method for single-sample problems, we evaluate one-sample online hotel rating data. We compared the accuracy of the PN method proposed in this paper with that of traditional approaches. Table 5 reports the results, and the accuracy of proposed PN method are higher than K-nearest neighbor, logistic regression, decision tree, and Naive Bayes algorithms, and Figure 7 presents the loss function diagram, revealing the ability of the proposed PN method to handle the one-sample Las Vegas Strip problem.

4. Discussion

The rapid development of artificial intelligence in recent years is largely based on data development, with the increasing focus of emerging artificial intelligence put on big data. However, the data available for many real-world scenarios are limited. In this way, the acquired data can hardly be guaranteed as the massive data. Under such circumstances, conventional algorithms can hardly handle such issue, but it can be overcome through our method. Few-shot learning is able to deal with small dataset tasks. In

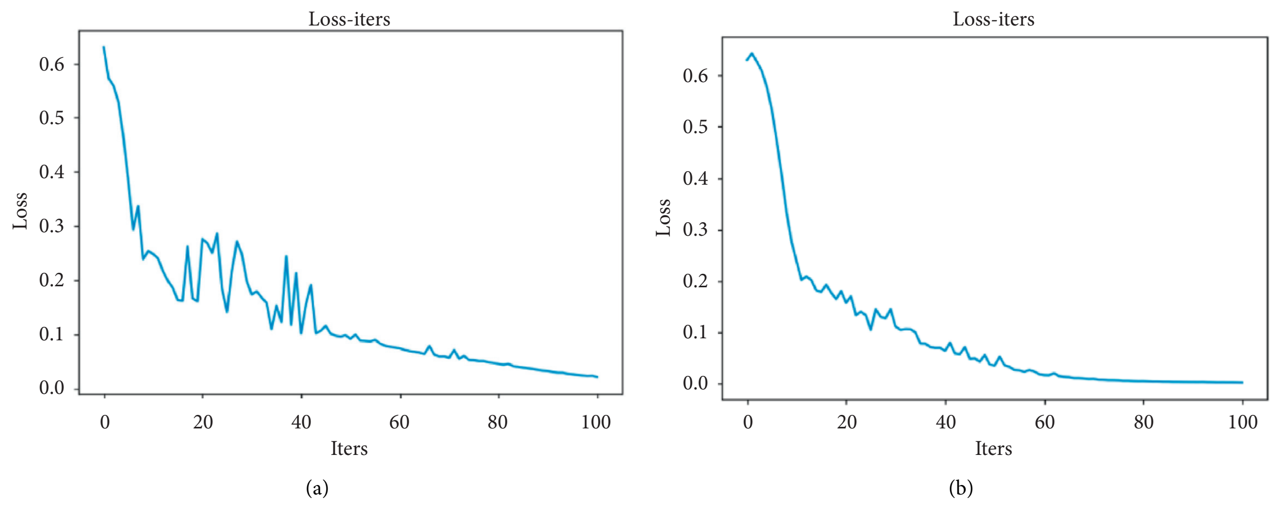


Figure 4: Variations in the loss function determined from data on absenteeism with $k > 1$. (a) $Q = 3$. (b) $Q = 5$.

Table 3: Comparison of classification results of the Las Vegas Strip dataset for $k > 1$.

Method	**Q** = 3	**Q** = 5
K-nearest neighbor	–	38%
Logistic regressive	–	39%
Decision tree	–	41%
Bayes	–	35%
PN	87%	89%

