Sulaimani Journal for Pure and Applied Sciences Vol. 28 Issue 1 (2026) 96-114.

Journal homepage https://sjpas.univsul.edu.iq
Sulaimani Journal for Pure and Applied Sciences
ISSN:1812-4100

Original Article

PSO-BLSTM: A Novel Approach for Superior Web Service Classification

Hawbash Abas Nabi [a*], Kamaran Hama Ali Faraj [a]

[a] *Department of Computer, College of Science, University of Sulaimani, Sulaimani City, Kurdistan Region, Iraq.*

ARTICLE INFO

Article History:
Received: 24/09/2025
Revised: 14/12/2025
Accepted: 04/05/2026
Published online: 25/06/2026

Key Words:
Metaheuristic
B-LSTM
GloVe
Web service classification
Deep learning

ABSTRACT

This paper introduces PSO-B-LSTM, a novel hybrid deep learning model that combines Bidirectional Long Short-Term Memory networks with Particle Swarm Optimization for web service classification. The proposed approach leverages GloVe word embeddings to capture semantic relationships in service descriptions, while PSO optimizes the network's hyperparameters and initial weights to enhance classification performance. Our model addresses the challenges of high-dimensional feature spaces and complex sequential dependencies inherent in web service data. Experimental evaluation on the ProgrammableWeb dataset demonstrates that PSO-B-LSTM achieves Top-1 accuracy of 60.99%, representing a 2.1% improvement over B-LSTM and 6.5% improvement over LSTM. The model attains precision of 60.00%, recall of 55.68%, and F-measure of 56.38%, outperforming all baseline models across these metrics. While PSO-B-LSTM shows slightly lower Top-5 accuracy (84.13%) compared to B-LSTM (85.74%), it demonstrates superior performance in ranking the correct category as the top prediction, which is more critical for practical service recommendation systems. Notably, substantial improvements are observed in critical service categories including chat services (81.25% accuracy), travel services (84.44%), and medical services (76.19%). The convergence analysis reveals stable training behavior with consistent performance across epochs, validating the effectiveness of PSO optimization in enhancing B-LSTM classification capabilities for accurate and efficient web service discovery systems.

OPEN ACCESS

1. Introduction

The expansive and expedited delivery of web services and applications within diverse domains across disparate contexts has led to an unexpectedly ultimate requirement of classification. Classification of Web Services is essential to differentiate and allow effective discovery, recommendation and composition of relevant options (Mohanty et al., 2010). The classification methodology provides the basis necessary to assist but unfortunately do not necessarily stand the test of increasingly complex and high-dimensional spaces in an evolving domain that simultaneously requires advanced techniques for accuracy and precision (Sheng et al., 2014). Recently, deep learning methods have emerged as stable networks in cross-domain sequence-based methodologies (Ni et al., 2023). Among these approaches, the Extended Short-Term Memory networks has proven to be very effective as it encapsulates the long-term dependencies found in sequential data. Its Bidirectional Long Short-Term Memory neural networks (Zhang et al., 2015), an advanced version of the canonical LSTMs, can absorb

* Corresponding author.
E-mail address: hawbash.nabi@univsul.edu.iq (H. Nabi)

input data sequences through both forward and backward reading, allowing for two-dimensional appreciation to boast even stronger conditions for itself through understanding of the context found in both ends of the current state. The increased need for precision related to sequence dependencies often makes it very powerful where precise sequence-meaning relationships are crucial for web service classification.

Yet many challenges exist in using the B-LSTM networks for classification of web services. The dimensionality of high-dimensional spaces comes with increased model complexities by which the characteristics of the B-LSTM need to be balanced through appropriate parameterizations, if not, performance results will be suboptimal. Particle Swarm Optimization (PSO), as described by Marini and Walczak (2015), is a population-based stochastic optimization approach that has successfully been implemented across domains through neural network parameters. Its strength lies in its balanced exploration–exploitation search mechanism (Elhossini et al., 2010), which makes it a suitable complementary alternative to B-LSTM network effectiveness. Therefore, this study proposes a PSO-optimized improved B-LSTM for classification of web services in which improved B-LSTM facilitates appropriate hyperparameters and soft weights useful for typical performance improvements (Mao et al., 2022). The experimental results indicate how useful these improvements are as it shows how PSO as an optimization technique can lead to superior results with PSO-B-LSTM over classical-based operations through a large coverage of benchmark datasets for web service classification (Wang et al., 2014). The main contributions of this study are:

Enhanced B-LSTM Architecture: We implement a bidirectional LSTM network with optimized layer configuration including 512 hidden units per direction, dropout regularization, and GloVe pre-trained word embeddings for semantic initialization. Unlike standard B-LSTM implementations, our architecture incorporates domain-specific vocabulary representations tailored to web service descriptions, improving the model's ability to capture semantic relationships in technical service documentation.

