Received: 16 December 2025 | Revised: 12 February 2026 | Accepted: 3 April 2026 | Published online: 12 May 2026

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2026, Vol. 00(00) 1–10
DOI: 10.47852/bonviewJCCE62028810

BON VIEW PUBLISHING

NCOHA-WSC: Handling Noise and Class Overlap in Web Service Classification

Feng Zhang[1], Lin Xue[1], Huiling Li[2] and Cong Liu[3,*]

[1] *School of Computer Science and Engineering, Shandong University of Science and Technology, China*

[2] *School of Computer Science and Technology, Shandong University of Technology, China*

[3] *NOVA Information Management School, Nova University of Lisbon, Portugal*

Abstract: With the rapid growth in the number of Web services, accurate and efficient Web service classification has become crucial for improving the quality-of-service discovery. However, existing classification approaches often overlook the issues of noise and class overlap inherent in Web service data, which leads to degraded classification precision. To address these challenges, this paper proposes NCOHA-WSC, an approach for Web service classification designed to handle both noise and class overlap and to be easily integrated with existing machine learning–based service classification models. Specifically, noisy samples in the training data are filtered using confidence learning and information entropy, thereby reducing the negative impact of noise on the classification model during preprocessing. In addition, during the testing phase, the prediction results for overlapping services are corrected based on the label prior distribution, further improving classification precision. Experiments conducted on the real-world ProgrammableWeb dataset demonstrate that NCOHA-WSC is compatible with mainstream Web service classification models and can enhance the Macro-F1 performance of models such as ServeNet and CARL-Net to varying degrees. These results indicate that the proposed approach effectively mitigates the impact of noisy data on Web service classification and improves the precision of existing models in the presence of overlapping service classes.

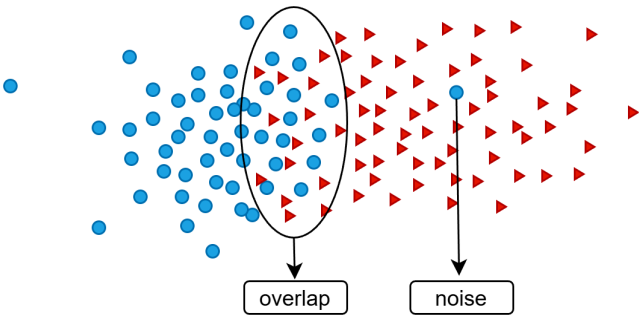
Keywords: Web service classification, noise handling, class overlap, confidence learning, prior distribution

1. Introduction

Amid the rapid evolution of service-oriented architecture, Web services have become a prevalent choice for building and deploying distributed systems [1]. The growing number of available Web services provides developers with a rich set of reusable business functionalities. However, this rapid growth also increases the difficulty of service discovery and selection. Service classification is an effective approach to improving service discovery efficiency, as it reduces the search space by grouping functionally similar services into predefined categories, thereby enabling the rapid identification of desired Web services. Consequently, Web service classification has become an important research focus in the field of service computing.

In recent years, the precision of Web service classification has been significantly improved through the use of deep learning techniques. However, the limitations of training data have become increasingly evident, as existing service classification approaches rely heavily on large-scale labeled datasets. As illustrated in Figure 1, real-world Web service data often suffer from noise [2] and class overlap issues [3], arising from factors such as poorly defined service categories, semantic gaps, and labeling errors. Noise refers to data instances with incorrect category labels,

Figure 1
Noisy and overlapping data



while class overlap describes cases in which services from different categories exhibit highly similar functional characteristics. Both issues adversely affect the precision of service classification models. First, the preprocessing of noisy data can be extremely time-consuming and may introduce additional labeling errors as the number of Web services continues to grow rapidly. Moreover, existing deep learning–based service classification approaches rely heavily on large-scale training data, and the presence of noise often leads to reduced classification precision and poor model robustness. Key factors hindering Web service processing, sorted

***Corresponding author:** Cong Liu, NOVA Information Management School, Nova University of Lisbon, Portugal. Email: cliu@novaims.unl.pt

by impact severity, are class overlap (highest, causing ambiguous predictions), data noise (degrading model robustness), dynamic system structural complexity, semantic sparsity of service descriptions, and class distribution imbalance (lowest). Consequently, the automatic filtering of noisy data has become a critical challenge for improving the precision of Web service classification.

Furthermore, recent studies have shown that the class overlap problem can be even more detrimental to classifier performance than class imbalance in certain datasets [4]. For instance, Figure 2 depicts two Web services labeled as *Messaging* and *Telephone*, whose functional descriptions are applicable to both categories [5]. Such instances are referred to as overlapping instances or high-confusion instances [6]. Unlike noisy data, overlapping data contain valuable categorical information that would be lost if they were indiscriminately removed, despite posing challenges to accurate classification. Therefore, enabling classification models to identify overlapping data and mitigate their impact is another critical issue for improving the performance of Web service classification.

Currently, there are two main approaches for handling noisy and overlapping Web service data. The first is data-level preprocessing, which includes techniques such as noisy word filtering [7] and high-quality training data selection [8]. The second focuses on algorithm-level improvements, such as incorporating service invocation information [9] and cost-sensitive learning strategies [10]. Although these approaches improve service classification performance to some extent, they still suffer from several limitations. First, vocabulary filtering may lead to inaccurate functional semantic representations of services due to the brevity of service descriptions. Second, most existing approaches are model-specific and lack the flexibility to be readily applied to other Web service classification models.

To address these challenges, this paper proposes a novel approach for handling noise and class overlap in Web service classification, termed NCOHA-WSC (Noise and Class Overlap Handling Approach for Web Service Classification). First, abnormal data in the training data are identified through confidence learning during the data preprocessing phase. Second, noisy data and overlapping data are distinguished, and the former are filtered based on information entropy. Finally, overlapping instances are identified, and their classification precision is further improved during the testing phase by applying a label prior correction algorithm [11]. The key contributions of this paper are as follows:

1) To tackle the problem of noisy data in Web services, this paper proposes an approach that integrates confidence learning and information entropy for noise filtering. Abnormal data in the dataset are first identified through confidence learning. Subsequently, information entropy is employed to distinguish the abnormal data into noisy data and overlapping data. Finally, noisy data are removed to reduce their negative impact on the classifier.
2) To mitigate class overlap in Web service classification, this paper proposes a predictive probability correction algorithm guided by label prior distributions. High-confusion instances in the test dataset are identified using information entropy, and their classification precision is further improved through the proposed label prior correction mechanism.
3) Experiments are conducted on a real-world Web service dataset, and model performance is evaluated using the Macro-F1 metric. The results demonstrate that the proposed approach is compatible with existing Web service classification models and consistently improves their classification performance.