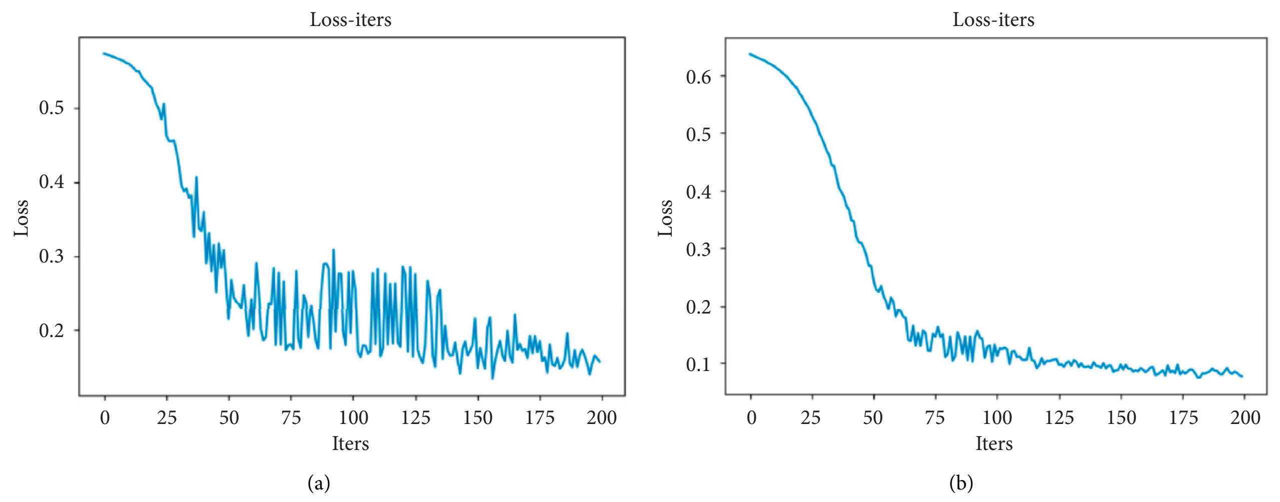
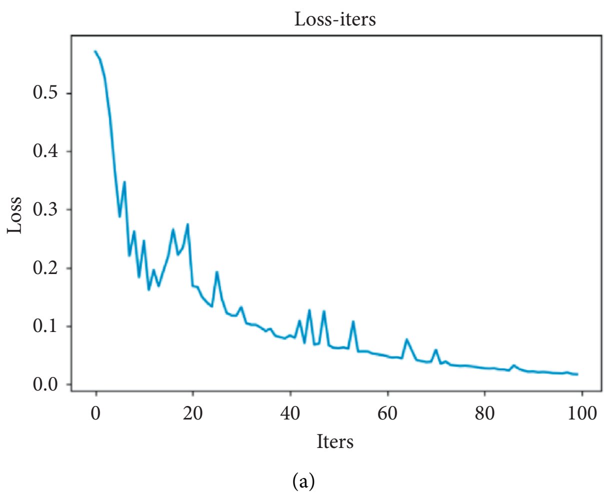


Figure 5: Variations in the loss function of the Las Vegas Strip dataset for $k > 1$. (a) $Q = 3$. (b) $Q = 5$.

Table 4: Comparison of classification results for absenteeism with $k = 1$.

Method	**Q** = 3	**Q** = 5
K-nearest neighbor	–	43%
Logistic regressive	–	43%
Decision tree	–	44%
Bayes	–	38%
PN	92%	95%

particular, it describes the task of learning from a few examples, which poses a great challenge for current machine learning algorithms. In applied statistics, few-shot learning and large samples are generally associated with smaller and larger sample sizes, respectively. In the current paper, it is used to describe a small amount of data and marked samples, a business that has failed to get established in the early stage or a large number of samples. However, accurate predictions are required to aid enterprises to expand their business and



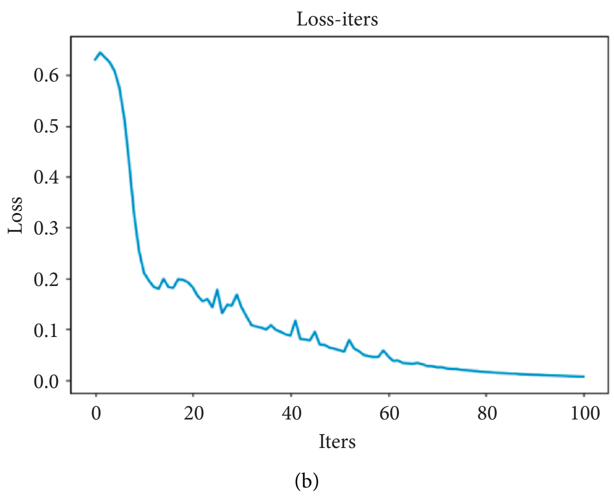
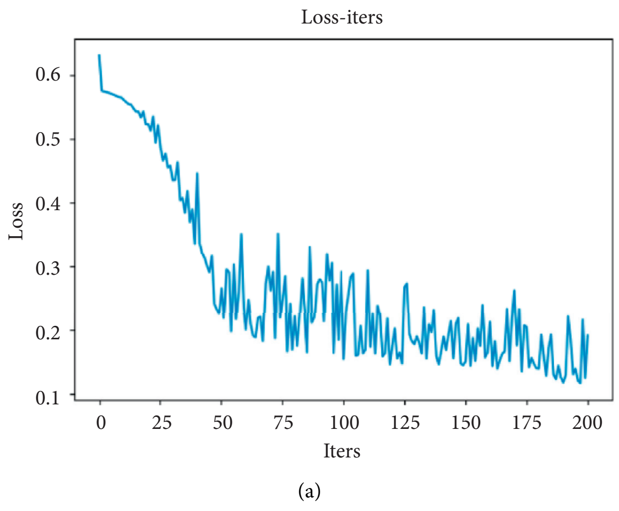


FIGURE 6: Variations in the loss function determined with $k = 1$. (a) $Q = 3$. (b) $Q = 5$.