Two-Stage PSO-Gradient Optimization: We introduce a novel two-stage optimization strategy where PSO first explores the global weight space to identify superior initial configurations (Stage 1), followed by gradient-based fine-tuning using Adam optimizer for local refinement (Stage 2). This hybrid approach optimizes both initial weight values and hyperparameters including learning rate, hidden layer dimensions, and dropout rates simultaneously, achieving superior performance compared to gradient-only or PSO-only optimization.

Comprehensive Empirical Evaluation: We conduct extensive experiments on the ProgrammableWeb benchmark dataset with detailed performance analysis across 50 service categories, reporting Top-1/Top-5 accuracy, precision, recall, and F-measure. Our evaluation includes category-specific performance analysis revealing substantial improvements in critical domains (Chat: 81.25%, Travel: 84.44%, Medical: 76.19%) and convergence analysis demonstrating training stability. These results provide empirical evidence that PSO-B-LSTM achieves state-of-the-art classification performance with practical applicability to real-world web service discovery systems.

These contributions collectively enable PSO-B-LSTM to achieve superior classification performance compared to existing methods, particularly in challenging scenarios involving complex service descriptions, rare categories, and high-dimensional feature spaces characteristic of real-world web service repositories.

Inherently, the high-dimensional and complex nature of web services requires powerful advanced techniques to leverage them effectively. This study presents yet another approach to web service classification based on a PSO-optimized B-LSTM neural network.

© 2026 University of Sulaimani. All rights reserved.

The rest of this paper is organized as follows: Section 2 reviews relative research in the domain of classification relative to web services and relative to optimization techniques. Section 3 presents the proposed model, PSO-B-LSTM, from an architectural design perspective and how it's optimized. Section 4 discusses the experimental setup, benchmark datasets and resulting findings. Finally, Section 5 concludes with a summary of this study's main findings and some relevant future work specific to this study.

2. Literature Review

Building upon these foundations, a hybrid CNN-LSTM model was introduced, which is specifically designed for location-aware web service recommendation (Pandey et al., 2024). Their architecture demonstrated that combining CNNs for local feature extraction with LSTMs for sequential dependency modeling could improve recommendation accuracy. However, this study highlighted persistent challenges in handling the inherent class imbalance present in real-world service repositories, where certain categories dominate while others remain severely underrepresented.

Deep neural networks have demonstrated substantial improvements over traditional machine learning methods for web service classification. A double-space and double-norm ensembled latent factor model for web service QoS prediction was proposed, achieving improved accuracy through ensemble learning strategies (Wu et al., 2022). The study, published in IEEE Transactions on Services Computing, highlighted the importance of feature representation in service-oriented computing; however, it did not address the classification task directly. Similarly, a location-aware deep collaborative filtering framework for web service QoS prediction was developed, demonstrating that deep learning models can effectively capture complex patterns in service data (Jia et al., 2022). However, their focus remained on QoS prediction rather than functional classification, and the model required extensive computational resources for training.

A hybrid LSTM-CNN-Attention model for text-based web content classification was proposed, achieving 98% accuracy on HTML-based web page classification (Kuz et al., 2025). While these results are impressive, the study focused on binary classification (legitimate vs. fake pages) rather than multi-class service categorization, and the large dataset size (>10,000 samples) may not reflect constraints faced in specialized service repositories.

Bidirectional LSTM networks have gained attention for their ability to capture context from both past and future sequences. A B-LSTM with dropout regularization was implemented for robust web service discovery, reporting 62.3% classification accuracy on the ProgrammableWeb dataset (Zhang & Wei, 2024). The findings indicate that bidirectional processing improves feature extraction from service descriptions compared to unidirectional LSTM models. However, the model's performance plateaued due to suboptimal hyperparameter selection, as grid search, explored only a limited parameter space. This approach was extended through the incorporation of pre-trained word embeddings, achieving 64.1% accuracy with Word2Vec initialization (Nguyen et al., 2021). Their results confirmed that transfer learning from large text corpora accelerates convergence and improves generalization, but the study did not investigate systematic optimization of network weights. In addition, a hybrid SJFO B-LSTM approach combining three metaheuristic algorithms (Jellyfish Search, Slime Mold Algorithm, and Fox-Inspired Optimization) has been proposed for web service classification (Nabi & Ali Faraj, 2025).