Notably, the structure of dynamic Web systems—integrating multi-language development (e.g., Java, Python) and collaborative program conditions/commands (e.g., REST/SOAP protocols, XML/JSON data formats)—is a critical factor affecting Web service processing performance. Such structural characteristics inherently increase the likelihood of noise (e.g., inconsistent format labeling) and class overlap (e.g., cross-protocol functional similarity) in service data. NCOHA-WSC's model-agnostic design and lightweight core modules (linear-complexity noise filtering and probability correction) are well-suited to adapt to dynamic Web system architectures, ensuring compatibility with diverse development stacks while mitigating data-quality issues.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed NCOHA-WSC approach. Section 4 presents the experimental analysis. Finally, Section 5 concludes the paper.

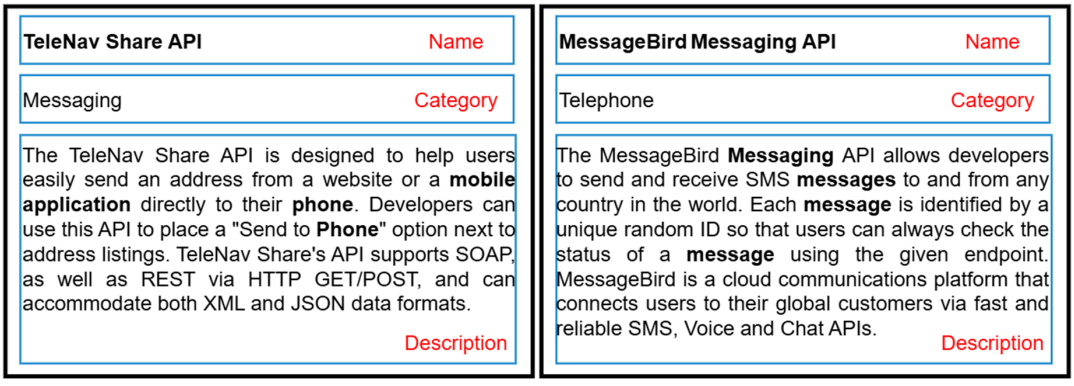
2. Related Work

This section first reviews existing work on Web service classification, followed by an introduction to the work related to handle noisy and overlapping data.

2.1. Web service classification

Web service classification is generally studied from two perspectives: quality-of-service-based service classification and functional semantic-based service classification [12]. Existing

Figure 2
Example of Web service class overlap



work predominantly targets the functional semantic perspective, which is commonly further grouped into two lines, namely structural feature-based service classification and text feature-based service classification.

Text feature-based service classification methods primarily aim to extract semantic information from service description documents via natural language processing techniques, including neural network-based text embeddings (Word2Vec and BERT) and topic modeling with latent Dirichlet allocation (LDA). Early service classification methods largely depended on traditional machine learning techniques to derive features from service descriptions, such as TF-IDF [13] and LDA [14]. Traditional approaches are hindered by their heavy reliance on manual feature engineering and the inherent semantic sparsity of service descriptions, limiting classification performance. The advent of deep contextual embedding models (BERT, RoBERTa) has transformed this landscape, enabling much richer automated feature extraction. Following this trend, Yang et al. [15] and Pan et al. [16] demonstrate significant performance gains by applying BERT and RoBERTa, respectively, to service classification.

Structural feature-based service classification methods mainly explore structural relations among Web services, including textual syntactic structures [17], service composition patterns [18], and tag-sharing links. Such relationships are commonly represented as graphs, and structural features are then learned via network representation learning methods, such as DeepWalk [19], Node2Vec [20], and other neural graph-embedding approaches. For instance, Liang et al. [21] build a heterogeneous network that captures both Mashup-API invocation interactions and tag-sharing links. On top of this network, they recommend API tags by combining random walk with restart and latent semantic analysis.

Although the aforementioned approaches enhance service classification performance by optimizing quality-of-service functional feature representations, they largely overlook the issues of noise and class overlap in Web service datasets, which limits further performance improvements.

2.2. Data noise processing

During the training of data-driven deep learning models, the quality of the training data has a significant impact on the generalization ability of the classifier. However, real-world datasets often contain substantial amounts of noisy data, which can cause the model to overfit noisy labels and consequently degrade its generalization performance. Existing approaches for handling noise can be broadly categorized into data-level and algorithm-level approaches. Data-level approaches typically retrain classification models after removing noisy data from the training dataset. A representative example is the confidence learning technique proposed by Northcutt et al. [22], which estimates the joint distribution of noisy and true labels to identify and filter noisy data before retraining the model, thereby mitigating the adverse effects of noise on model performance. Notably, this technique is model-agnostic and can be integrated with various classification models. Algorithm-level approaches focus on improving classification models to mitigate the adverse effects of noisy data. For example, Ortego et al. [23] propose an approach for detecting label noise by leveraging feature representations learned through contrastive learning to estimate soft labels for each sample. By comparing the soft labels with the original labels, noisy samples are identified and treated as unlabeled samples. The classifier is then trained in a semi-supervised manner, thereby improving overall classification performance.

2.3. Class overlap handling

Recent studies have demonstrated that perfect classification can be achieved on linearly separable datasets, even when class distributions are imbalanced. However, accurate classification becomes challenging when classes overlap, even in uniformly distributed datasets [4].

Consequently, class overlap poses a significant constraint on classification performance. Existing approaches for addressing class overlap can also be broadly categorized into data-level and algorithm-level approaches. Data-level approaches typically reduce the negative impact of overlapping data on classifiers by eliminating overlapping data through resampling techniques. For example, Vuttipittayamongkol et al. [24] propose an overlap-based under-sampling framework that employs a soft clustering algorithm to identify and eliminate majority-class data from potential overlapping regions.

However, these approaches may lead to the loss of valuable information and face challenges when applied to deep learning tasks, particularly those involving unstructured text data [25], as they mitigate the impact of class overlap on classifiers by eliminating data. Accordingly, Jia et al. [11] introduce a label prior–guided predictive probability correction algorithm that enhances classification on overlapping data without relying on data resampling. This algorithm leverages label prior information to adjust the predicted label distribution of overlapping data in the test dataset, thereby enhancing classification accuracy for overlapping cases.