TABLE 5: Comparison of classification results for the Las Vegas Strip dataset with $k = 1$.

Method	**Q** = 3	**Q** = 5
K-nearest neighbor	–	36%
Logistic regressive	–	38%
Decision tree	–	42%
Bayes	–	36%
PN	80%	85%



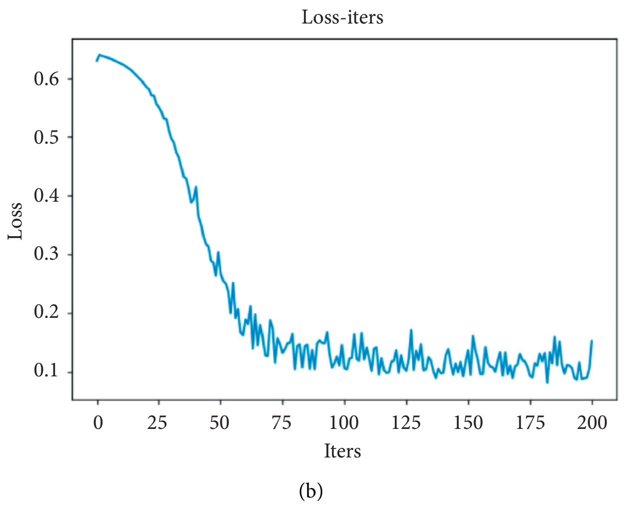


FIGURE 7: Variations in the loss function for the Las Vegas Strip dataset for $k = 1$. (a) $Q = 3$. (b) $Q = 5$.

enhance their competitiveness and service. The few-shot learning problem boils down to the lack of information in essence; thus, the potential data information cannot be fully mined by machine learning. To overcome this, a more intelligent technique is needed to effectively handle fewer sample tasks and achieve a higher model accuracy.

Solving the business-related few-shot problem can create new opportunities and challenges for the development of the business industry. Through the strengthened application and guidance of science technology, business enterprises can achieve sustainable and high-quality development, thus enhancing their competitiveness and maximizing enterprise trading profits.

5. Conclusion

In the current paper, the few-shot learning for business applications is investigated. In particular, we developed the PN method with feature extraction, analogy comparisons, and loss function analysis. Furthermore, two case studies are adopted on the prediction of absenteeism and online hotel reviews to validate and prove the generalization ability of the proposed method. It is demonstrated by the results that the proposed method is of high accuracy in dealing with these two problems and has the stable variation of the corresponding loss function. Thus, the method proposed in this paper can witness a successful prospect by being applied to commercial few-shot learning. However, the business-related few-shot regression problems await to be solved, which can enrich the predicted algorithm. Even though further details are still under dispute, yet it remains as the primary direction of future research.

Data Availability

The data used to support the findings of this study were obtained from the UCI machine learning repository datasets (https://archive.ics.uci.edu).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally to the manuscript and typed, read, and approved the final manuscript.

Acknowledgments

This work was supported by the foundation of Beijing Information Science and Technology University (2025034).

References

[1] Z. Zhang, Z. L. Guan, J. Zhang, and X. Xie, "A novel job-shop scheduling strategy based on particle swarm optimization and neural network," *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 699–707, 2019.

[2] D. Lapkova, "Education in professional defense-possibilities of classification of training level with the help of impulse," *Journal of System and Management Sciences*, vol. 8, no. 1, pp. 23–44, 2018.

[3] J. Yoon and S. Joung, "A big data based cosmetic recommendation algorithm," *Journal of System and Management Sciences*, vol. 10, no. 2, pp. 40–52, 2020.

[4] H.-M. Afify, K.-K. Mohammed, and A.-E. Hassanien, "Multi-images recognition of breast cancer histopathological via probabilistic neural network approach," *Journal of System and Management Sciences*, vol. 1, no. 2, pp. 53–68, 2020.

[5] L. Qin, N. Yu, and D. Zhao, "Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video," *Tehnicki Vjesnik*, vol. 25, no. 2, pp. 528–535, 2018.

[6] R. H. Frank, T. Gilovich, and D. T. Regan, "The evolution of one-shot cooperation: an experiment," *Ethology and Sociobiology*, vol. 14, no. 4, pp. 247–256, 1993.