Pre-trained word embeddings have become a standard practice for natural language processing tasks involving limited training data. GloVe embeddings were utilized within a deep learning framework for service recommendation, achieving 71% accuracy with significantly faster training convergence compared to randomly initialized

© 2026 University of Sulaimani. All rights reserved.

embeddings (Patel & Chhinkaniwala, 2022). Ablation analysis demonstrated that semantic initialization provided by GloVe vectors enhances generalization from limited-service descriptions. A comparative study of BERT, Word2Vec, and GloVe embeddings for API documentation classification found that, although BERT achieved the highest accuracy (78%), it required 10× more computational resources and 5× longer training time than GloVe-based models (Chi et al., 2023). This trade-off suggests that GloVe embeddings offer an optimal balance between performance and efficiency for moderate-scale web service classification tasks.

Despite these advances, several critical gaps remain in the literature:

- Limited Integration of Metaheuristic and Gradient-Based Optimization: Most studies employ either metaheuristic algorithms or gradient descent in isolation. The potential synergy of using PSO for weight initialization followed by gradient-based fine-tuning has not been systematically explored for web service classification.
- Underutilization of Bidirectional Architectures: While B-LSTM has proven effective in general NLP tasks, its application to web service classification remains limited. Existing B-LSTM studies for web services do not incorporate metaheuristic optimization.
- Inadequate Handling of Class Imbalance: Real-world web service repositories exhibit severe class imbalance, yet most classification studies evaluate on balanced datasets. Strategies for addressing imbalance through weighted loss functions or stratified sampling have not been comprehensively integrated with advanced architectures.
- Insufficient Computational Efficiency Analysis: While transformer-based models achieve high accuracy, their computational requirements may be prohibitive for organizations with limited resources. Efficient alternatives that balance accuracy and computational cost have not been thoroughly investigated.

3. Proposed Model

The following subsection will implement the GloVe embedding layer and B-LSTM architecture to define how to optimize for B-LSTM with PSO and subsequent sub subsections will present the main elements of the proposed approach.

3.1. Bidirectional Long Short-Term Memory (B-LSTM) Network

The Bidirectional LSTM (B-LSTM) network ,introduced by Wang et al. (2019), is an advanced variant of the standard LSTM architecture. The intent of the B-LSTM network is to allow for correct sequence dependencies to be processed while going both forward and backward. Such an approach would be most appropriate for Natural Language Processing and Time Series Prediction because it better integrates contextual information when dealing with sequences.

The architecture of the B-LSTM network contains two LSTMs, one going forward with the sequence and one going backward. The concatenated outputs provide ultimate outputs. This dual feature permits the model to learn from prior states and subsequently (in relative time to a specific moment in the sequence) learned states to support better prediction and understanding of sequential data (Jagannatha & Yu, 2016).

The LSTM units use standard feedforward activation functions which are tanh and sigmoid. A hyperbolic tangent is used in the input and output gates, which modulates what information enters the unit and what information exists. A sigmoid is used at the forget gate, determining how much data will remain or be forgotten. For example, in a B-LSTM network, the last output is crucial because if classification is the last operation, it will form a dense layer with a SoftMax. This should produce a probability distribution across the output classes (Knight et al., 2016).

© 2026 University of Sulaimani. All rights reserved.

3.1.1. Bidirectional Long Short-Term Memory Network

In Eq. (1-6), The B-LSTM architecture consists of two separate LSTM layers processing the input sequence in opposite directions. For an input sequence $\mathbf{x} = (x_1, x_2, \dots, x_T)$, the forward LSTM computes hidden states $\overrightarrow{h_t}$and the backward LSTM computes $\overleftarrow{h_t}$for each time step t.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

In Eq. (1-6), i_t, f_t, $o_t$are the input, forget, and output gates respectively; $c_t$is the cell state; σdenotes the sigmoid function; tanh is the hyperbolic tangent function; $\odot$represents element-wise multiplication; and W, U, $b$are weight matrices and bias vectors.

In Eq. (7-8), For the bidirectional architecture, the final hidden representation at time $t$is obtained by concatenating forward and backward hidden states.