3. NCOHA-WSC

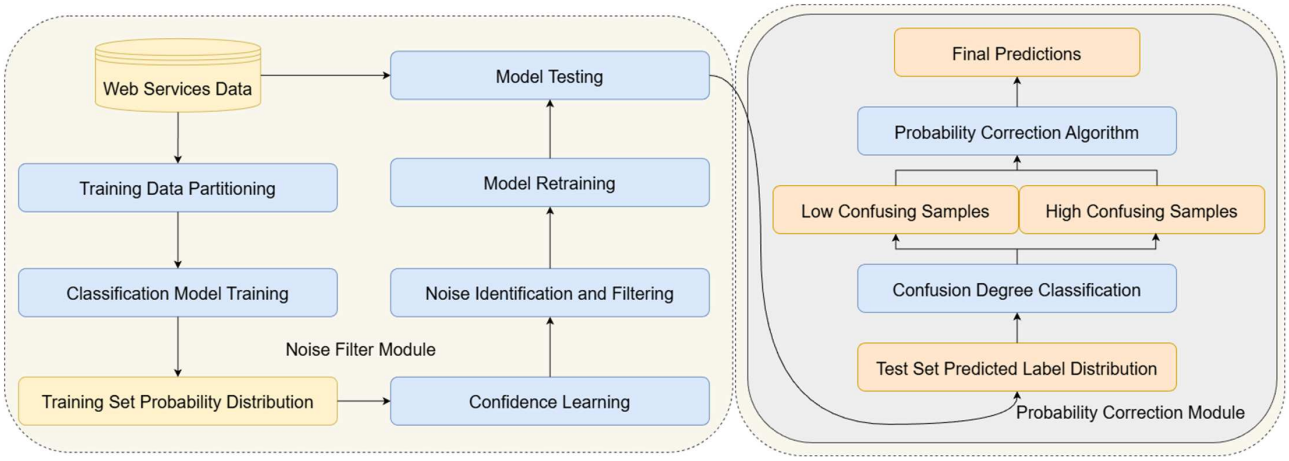
In this section, we address the problems of noise and class overlap in Web services using confidence learning and a prior correction technique. The overall framework of the proposed approach is illustrated in Figure 3.

During the data preprocessing phase, abnormal data in Web services—including noisy data and overlapping data—are identified through confidence learning. Normalized information entropy is then employed to distinguish between overlapping data and noisy data, and the noisy data are removed to mitigate their negative impact on the classification model. During the testing phase, overlapping data are further identified using normalized information entropy. Their predictive probability distributions are subsequently corrected based on the label prior distribution to improve classification precision. The proposed approach consists of two main modules: a noise filtering module and a probability correction module.

3.1. Confidence learning–based noise filtering

In deep learning–based Web service classification models, the quality of training data directly affects model performance. However, real-world Web service data often contain noise. To mitigate the adverse effects of noisy data, this paper proposes a confidence learning–based noise filtering approach for Web service classification. First, confidence learning is employed to estimate the conditional probability distribution between the manually labeled categories $\tilde{y}$ (i.e., the initial categories of the dataset) and the latent correct categories y^* (unknown), thereby enabling the identification of abnormal samples. Given the imbalanced class distribution

Figure 3
Overview of NCOHA-WSC



in Web service data [26], directly removing abnormal samples may adversely affect model training. Therefore, this paper employs information entropy to filter noisy samples while retaining overlapping samples. The overall process can be divided into three steps.

Potential correct label prediction. First, confidence learning requires the predicted probability distributions of the training data. To prevent overfitting, cross-validation is employed to obtain reliable predicted probabilities for the training data. For the h-th service x_h in the training dataset X, the predicted probability distribution $\hat{p}_{x_h} = [\hat{p}^1_{x_h}, \hat{p}^2_{x_h}, \cdots, \hat{p}^c_{x_h}]$ is obtained by the model θ. Here, $\hat{p}^j_{x_h} = p(\hat{y} = j; x_h, \theta)$ denotes the predicted probability of category $\hat{y} = j$ for service x_h and model θ, $j \epsilon \{1, \ldots, c\}$. The parameter c denotes the total number of categories, $h \epsilon \{1, \ldots, n\}$, and n denotes the number of services in the training dataset X. Subsequently, a noise threshold t_j is defined for each category j to filter out services whose predicted probabilities satisfy the corresponding noise criterion. The noise threshold t_j is defined as shown in Equation (1):

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \times \sum_{x \in X_{\tilde{y}=j}} \hat{p}^j_{x_h} \tag{1}$$

where $\sum_{x \in X_{\tilde{y}=j}} \hat{p}^j_{x_h}$ denotes the sum of the predicted probabilities of samples with initial category $\tilde{y} = j$ and predicted category $\hat{y} = j$. $X_{\tilde{y}=j}$ represents the subset of services in X with initial category $\tilde{y} = j$, and $X_{\tilde{y}=j}$ denotes the number of services in X with initial category $\tilde{y} = j$. The noise threshold t_j is defined as the average predicted probability of all samples belonging to category j.

Next, noisy sample filtering is performed, and the latent correct category of each sample is determined based on the confusion matrix derived from the model's prediction results and the noise threshold t_j. For each service x, the predicted probability distribution $\hat{p}_x = [\hat{p}_x{}^1, \hat{p}_x{}^2, \ldots, \hat{p}_x{}^c]$ is compared with the noise threshold t_j for each category $j \epsilon [1, c]$. The latent correct category $y_x{}^*$ of service x is selected as the category with the highest predicted probability among those whose predicted probabilities exceed the corresponding noise threshold t_j. If all predicted probabilities for service x are lower than their respective noise thresholds t_j, the service is classified as a noisy sample. The formal definition of the latent correct category for service x is given in Equation (2):

$$y^*_x = \left\{ arg\ max\ \hat{p}^j_x | \hat{p}^j_x \geq t_j \right\} \tag{2}$$

Calculation of the confidence joint distribution matrix. After obtaining the latent correct categories, the confidence joint distribution matrix $C_{\tilde{y},y^*}$ is constructed based on the model's prediction results on the training data. $C_{y=i,y^*=j}$ denotes the number of samples whose initial category is $\tilde{y} = i$ and whose latent correct category is $y^* = j$. Formally, the confidence joint distribution matrix $C_{\tilde{y},y^*}$ is calculated as shown in Equation (3):

$$C_{\tilde{y}=i,y^*=j} = \left| X_{\tilde{y}=i,y^*=j} \right| \tag{3}$$

where $|X_{\tilde{y}=i,\ y^*=j}|$ denotes the number of services in X whose initial category is $\tilde{y} = i$ and whose latent correct category is $y^* = j$. The confidence joint distribution matrix $C_{\tilde{y},y^*}$ differs from a conventional confusion matrix, as a portion of noisy services is filtered out using the noise threshold t_j. This filtering process increases the reliability of the matrix by making it more robust to categories with extremely large or small predicted probabilities. Subsequently, the estimated joint distribution matrix $\widehat{Q}_{\tilde{y},y^*}$ is obtained by normalizing the confidence joint distribution matrix $C_{\tilde{y},y^*}$. The matrix $\widehat{Q}_{\tilde{y},y^*}$ reflects the noise distribution associated with each category in the service training dataset X. The theoretical foundations underlying this estimation are detailed in Reference [22], and the corresponding calculation is given in Equation (4):

$$\widehat{Q}_{\tilde{y},y^*} = \frac{1}{n} \times \frac{C_{\tilde{y}=i,y^*=j}}{\sum_{j=1}^{c} C_{\tilde{y}=i,y^*=j}} \times \left| X_{\tilde{y}=i} \right| \tag{4}$$

where $|X_{\tilde{y}=i}|$ denotes the number of services whose initial category is $\tilde{y} = i$ and n represents the total number of services in the training dataset X. In addition, $\sum_{j=1}^{c} C_{\tilde{y}=i,y^*=j}$ denotes the total number of samples with initial category $\tilde{y} = i$ in the confidence joint distribution matrix $C_{\tilde{y},y^*}$.