[7] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[8] S. Taasan, "One shot methods for optimal control of distributed parameter systems 1: finite dimensional control," *ICASE Report*, vol. 91, no. 2, pp. 1–20, 1991.

[9] F.-F. Li, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1134–1141, Paris, France, October 2003.

[10] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[11] B.-M. Lake, R. Salakhutdinov, and J.-B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6, pp. 1332–1338, 2015.

[12] B.-M. Lake, C.-Y. Lee, J. Glass, and J. Tenenbaum, "One-shot learning of generative speech concepts," *MIT Education*, vol. 36, 2014.

[13] B.-M. Lake, R. Salakhutdinov, and J.-B. Tenenbaum, "One-shot learning by inverting a compositional causal process," *Advances in Neural Information Processing Systems*, vol. 2, 2015.

[14] Z. Ji, X.-L. Chai, Y.-L. Yu, Y.-W. Pang, and Z.-F. Zhang, "Improved prototypical networks for few-shot learning," *Pattern Recognition Letters*, vol. 140, pp. 81–87, 2020.

[15] X. Liu, F. Zhou, J. Liu, and L. Jiang, "Meta-learning based prototype-relation network for few-shot classification," *Neurocomputing*, vol. 383, pp. 224–234, 2020.

[16] Z. Chen, W. Ma, N. Xu, C.-T. Ji, and Y.-L. Zhang, "SiameseCCR: a novel method for one-shot and few-shot Chinese CAPTCHA recognition using deep siamese network," *IET Image Processing*, vol. 14, no. 12, pp. 2855–2859, 2020.

[17] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2017, https://arxiv.org/abs/1711.04043.

[18] Y. Zhu, W. Min, and S. Jiang, "Attribute-guided feature learning for few-shot image recognition," *IEEE Transactions on Multimedia*, vol. 99, p. 1, 2020.

[19] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," 2016, https://arxiv.org/abs/1605.06065.

[20] J.-L. Lancaster, A.-R. Laird, P.-M. Fox, E.-G. David, and T.-F. Peter, "Automated analysis of meta-analysis networks," *Human Brain Mapping*, vol. 25, no. 1, p. 174, 2010.

[21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1126–1135, JMLR. org, Sydney, Australia, August 2017.

[22] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, https://arxiv.org/abs/1803.02999.

[23] S. Chakraborty and S. Das, "k−means clustering with a new divergence-based distance metric: convergence and performance analysis," *Pattern Recognition Letters*, vol. 100, pp. 67–73, 2017.

[24] H. A. Vrooman, C. A. Cocosco, F. van der Lijn et al., "Multi-spectral brain tissue segmentation using automatically trained k-nearest-neighbor classification," *Neuroimage*, vol. 37, no. 1, pp. 71–81, 2007.

[25] L. Hu, J. Hu, Z. Ye, C. Shen, and Y. Peng, "Performance analysis for SVM combining with metric learning," *Neural Processing Letters*, vol. 48, no. 3, pp. 1373–1394, 2018.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” *ICML Deep Learning Workshop*, vol. 2, 2015.

[27] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” *Advances in Neural Information Processing Systems*, vol. 12, pp. 3630–3638, 2016.

[28] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” *Advances in Neural Information Processing Systems*, vol. 3, pp. 4077–4087, 2017.

[29] M.-R. Nikoo, R.-A. Kerachian, R. A. Mohammad, and R. Mohammad, “A fuzzy KNN-based model for significant wave height prediction in large lakes,” *Oceanologia*, vol. 60, no. 2, pp. 153–168, 2018.

[30] J. A. Mccarty and M. Hastak, “Segmentation approaches in data-mining: a comparison of RFM, CHAID, and logistic regression,” *Journal of Business Research*, vol. 60, no. 6, pp. 656–662, 2007.

[31] A. Jordan and M. Danijela, “Upgrading the business intelligence system by implementing the decision tree model in the *R* software package,” *Studies in Informatics and Control*, vol. 29, no. 2, pp. 243–254, 2020.

[32] J. Cao, R. Panetta, S. Yue, A. Steyaert, and M.-B. Young, “A naive Bayes model to predict coupling between seven transmembrane domain receptors and G-proteins,” *Bioinformatics*, vol. 19, no. 1, pp. 234–240, 2003.

[33] Y. Lin, W. Liu, and Y. Wang, “An integrated approach using cross-efficiency and shapley value in performance evaluation,” *Economic Computation and Economic Cybernetics Studies And Research*, vol. 53, no. 4, pp. 209–224, 2019.