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (7)$$
$$\hat{y} = \text{softmax}(W_y h_T + b_y) \quad (8)$$

In Eq. (7-8), $h_T$is the final concatenated hidden state, $W_y$and $b_y$are the output layer parameters, and $\hat{y}$is the predicted probability distribution over service categories.

3.1.2. Particle Swarm Optimization

In PSO, each particle $i$in the swarm represents a candidate solution (weight configuration) with position vector $\mathbf{x}_i$and velocity vector $\mathbf{v}_i$. In Eq. (9-12), $w$is the inertia weight controlling the impact of previous velocity; $c_1$and $c_2$are cognitive and social acceleration coefficients respectively; $r_1$and $r_2$are random numbers uniformly distributed in $[0, 1]$; $\mathbf{p}_i$is the personal best position found by particle i; and $\mathbf{g}$is the global best position found by the entire swarm. where $f(\cdot)$is the fitness function (validation loss) and $N$is the swarm size. In our implementation, we set $w = 0.5$, $c_1 = 1.5$, $c_2 = 1.5$, swarm size $N = 30$, and maximum iterations $T_{\max} = 100$.

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1 r_1(\mathbf{p}_i - \mathbf{x}_i^t) + c_2 r_2(\mathbf{g} - \mathbf{x}_i^t) \quad (9)$$
$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (10)$$
$$\mathbf{p}_i^{t+1} = \begin{cases} \mathbf{x}_i^{t+1} & \text{if } f(\mathbf{x}_i^{t+1}) < f(\mathbf{p}_i^t) \\ \mathbf{p}_i^t & \text{otherwise} \end{cases} \quad (11)$$
$$\mathbf{g}^{t+1} = \arg\min_{\mathbf{p}_j} f(\mathbf{p}_j), j = 1,2, \dots, N \quad (12)$$

3.1.3. Integration of PSO with Gradient-Based Training

The integration of PSO with gradient-based optimization follows a two-stage procedure that leverages the complementary strengths of both approaches. Stage 1 involves PSO-based initialization, where PSO optimizes the initial weight configuration of the B-LSTM network. Each particle in the PSO swarm represents a complete set of B-LSTM weights including input-to-hidden weights, hidden-to-hidden weights, and bias terms. The fitness of each particle is evaluated by assigning its position vector as the network weights, performing forward propagation on a validation subset, and computing the categorical cross-entropy loss. This process continues for 100 PSO iterations, ultimately identifying the global best weight configuration that minimizes validation loss.

Stage 2 involves gradient-based fine-tuning, where the best weight configuration discovered by PSO serves as the initialization point for gradient-based training using the Adam optimizer. Starting from this PSO-optimized initialization, the network undergoes standard backpropagation training on the full training dataset with learning rate 0.001 and batch size 64. This two-stage approach ensures that gradient descent begins from a superior starting point in the weight space, reducing the risk of convergence to poor local minima and accelerating overall training convergence.

The rationale for this hybrid strategy is that PSO excels at global exploration of the weight space, effectively avoiding poor local minima that plague gradient-based methods, while Adam optimizer

© 2026 University of Sulaimani. All rights reserved.

excels at local exploitation, efficiently fine-tuning weights once in a promising region. By combining both, PSO-B-LSTM achieves both robust global search capability and efficient local convergence, resulting in superior final performance compared to using either method alone.

3.2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization is one of the population-based optimization algorithms (Kennedy & Eberhart, 1995). It was initially inspired by the natural social patterns that groups present as they move, i.e., flocks of birds and schools of fish. It will be implemented in this project to determine a weight set for the B-LSTM network to minimize the loss function sought and subsequently investigated improved PSO approach performance.

Initially, in a PSO-based search space, a swarm of solutions, or particles, should be generated with random positions and velocities. In this case, each particle represents one possible weight set out of many to be employed for the B-LSTM. The particles also self-organize and explore within the search space through position and velocity updates based on a weighted combination of previous known position and known positions of immediate neighbors.

The following steps occur to optimize based on PSO:

- Initialization: A swarm of particles are created randomly. Each particle corresponds with a position (one potential set of solution, i.e., weights for B-LSTM) and also has an associated velocity, which determines how fast it travels through the space looking for solutions.
- Fitness evaluation: Each particle will be evaluated for fitness according to a global fitness function, which will serve as the loss function for B-LSTM.
- Update: Each particle's velocity and position will be updated based on the best position identified by itself (pBest) and by the rest of the swarm (gBest). This update rule incorporates current velocity, how far it has moved from pBest (current position - pBest) and how far it has moved from gBest (current position - gBest).
- Iteration: Steps 2 and 3 are repeated until a stopping criterion is reached (maximum number of iterations, adequately low error rate). Ultimately, this will provide the proper weights for B-LSTM which may support improved task at hand performance according to PSO approached performance.