Noise data filtering. In this phase, the number of potentially abnormal samples in each category is determined based on the estimated joint distribution matrix $\widehat{Q}_{\tilde{y},y^*}$. However, directly removing abnormal samples may substantially reduce the number

of samples in minority categories within the Web service dataset. This occurs because confidence learning defines overlapping samples as abnormal samples, while the number of truly noisy samples is substantially smaller than that of overlapping samples. Consequently, directly removing all abnormal samples may discard important classification-relevant information, ultimately degrading the model's classification performance. Therefore, noisy data and overlapping data are distinguished using normalized information entropy [27]. This process filters out noisy data while retaining overlapping data, thereby mitigating the data imbalance that would result from excessive removal. The specific procedure is as follows. First, the number of abnormal services num_i potentially present in each category i is calculated using the joint distribution matrix $\widehat{Q}_{\tilde{y},y^*}$. The calculation of num_i is given in Equation (5):

$$num_i = \left\lceil n \times \sum_{j=1;j\neq i}^{c} \widehat{Q}_{\tilde{y}=i,y^*=j} \right\rceil \tag{5}$$

where $\sum_{j=1;j\neq i}^{c} \widehat{Q}_{\tilde{y}=i,y^*=j}$ represents the sum of the entries in the estimated joint distribution matrix $\widehat{Q}_{\tilde{y},y^*}$ for which the initial category $\tilde{y}$ differs from the latent correct category y^*. This sum is then multiplied by the total number of samples n and rounded to obtain the number of abnormal samples in category i, denoted as num_i. Subsequently, for each category i, all samples $x \epsilon X_{\tilde{y}=i}$ are sorted in ascending order according to $P(\hat{y} = i; x, \theta)$. The first num_i samples with the lowest predicted probabilities are classified as abnormal samples. Finally, the normalized information entropy $H(p_x)$ is computed for each abnormal sample to quantify the uncertainty of its predicted probability distribution $p_x = [p_x{}^1, p_x{}^2, \ldots, p_x{}^c]$. The more uniform the predicted probability distribution p_x is, the higher the degree of uncertainty, and the more likely the sample is to belong to the overlapping data. Conversely, a less uniform distribution indicates a higher likelihood that the sample is noisy. The normalized information entropy is computed as shown in Equation (6):

$$H(p_x) = \frac{-\sum_{i=1}^{c} p_x^i \log p_x^i}{\log \frac{1}{c}} \tag{6}$$

where c denotes the total number of service categories. To control the number of noisy services, a confidence threshold β is set after obtaining the normalized information entropy for all abnormal samples. Specifically, if $H(p_x) \geq \beta$, the abnormal sample x is retained and treated as an overlapping sample. Otherwise, sample x is removed from the dataset and regarded as a noisy sample. The service classification model θ is then retrained using the cleaned dataset.

3.2. Labeled prior distribution–based predictive probability correction

Although eliminating noisy samples can improve the classification precision to some extent, the number of overlapping samples in the Web service dataset is significantly larger than that of noisy samples. Directly removing these overlapping samples may adversely affect model performance. However, retaining overlapping samples can lead to ambiguous predictions, in which the predicted probabilities across multiple categories are extremely close, making classification difficult. To improve the classification precision of highly confusing samples during the testing phase, a prediction probability correction approach based on label priors is proposed. First, highly confusing samples are identified using normalized information entropy. Then, the predictive probabilities are corrected according to the label prior distribution.

Identification of high-confusion samples using normalized information entropy. The uncertainty of the predicted probability distribution for each test sample is quantified using normalized information entropy. The degree of confusion increases as the predicted probability distribution $p_x = [p_x{}^1, p_x{}^2, \ldots, p_x{}^c]$ for sample x becomes more uniform. The normalized information entropy is calculated using Equation (6). After computing the normalized information entropy $H(p_x)$ of the predicted probability distribution p_x, a confusion threshold $0 \leq \sigma \leq 1$ is set. Samples with normalized information entropy greater than σ are assigned to the high-confusion sample set s_{high}, while the remaining samples are assigned to the low-confusion sample set s_{low}.

Predictive probability correction guided by the label prior distribution. During this phase, each high-confusion probability vector p_{high} in the high-confusion set s_{high} is updated in accordance with the label prior distribution to enhance overall classification accuracy. To clarify the rationale behind the proposed correction scheme, we provide an illustrative example. Assume that two samples meet the following conditions: (1) their category labels are restricted to either 0 or 1, and (2) the two samples belong to different categories. With the stated constraints in place, if the first sample is classified as 1, the second sample is more likely to be predicted as 0 in order to satisfy the second condition. Assuming the first sample is predicted with adequate reliability, this strategy can yield optimal overall classification performance.

Applying the above theory to the proposed approach, the low-confusion instance set S_{low} is used as a reference to correct each sample point in the high-confusion instance set S_{high}. During correction, the two instances in Figure 2 are more likely to be reassigned to *Telephone* when, within the set S_{low}, the probability mass of the *Messaging* class is closer to its counterpart in the label prior distribution than that of *Telephone*, and the opposite holds otherwise. With suitable parameter choices, this rule can deliver near-optimal overall classification performance. Accordingly, we first count the services associated with each label in the training set and then normalize the counts to derive the label prior distribution $\tilde{p} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_c]$, as defined by Equation (7):

$$\tilde{p}_i = \frac{\sum_{j=1}^{c} C_{\tilde{y}=i,y^*=j}}{n} \tag{7}$$

Here, $C_{\tilde{y}=i}$ represents the number of Web service data points with the initial category $\tilde{y}$ = i, $\tilde{p}_i$ represents the prior probability of category i, and n represents the total number of training data samples. The label prior distribution $\tilde{p}$ serves as a reference for adjusting the probability vector $p_{high} = [p_{high}{}^1, p_{high}{}^2, \ldots, p_{high}{}^c]$ of samples in the set S_{high}. In practice, high-confusion samples are usually semantically related to only a subset of labels, so the correction needs to be applied only to part of the probability mass. Therefore, we compute the mean of p_{high} together with all probability distributions in the set S_{low} using Equation (8):

$$\overline{p}_{high} = \frac{1}{|s_{low}| + 1} \left(p_{high} + \sum_{x \in s_{low}} p_x \right) \tag{8}$$

Here, $\bar{p}_{high}$ represents a probability distribution closer to the label prior $\tilde{p}$, which helps avoid excessive adjustments to the entries of p_{high}. Next, we quantify the deviation between $\bar{p}_{high}$ and $\tilde{p}$ and use it to further refine p_{high} according to Equation (9):

$$p_{high} = p_{high} \times \frac{\tilde{p}}{\bar{p}_{high}} \tag{9}$$

After that, p_{hig} is brought closer to the label prior distribution $\tilde{p}$ and subsequently normalized to yield the final label distribution, as given by Equation (10):

$$p_{high} = \frac{p_{high}}{\sum_{i=1}^{c} p_{high}^{i}} \tag{10}$$

Using the same procedure, we adjust each predicted probability vector in S_{high}, so that the resulting label distribution over the test set better aligns with the prior label distribution $\tilde{p}$. For each test instance, the predicted class is selected as the label with the highest probability in the corrected vector.

4. Experiments

4.1. Experimental dataset and environment

Experiments are conducted on API services registered on the ProgrammableWeb website to evaluate the effectiveness of the proposed approach. Services with extremely brief descriptions, duplicate registrations, and those belonging to categories with insufficient numbers of services are removed from the experimental dataset [28]. To ensure data validity, all selected Web services comply with W3C standards (e.g., standardized service description formats, consistent protocol specifications), serving as the basis for subsequent NCOHA-WSC processing. Services failing to meet these standards are excluded during dataset preprocessing.

Following the category selection strategy commonly adopted in the existing literature, the top 10–50 most populated categories are selected for the experiments. Detailed statistics of the dataset are presented in Table 1, while descriptive information for services in the top 10 categories is provided in Table 2.

All experiments are conducted on a Tesla T4 GPU with 16 GB memory and CUDA version 12.2. The models are implemented using Python 3.8 and PyTorch 2.1. The pre-trained BERT language model is obtained using the Transformers library (version 4.37.2), and the confidence learning algorithm is implemented using the open-source Cleanlab framework (version 2.6.1) [22].

Table 1
Web services datasets

Dataset	Service category	Number of services
DS_1	Top10	6007
DS_2	Top30	11668
DS_3	Top50	14524

Table 2
Information of top 10 Web service category

Service category	Number of services	Service category	Number of services
Financial	1002	Enterprise	493
Tools	838	Social	482
Payments	642	Mapping	61
Messaging	637	Science	430
eCommerce	614	Government	408

4.2. Experimental results and discussion

To demonstrate robustness against noisy and overlapping data, the proposed NCOHA-WSC approach is applied to several state-of-the-art Web service classification models. An overview of these models is provided below:

ServeNet [29]**:** ServeNet is a deep neural network-based service classification model that represents service descriptions using GloVe word embeddings. It employs convolutional neural networks (CNNs) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks to automatically extract functional features from service description embeddings, thereby eliminating the need for manual feature engineering.

ServeNet-BERT [15]**:** ServeNet-BERT is an enhanced version of ServeNet that uses the BERT model to generate contextualized text embeddings from names and Web service descriptions. It employs Bi-LSTM and CNN networks to derive functional representations of services, followed by a fully connected layer for service classification.

BERT(cls) [30]**:** This approach utilizes the cls token of the pre-trained BERT language model for service classification. The representation of the cls token is refined through fine-tuning on the Web service dataset, and service classification is performed using a fully connected feed-forward neural network.

CARL-Net [31]**:** CARL-Net is a deep neural network that integrates keywords, service names, and collaborative attention representations for service classification. Service names and keywords are used to construct data augmentation vectors, while a collaborative attention mechanism is employed to capture the importance of words in both the service description vectors and the augmentation vectors. Finally, these representations are fused to perform service classification.

The model architectures and hyperparameter settings used in our experiments are consistent with the original papers. During the confidence learning phase, the dataset is randomly split into a training set and a validation set with a ratio of 9:1. The training dataset is used to train the service classification model, while the validation dataset is used to obtain the predicted probability distributions. Cross-validation is employed to obtain predicted probability distributions for all training data. During model training, an early stopping strategy is adopted to prevent overfitting by terminating training when classification performance on the validation dataset begins to degrade. Macro-F1 is used as the evaluation metric. The experimental results obtained by applying the proposed approach to different service classification models are presented in Table 3. Here, $\Delta_{F1\text{-}avg}$ denotes the average improvement in Macro-F1 across all service classification models; –NCO denotes that the proposed approach is not applied, whereas +NCO denotes that it is applied. DS_1, DS_2, and DS_3 represent Web service datasets containing 10, 30, and 50 categories, respectively.

From the results, we can draw the following conclusions: First, for all datasets, the Macro-F1 scores of most models are improved after using this approach, indicating that this approach has a positive effect on service classification performance.

Based on the results, several conclusions can be drawn:

1) Across all datasets, the Macro-F1 scores of most models improve after applying the proposed approach, indicating its positive effect on service classification performance.
2) Classification models that rely on global semantic representations, such as BERT(cls) and BERT-DPCNN, exhibit more pronounced improvements after applying the proposed

Table 3
The classification experiment results of NCOHA-WSC

Metric	Model	DS_1		DS_2		DS_3	
		–NCO	*+NCO*	*–NCO*	*+NCO*	*–NCO*	*+NCO*
Macro-F1	ServeNet	0.745	0.752	0.654	0.657	0.599	0.615
	ServeNet-BERT	0.807	0.801	0.755	0.753	0.679	0.683
	BERT(cls)	0.820	0.824	0.759	0.762	0.683	0.685
	BERT-DPCNN	0.819	0.822	0.760	0.767	0.682	0.685
	CARL-Net	0.823	0.820	0.762	0.764	0.684	0.688
$\Delta_{F1\text{-}avg}$		+0.1%		+0.26 %		+0.58%	

approach on datasets with relatively fewer categories, such as DS_1 and DS_2. However, models that incorporate additional data features, such as CARL-Net and ServeNet-BERT, exhibit smaller performance gains or even slight performance degradation. This can be attributed to the fact that classification models relying primarily on the global semantics of service descriptions are more sensitive to abnormal data in datasets with fewer categories. In contrast, incorporating additional data features, such as service names, can partially mitigate the adverse effects of noisy and overlapping data, thereby improving model robustness and stabilizing classification performance.