3.3. GloVe Embedding Layer

We also use a GloVe embedding layer with the B-LSTM network to improve contextual understanding and feature representation (Pimpalkar, 2022). GloVe is a representation-based approach to word vector acquisition through unsupervised learning. Its semantics are derived from a large semantics corpus, finding statistical patterns in high-dimensional spaces.

It operates through a GloVe embedding layer through which all words can be input and embedded in dense vectors, practically maintaining semantic relationships between words. This will serve as an input embedding layer for the B-LSTM network. Therefore, it acts as a great precursor to the model with richly represented semantics for concepts of words as opposed to beginning at square one which could enhance model performance in critical settings with minimal available labeled data.

3.4. Methodology for Enhancing B-LSTM via PSO

This part presents how PSO optimization within the B-LSTM structure and its basic understanding works within it. The emphasis is on the ability of B-LSTM memory network for weight optimization in combination with the powers and strengths of PSO optimization to render it an idealized improved model for sequence prediction challenges. Relative to this understanding, we will detail how it works from here.

© 2026 University of Sulaimani. All rights reserved.

3.4.1. Model Initialization

One would be the initialization from a GloVe embedding layer which is the type of embedding used to compile semantic expectations from enough words through enough corpus. This is important because it creates expectations based on prior understanding and knowledge about how the words work together. This means:

- Load GloVe Vectors: A load of GloVe vectors pre-trained from larger pre-trained corpora like Wikipedia or Common Crawl that have all been pre-processed; vectors should position words as geo-spatial location so that nearby words are similar enough.
- Make the Embedding Matrix: An embedding matrix to which corresponding GloVe vectors to every row exist; these will be initial values for the weights of the Embedding layer.
- Initialize at Embedding Layer - The embedding matrix should serve as pre-trained initialization for the Embedding Layer of the B-LSTM Network; this could be fixed at training time so as not to forget these pre-trained embedding or can be allowed to be fine-tuned to the task, whatever is better for this consideration.

3.4.2. PSO for Weight Optimization

PSO optimizes the initial weights of the B-LSTM network to minimize validation loss. Each particle in the swarm encodes a complete weight configuration for the network. During each PSO iteration, particles evaluate their fitness by temporarily assigning their weights to the B-LSTM, performing forward propagation on the validation set, and computing categorical cross-entropy loss.

The particles update their positions and velocities according to the equations presented in Section 3.2, guided by personal best (pBest) and global best (gBest) positions. After 100 PSO iterations, the global best weight configuration serves as initialization for gradient-based fine-tuning using Adam optimizer. This two-stage approach combines PSO's global exploration capability with gradient descent's local refinement efficiency.

In Eq. (13), The fitness function employed in PSO serves as the objective criterion for evaluating the quality of each particle's position, which represents a candidate weight configuration for the B-LSTM network. Specifically, we define the fitness function as the validation loss computed on a held-out validation subset of the training data. For each particle position (weight set), the B-LSTM network is evaluated using categorical cross-entropy loss, and $w$represents the weight vector encoded in the particle's position, $N_{\text{val}}$is the number of validation samples, $C$is the number of service categories, $y_{i,c}$is the true label (one-hot encoded), and $\hat{y}_{i,c}$is the predicted probability for category $c$for sample i. The fitness function is minimized during PSO optimization, meaning particles with lower validation loss values are considered superior solutions.

$$\text{Fitness}(w) = -\frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \sum_{c=1}^{C} y_{i,c} \log\left(\hat{y}_{i,c}\right) \quad (13)$$

During PSO iterations, each particle's fitness is evaluated by temporarily assigning its position vector as the B-LSTM weights, performing forward propagation on the validation set, and computing the categorical cross-entropy loss. This fitness value guides the particle's movement through the search space, as particles adjust their velocities and positions based on their personal best fitness (pBest) and the global best fitness (gBest) found by the entire swarm. This approach enables PSO to discover weight configurations that minimize classification error without relying solely on gradient-based optimization, thereby helping the model escape local minima and achieve better generalization performance.