3) Models that employ CNNs to extract local functional features from service descriptions, such as CARL-Net, BERT-DPCNN, and ServeNet-BERT, exhibit more pronounced performance improvements as the number of categories increases, compared with models that rely primarily on global semantic representations, such as BERT(cls). CNNs are effective at capturing salient textual features from service descriptions. However, in large-scale datasets, the classification performance of these models is constrained by the high similarity of key textual features among overlapping data. Notably, performance gains are limited in two scenarios: first, for models integrating additional data features (e.g., CARL-Net, ServeNet-BERT) on datasets with fewer categories (DS1, DS2), as supplementary features (e.g., service names, keywords) already mitigate noise and overlap impacts, leaving limited room for NCOHA-WSC optimization, and second, when the proportion of noise/overlap data is extremely low, as the core modules of NCOHA-WSC (noise filtering, probability correction) have minimal redundant data to process, leading to marginal performance improvements.

4.3. Ablation analysis

To evaluate the effectiveness of each module in NCOHA-WSC, ablation experiments are conducted under the following settings:

1) enabling only confidence learning (+CL)
2) enabling only label prior correction (+PC)

The results of the ablation experiments are presented in Table 4, where $\Delta_{F1\text{-}avg}$ represents the average improvement in Macro-F1 across all service classification models.

Based on the results, several conclusions can be drawn:

1) The confidence learning (+CL) and prior correction (+PC) modules yield limited performance improvements on datasets with fewer categories, such as DS_1 and DS_2. This is because the adverse effects of noisy and overlapping data on deep learning models are relatively minor in low-category settings. As the number of categories increases, however, each module of NCOHA-WSC becomes more effective in enhancing service classification performance.
2) As the number of categories increases, models that rely on global functional semantics of service descriptions become more sensitive to the adverse effects of noisy data, whereas models that emphasize local functional semantics are more susceptible to the negative impact of overlapping data. For example, on the DS_3 dataset, classification models such as ServeNet-BERT, CARL-Net, and ServeNet, which employ CNNs to extract service functional features, exhibit more pronounced performance improvements when the prior correction (+PC) module is applied, while showing relatively smaller gains when only the confidence learning (+CL) module is used. By contrast, classification models such as BERT(cls) and

Table 4
Ablation experiment results

Metric	Model	DS_1		DS_2		DS_3	
		–CL	*+PC*	*–CL*	*+PC*	*–CL*	*+PC*
Macro-F1	ServeNet	0.742	0.734	0.655	0.657	0.610	0.612
	ServeNet-BERT	0.804	0.817	0.756	0.756	0.680	0.682
	BERT(cls)	0.822	0.825	0.763	0.762	0.687	0.683
	BERT-DPCNN	0.821	0.825	0.760	0.762	0.685	0.683
	CARL-Net	0.821	0.821	0.763	0.765	0.684	0.689
$\Delta_{F1\text{-}avg}$		–0.08%	+0.16%	+0.14%	+0.28%	+0.38%	+0.44%

BERT-DPCNN, which extract service functional features based on global semantic representations of service descriptions, exhibit more substantial performance improvements when the confidence learning (+CL) module is applied and relatively smaller gains when the prior correction (+PC) module is used.

4.4. Hyperparameter experiments

During the confidence learning phase, a confidence threshold β is employed to distinguish between noisy data and overlapping data. During the probability correction phase, a confusion threshold σ is used to differentiate between low-confusion data and high-confusion data. Experiments are conducted on the DS_3 dataset with β ϵ[0, 0.8] for the confidence learning module and σ ϵ[0.5, 1.0] for the probability correction module to investigate the effects of the confidence threshold β and the confusion threshold σ on their respective modules. The experimental results are presented in Figures 4 and 5, respectively.

From Figure 4, several observations can be made:

1) The noise robustness of different models on the DS_3 dataset varies. For example, ServeNet, which is based on GloVe word embeddings, exhibits relatively weaker noise resistance due to the limited quality of its service description text representations. As a result, when the confidence threshold β is set to a large value, overlapping samples are more likely to be misidentified as noisy samples, leading to the removal of classification-relevant information and a consequent degradation in model performance. In contrast, model performance improves only when β is kept at a relatively low level.
2) Classification models that exploit local semantic features of service descriptions, such as ServeNet-BERT and CARL-Net, produce more concentrated predicted probability distributions and correspondingly lower normalized information entropy values. Consequently, these models achieve higher accuracy in identifying overlapping data when β ϵ[0, 0.1]. In contrast, models that rely on global semantic representations, such as BERT(cls) and BERT-DPCNN, tend to produce more uniform predicted probability distributions. As a result, they are more accurate at identifying overlapping samples when β ϵ[0.1, 0.2].

From Figure 5, several observations can be made:

Figure 4
The effect of confidence threshold on confidence learning

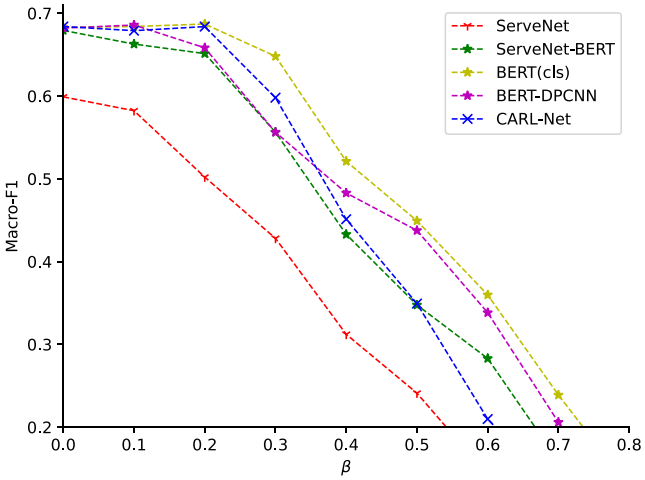
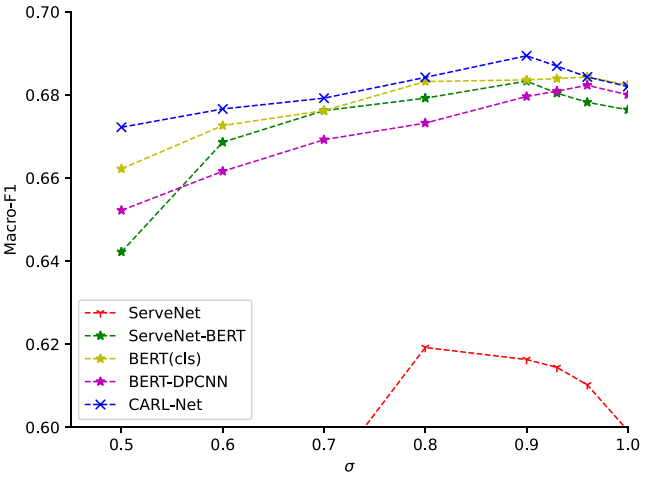