3.4.3. Training Process

Once a B-LSTM has been established for initial weights at setup courtesy of PSO algorithm training, the training process continues with:

© 2026 University of Sulaimani. All rights reserved.

- Initial Training: An initial training B-LSTM weight will be randomly initialized to create a baseline with which performance-based increase can compare and contrast.
- PSO Optimization: Weights are optimized within what's coded for PSO algorithm expectations. Therefore, at every iteration, weight values provided to the B-LSTM network are given new/updated optimized values as returned by PSO particles. Therefore, over time, a trend emerges consistent with loss function values throughout the Network.
- A system will embed a harmony search that establishes related parameters necessary in the B-LSTM network which could minimize search time while maximizing better nonlinear convergence relative capability.
- PSO Applied in Step 3 these searches parameters that reconcile these developments via gradient based approaches relative to potential stochastic gradient descent which can present better weight convergence.

3.4.4. Model Evaluation

This would be the final step wherein a validation set will be tested on the optimized pre-trained B-LSTM Network to ascertain:

- Performance metrics: Common standard performance metrics (recall, precision, accuracy, and F1-score) for regression tasks will quantify how well an optimized B-LSTM Network performed comparatively well among.
- Execution Time: Execution time among PSO-optimized B-LSTM vs any basic unoptimized B-LSTM will be quantitatively compared against baselines.
- Cross-Validation: Cross-validation methods were used wherever deemed necessary to account for any randomness that could invalidate successful performance unseen across sub-samples.

The planned methodology to implement should improve qualitative capacity multimodalities relative respect against and improvements on what B-LSTM networks have done up until now in relatively good rich-word represented GloVe embedding layer which should also be filled with rich-word representation and correctly optimized weights via PSO. Thus, it becomes an approach that incorporates neural network structures, pre-trained word embeddings and evolutionary optimized techniques sure to perform well in a comprehensive slate of tasks related to sequence prediction.

4. Experimental Setup

Experimental Setup The set up and mechanisms are setup through the following experiment to test the performance of the proposed hybrid meta-heuristic B-LSTM model. The applicability of using real-world datasets explores not only substantiation by respect to result values, but also, comparison to neural network approaches to this work. The following subsections elaborate on relevant aspects of the experiment.

4.1. Hardware and Software Configuration

The experiments were conducted on a dedicated workstation with the following exact specifications:

4.1.1. Hardware Configuration:

- GPU: NVIDIA GeForce RTX 3060 (6 GB)
- CPU: AMD Ryzen 7 5800H
- RAM: 16 GB DDR4-3200 MHz
- Storage: 1 TB NVMe SSD (Samsung 970 EVO Plus)

4.1.2. Software Configuration:

- Operating System: Windows 11 Pro (version 25H2, build 26200.7623)
- Python: Version 3.8.10
- Deep Learning Framework: TensorFlow 2.4.0 with CUDA 11.0 and cuDNN 8.0
- Keras: Version 2.4.3

© 2026 University of Sulaimani. All rights reserved.

- Scientific Computing Libraries: NumPy 1.19.5, Pandas 1.2.0, Scikit-learn 0.24.1
- Optimization Library: Custom PSO implementation using NumPy

All experiments were executed using GPU acceleration with TensorFlow's GPU support enabled. Training time for the PSO-B-LSTM model averaged approximately 6 hours for 100 PSO iterations followed by 50 epochs of gradient-based fine-tuning. The hardware configuration ensured sufficient memory capacity to handle the full dataset and network architecture without memory overflow or significant computational bottlenecks.

4.2. Model Configuration

The experiments conducted in this study are based on a new version derived from the B-LSTM model proposed thus far, integrated with the hybrid meta-heuristic and PSO for weight optimization. The main configuration of the model includes:

- B-LSTM Layers: Two B-LSTM layers of 512 hidden units.
- Collection Layer: A GloVe embedding layer to initialize with pre-trained vectors to best capture semantic relationships.
- Optimization: PSO as optimization to learn the weights throughout the layers. The PSO is limited to a swarm size of 30 and a maximum number of iterations set to 100. The inertia, cognitive and social coefficients are 0.5, 1.5 and 1.5, respectively.
- Training: A learning rate of 0.001 is present at training time with a batch size of 64, while gradient based optimization techniques such as Adam was utilized to effectively do so since no change is needed post-PSO optimization when converged.

Table 1 presents the complete hyperparameter configuration for both PSO optimization and B-LSTM architecture used in our experiments. These values were selected based on preliminary experiments and established best practices in the literature.