Figure 5
The effect of confusion threshold on prior correction



1) On the DS_3 dataset, ServeNet, which is based on GloVe word embeddings, achieves relatively better performance when the confusion threshold σ is set to approximately 0.8. In contrast, service classification models based on BERT perform better when σ is around 0.9. This difference can be attributed to the limited capability of GloVe-based models to effectively discriminate overlapping samples. Specifically, the probability distributions of most overlapping samples are not uniform, which leads to a smaller proportion of overlapping samples being identified based on normalized information entropy.
2) Models based on global semantic representations, such as BERT(cls) and BERT-DPCNN, exhibit limited accuracy in identifying overlapping samples, resulting in many non-overlapping samples being misclassified as overlapping. Therefore, correction precision can be improved only when the confusion threshold σ is set within a relatively high range, specifically σ ϵ[0.96, 1]. In contrast, service classification models that rely on local semantic representations, such as ServeNet-BERT and CARL-Net, demonstrate higher accuracy in identifying overlapping data. Consequently, the probability correction module is most effective when the confusion threshold σ is set to approximately 0.9.
3) To address the computational overhead and scalability of NCOHA-WSC, a theoretical complexity analysis of its core components is presented. The confidence learning module primarily involves cross-validation prediction, threshold calculation, and joint distribution statistics, with an overall time complexity of O(N × D) (where N denotes the number of samples and D represents the feature dimension), aligned with the linear complexity of base classification models. The entropy calculation module requires traversing the predicted probability distribution of each sample, resulting in a time complexity of O(N × C) (C is the number of categories); given that C is a constant in Web service classification tasks, this complexity can be simplified to O(N). The probability correction module includes label prior statistics and probability adjustment, with a time complexity of O(N) as it only involves linear traversal and vector operations. All core components avoid exponential or polynomial overhead, ensuring that NCOHA-WSC can efficiently scale to large-scale or real-time Web service classification scenarios.

5. Conclusion and Future Works

This paper proposes a Web service classification approach, called NCOHA-WSC, to address the challenges of data noise and class overlap in existing service classification. The approach filters noisy data during the data preprocessing phase using confidence learning and identifies high-confusion samples during the testing phase of the classification model via information entropy. Furthermore, based on the label prior distribution, it corrects the predicted probability distributions of high-confusion samples. Experimental results demonstrate that the proposed approach effectively filters noisy data from the dataset and significantly improves the overall classification performance of existing Web service classification models. Notably, NCOHA-WSC exhibits strong generalizability to other text classification and noisy-label domains. Its model-agnostic design and core modules (confidence learning–based noise filtering, label prior–driven probability correction) rely only on textual features and category labels, enabling direct adaptation to document classification (filtering mislabeled thematic documents), API tagging (resolving conflicting tag assignments), and social media data classification (mitigating noise from subjective annotations and multi-topic overlap). This cross-domain adaptability extends the framework's practical value beyond Web service classification.

NCOHA-WSC currently requires manual hyperparameter tuning across different datasets, which is time-consuming and labor-intensive. Future work will focus on enhancing the proposed approach by enabling self-adaptive hyperparameter optimization across diverse datasets. In addition, in more realistic application scenarios, datasets may be incremental rather than static and may also be extremely large. Therefore, more advanced incremental prediction techniques [32] and data sampling strategies [33, 34] should be explored to ensure scalability and robustness in such environments.

Funding Support

This work was supported by the National Natural Science Foundation of China under Grant 62472264 and 52574256, the Natural Science Distinguished Youth Foundation of Shandong Province under Grant ZR2025QA13, and the national funds through FCT,I.P. (Fundação para a Ciência e a Tecnologia,I.P.), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS (DOI: 10.54499/UID/04152/2025, and by the European Union – NextGenerationEU under the project UID/PRR/04152/2025 (DOI: 10.54499/UID/PRR/04152/2025).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available on GitHub at https://github.com/HIT-ICES/Correted-ProgrammableWeb-dataset.git.

Author Contribution Statement

Feng Zhang: Writing – original draft, Writing – review & editing, Visualization, Project administration, Formal analysis. **Lin Xue:** Writing – review & editing, Software, Validation, Project administration. **Huiling Li:** Writing – original draft, Writing – review & editing, Methodology, Investigation. **Cong Liu:** Writing – review & editing, Conceptualization, Resources, Data curation, Supervision, Funding acquisition.

References

[1] Raj, V., & Sadam, R. (2021). Evaluation of SOA-based web services and microservices architecture using complexity metrics. *SN Computer Science*, *2*(5), 374. https://doi.org/10.1007/s42979-021-00767-6

[2] Wang, Q., Wu, W., Zhao, Y., Zhuang, Y., & Wang, Y. (2021). Combining label-wise attention and adversarial training for tag prediction of web services. In *2021 IEEE International Conference on Web Services*, 358–363. https://doi.org/10.1109/ICWS53863.2021.00054

[3] Li, B., Li, Z., & Yang, Y. (2021). Residual attention graph convolutional network for web services classification. *Neurocomputing*, *440*, 45–57. https://doi.org/10.1016/j.neucom.2021.01.089

[4] Li, F., Wang, B., Wang, P., Jiang, M., & Li, Y. (2023). An imbalanced ensemble learning method based on dual clustering and stage-wise hybrid sampling. *Applied Intelligence*, *53*, 21167–21191. https://doi.org/10.1007/s10489-023-04650-0

[5] Xue, L., & Zhang, F. (2024). LCPCWSC: A web service classification approach based on label confusion and priori correction. *International Journal of Web Information Systems*, *20*(3), 213–228. https://doi.org/10.1108/IJWIS-12-2023-0243

[6] Vuttipittayamongkol, P., Elyan, E., & Petrovski, A. (2021). On the class overlap problem in imbalanced data classification. *Knowledge-Based Systems*, *212*, 106631. https://doi.org/10.1016/j.knosys.2020.106631

[7] Demir, S., & Topcu, B. (2022). Graph-based Turkish text normalization and its impact on noisy text processing. *Engineering Science and Technology, an International Journal*, *35*, 101192. https://doi.org/10.1016/j.jestch.2022.101192

[8] Bonab, M. N., Tanha, J., & Masdari, M. (2024). A semi-supervised learning approach to quality-based web service classification. *IEEE Access*, *12*, 50489–50503. https://doi.org/10.1109/ACCESS.2024.3385341