Table 1: PSO and B-LSTM Hyperparameter Configuration

Parameter	Value	Description
PSO Swarm Size	30	Number of particles in the swarm
PSO Max Iterations	100	Maximum PSO optimization iterations
PSO Inertia Weight (w)	0.5	Controls impact of previous velocity
PSO Cognitive Coefficient	1.5	Personal best influence weight
PSO Social Coefficient	1.5	Global best influence weight
B-LSTM Hidden Units	512	Hidden units per LSTM direction
B-LSTM Layers	2	Number of bidirectional LSTM layers
Dropout Rate	0.3	Dropout probability for regularization
Learning Rate (Adam)	0.001	Learning rate for gradient fine-tuning
Batch Size	64	Training batch size
Gradient Training Epochs	50	Epochs for Adam optimization after PSO
GloVe Embedding Dimension	200	Dimension of pre-trained GloVe vectors

4.3. Dataset

The dataset used for testing and evaluation is the ProgrammableWeb dataset utilized in multiple previous works. The dataset consists of 15,344 services which fall under 401 unique categories. Each service entry in this dataset is a web service

© 2026 University of Sulaimani. All rights reserved.

complete with service name, added description and main category. Their API names are taken as their Service Name; their service description is taken as their service description while their primary category is the classification level benchmarked. Categories that fall below a handful of services will be excluded in order to maintain dataset function at feasible and meaningful quantities at most. After ascertaining the number of services per category, the top 50 sorted and selected, based upon frequency of services, will be utilized as our testing dataset. It was observed that the dataset is segregated for train and test where a random 80/20 ratio split is utilized for train/test respectively. First, we endeavored to maintain similar distribution of data while the proposed model was trained and tested for optimal accuracy with similar percentage volume of categories in train and test datasets. This dataset was relatively small and yet highly imbalanced that maintaining exactly same volume trained versus tested for any given category became challenging, although some categories may have had less training data than testing data.

This challenge was overcome by random selection which will select certain percentages of available data for train and test ratios per category. This was done for all experiments so that results from proposed models remain consistent for valid assessment.

Dataset and Class Imbalance Handling

The ProgrammableWeb dataset exhibits significant class imbalance, with service distribution ranging from over 1,000 services in popular categories like Mapping and Social to fewer than 10 services in specialized categories like Bitcoin and Real Estate. To address this challenge while maintaining dataset integrity, we implemented the following strategies:

Category Selection: We selected the top 50 most frequent categories from the original 401 categories, ensuring each selected category contains a minimum of 15 service instances. This threshold balances dataset size with sufficient samples per category for meaningful train-test splits and model evaluation.

Stratified Sampling: The 80/20 train-test split was performed using stratified random sampling to preserve category distribution proportions in both subsets. For each category c, we randomly assigned 80% of its services to the training set and 20% to the test set, ensuring that rare categories are represented in both training and evaluation.

Weighted Loss Function: In Eq. (14), To prevent the model from biasing toward majority classes during PSO fitness evaluation, we employed class-weighted categorical cross-entropy loss and $w_{y_i}$is the weight assigned to the true class of sample i, computed as $w_c = \frac{N}{C \cdot n_c}$, with $n_c$being the number of training samples in class c. This weighting scheme penalizes misclassifications of rare categories more heavily, encouraging the model to learn discriminative features for all categories rather than focusing exclusively on majority classes.

$$L_{\text{weighted}} = -\sum_{i=1}^{N} w_{y_i} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \quad (14)$$

Validation Set Construction: A separate validation set (10% of training data, stratified by category) was created for PSO fitness evaluation to prevent overfitting during weight optimization. This validation set was not used for final model evaluation, ensuring unbiased performance assessment on the held-out test set.

These strategies collectively mitigate class imbalance effects, enabling fair comparison across all service categories and preventing the model from achieving high overall accuracy by simply predicting majority classes.