[9] Zeng, K., & Paik, I. (2024). Web service embedding: Representing the invocation association between services with practical-valued vectors. *Expert Systems with Applications*, *238*, 122196. https://doi.org/10.1016/j.eswa.2023.122196

[10] Park, J., Choi, B., Lee, C., & Han, D. (2024). Graph neural network-based SLO-aware proactive resource autoscaling framework for microservices. *IEEE/ACM Transactions on Networking*, *32*(4), 3331–3346. https://doi.org/10.1109/TNET.2024.3393427

[11] Jia, M., Reiter, A., Lim, S.-N., Artzi, Y., & Cardie, C. (2021). When in doubt: Improving classification performance with alternating normalization. In *Findings of the Association for Computational Linguistics: Proceedings of EMNLP 2021*, 1716–1723. https://doi.org/10.18653/v1/2021.findings-emnlp.148

[12] Xiao, Y., Liu, J. X., Hu, R., Cao, B. Q., & Cao, Y. C. (2021). GAT2VEC-based web services classification method. *Journal of Software*, *32*(12), 3751–3767.

[13] Hu, Q., Shen, J., Wang, K., Du, J., & Du, Y. (2022). A web service clustering method based on topic enhanced Gibbs sampling algorithm for the Dirichlet Multinomial Mixture model and service collaboration graph. *Information Sciences*, *586*, 239–260. https://doi.org/10.1016/j.ins.2021.11.087

[14] Shen, J., Huang, W., & Hu, Q. (2022). PICF-LDA: A topic enhanced LDA with probability incremental correction factor for Web API service clustering. *Journal of Cloud Computing*, *11*(1), 19. https://doi.org/10.1186/s13677-022-00291-9

[15] Yang, Y., Qamar, N., Liu, P., Grolinger, K., Wang, W., Li, Z., & Liao, Z. (2020). ServeNet: A deep neural network for web services classification. In *2020 IEEE International Conference on Web Services*, 168–175. https://doi.org/10.1109/ICWS49710.2020.00029

[16] Pan, G., Chang, Y., Qi, H., & Hu, Q. (2023). Web service category recommendation with feature word semantic enhancement and tag co-occurrence. In *2023 International Conference on Networking and Network Applications*, 686–689. https://doi.org/10.1109/NaNA60121.2023.00118

[17] Zhao, K., Liu, J., Xu, Z., Liu, X., Xue, L., Xie, Z., . . . , & Zhou, Y. (2022). Graph4Web: A relation-aware graph attention network for web service classification. *Journal of Systems and Software*, *190*, 111324. https://doi.org/10.1016/j.jss.2022.111324

[18] Liang, B., Kang, G., Liu, J., Cao, B., & Xiang, J. (2022). Attentional neural factorization machine for web services classification via exploring content and structural semantics. In *2022 International Joint Conference on Neural Networks*, 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892320

[19] Wang, Y. Q., Dong, L. Y., Jiang, X. Q., Ma, X. T., Li, Y. L., & Zhang, H. (2021). KG2Vec: A node2vec-based vectorization model for knowledge graph. *PLOS One*, *16*(3), e0248552. https://doi.org/10.1371/journal.pone.0248552

[20] Ha, J. (2025). DeepWalk-based graph embeddings for miRNA–Disease association prediction using deep neural network. *Biomedicines*, *13*(3), 536. https://doi.org/10.3390/biomedicines13030536

[21] Liang, T., Chen, L., Wu, J., & Bouguettaya, A. (2016). Exploiting heterogeneous information for tag recommendation in API management. In *2016 IEEE International Conference on Web Services*, 436–443. https://doi.org/10.1109/ICWS.2016.63

[22] Northcutt, C. G., Jiang, L., & Chuang, I. L. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, *70*, 1373–1411. https://doi.org/10.1613/jair.1.12125

[23] Ortego, D., Arazo, E., Albert, P., O'Connor, N. E., & McGuinness, K. (2021). Multi-objective interpolation training for robustness to label noise. *arXiv Preprint:2012.04462*.

[24] Vuttipittayamongkol, P., Elyan, E., Petrovski, A., & Jayne, C. (2018). Overlap-based undersampling for improving imbalanced data classification. In *Intelligent Data Engineering and Automated Learning – IDEAL 2018: 19th International Conference*, 689–697. https://doi.org/10.1007/978-3-030-03493-1_72

[25] Dablain, D., Krawczyk, B., & Chawla, N. V. (2023). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9), 6390–6404. https://doi.org/10.1109/TNNLS.2021.3136503

[26] Liu, Y., Li, B., Wang, J., Li, D., & Ma, Y. (2022). Multi-information fusion based few-shot web service classification. *Future Generation Computer Systems*, *130*, 231–240. https://doi.org/10.1016/j.future.2021.12.020

[27] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[28] Liu, M., Tu, Z., Zhu, Y., Xu, X., Wang, Z., & Sheng, Q. Z. (2021). Data correction and evolution analysis of the ProgrammableWeb service ecosystem. *Journal of Systems and Software*, *182*, 111066. https://doi.org/10.1016/j.jss.2021.111066

[29] Yang, Y., Ke, W., Wang, W., & Zhao, Y. (2019). Deep learning for web services classification. In *2019 IEEE International Conference on Web Services*, 440–442. https://doi.org/10.1109/ICWS.2019.00079

[30] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, *Volume 1*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[31] Tang, B., Yan, M., Zhang, N., Xu, L., Zhang, X., & Ren, H. (2021). Co-attentive representation learning for web services classification. *Expert Systems with Applications*, *180*, 115070. https://doi.org/10.1016/j.eswa.2021.115070

[32] Guo, N., Liu, C., Mo, Q., Cao, J., Ouyang, C., Lu, X., & Zeng, Q. (2025). Business process remaining time prediction based on incremental event logs. *IEEE Transactions on Services Computing*, *18*(3), 1308–1320. https://doi.org/10.1109/TSC.2025.3562338

[33] Liu, C., Pei, Y., Cheng, L., Zeng, Q., & Duan, H. (2021). Sampling business process event logs using graph-based ranking model. *Concurrency and Computation: Practice and Experience*, *33*(5), e5974. https://doi.org/10.1002/cpe.5974

[34] Li, H., Liu, C., Du, Q., Zeng, Q., Zhang, J., Theodoropoulo, G., & Cheng, L. (2025). Sampling-based next-event prediction for wind-turbine maintenance processes. *Energies*, *18*(16), 4238. https://doi.org/10.3390/en18164238

How to Cite: Zhang, F., Xue, L., Li, H., & Liu, C. (2026). NCOHA-WSC: Handling Noise and Class Overlap in Web Service Classification. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE62028810