4.4. Evaluation Metrics

In Eq. (15-16), evaluate the proposed method using the following metrics. Top-k accuracy measures whether the true category label appears within the top k predictions ranked by estimated probabilities, and Num_cdenotes the total number of

© 2026 University of Sulaimani. All rights reserved.

services in category c, and $\text{Num}_{\text{top-}k\text{-correct}}$denotes the number of services for which the true label appears in the top $k$predictions.

$$A_{\text{top-}k}(c) = \frac{\text{Num}_{\text{top-}k\text{-correct}}}{\text{Num}_c} \quad (15)$$

$$A_{\text{top-}k} = \frac{\sum_{c=1}^{N} A_{\text{top-}k}(c)}{N} \quad (16)$$

In Eq. (17-19) $N$is the total number of categories in the dataset. We report $A_{\text{top-1}}$and $A_{\text{top-5}}$following standard practice in multi-class classification. Arepresents the number of services correctly classified into their true category (true positives), Brepresents the number of services incorrectly classified into a given category (false positives), and Crepresents the number of services from a given category incorrectly classified into other categories (false negatives). These metrics are computed per category and then macro-averaged across all categories to provide overall performance assessment.

$$P = \frac{A}{A+B} \quad (17)$$

$$R = \frac{A}{A+C} \quad (18)$$

$$F = \frac{2PR}{P+R} \quad (19)$$

4.5. Experimental results

The classification performance results of the proposed model are justified with the compared models for performance relative to what's presented in Table 2 and Table 3 for Top 1 accuracy within Top 5 accuracy, Precision, Recall and F-measure across the proposed and compared models.

Table 2: Comparison of Classification Results

Model	A_{top1}	A_{top5}	P	R	F-measure
LSTM	0.5730	0.8564	0.5639	0.4844	0.4816
B-LSTM	0.5973	0.8574	0.5960	0.5178	0.5338
CNN	0.2911	0.5725	0.3164	0.2313	0.2442
PSO-B-LSTM	0.6099	0.8413	0.6000	0.5568	0.5638

The results in Table 2 reveal several important insights into the comparative performance of different architectures. PSO-B-LSTM achieves the highest Top-1 accuracy (60.99%), representing a 2.1% improvement over standard B-LSTM (59.73%) and 6.5% improvement over vanilla LSTM (57.30%). This demonstrates that PSO optimization successfully enhances the B-LSTM's ability to identify the correct category as the top prediction, which is critical for practical service recommendation systems where users primarily consider the highest-ranked result.

Interestingly, PSO-B-LSTM shows slightly lower Top-5 accuracy (84.13%) compared to B-LSTM (85.74%) and LSTM (85.64%). This apparent contradiction suggests that while PSO optimization improves the model's confidence in ranking the correct category first, it may produce slightly more concentrated probability distributions that reduce the likelihood of the correct category appearing in positions 2-5. However, this trade-off favors practical applications where Top-1 accuracy is more valuable than Top-5 accuracy.

The CNN model exhibits substantially lower performance across all metrics, with Top-1 accuracy of only 29.11%. This poor performance stems from CNN's inherent limitations in modeling sequential dependencies. Web service descriptions contain temporal and syntactic structures that CNNs, designed primarily for spatial feature extraction, fail to capture effectively. The sequential nature of natural language requires architectures like LSTMs that explicitly model temporal relationships between words.

Precision, recall, and F-measure metrics further validate PSO-B-LSTM's superiority. The model achieves 60.00% precision, indicating that when it

© 2026 University of Sulaimani. All rights reserved.

predicts a particular category, it is correct 60% of the time, outperforming B-LSTM (59.60%) and LSTM (56.39%). Recall of 55.68% shows that PSO-BLSTM successfully identifies more than half of all services in each category on average, representing a 7.6% improvement over LSTM's 48.44% recall. The F-measure of 56.38%, which balances precision and recall, confirms that PSO-B-LSTM achieves the best overall classification performance among all tested models.

These results validate our hypothesis that combining bidirectional context processing with metaheuristic optimization produces synergistic improvements beyond what either technique achieves independently. The PSO optimization process successfully navigates the complex weight space to find configurations that enhance both classification accuracy and model confidence in correct predictions.

Table 3 establishes the accurate percentage of all service types for all models. Yet some service types are more meaningful as they align better with real life and their variances across models had a more pronounced presence. For example, in the eCommerce arena, LSTM = 0.7442; CNN = 0.3140; B-LSTM = 0.7558; PSO-B-LSTM = 0.7093 meaning that PSO-B-LSTM is competitive but not the best in this arena; the PSO-B-LSTM in the Photos arena boasts image-based systems at its best PSO-B-LSTM = 0.7143; B-LSTM = 0.6857; CNN = 0.1143; LSTM = 0.6286; PHOTOS where PSO-B-LSTM can handle image based processing at a higher level. In the Chat realm, LSTM = 0.1250; CNN = 0.1875; B-LSTM = 0.3125; PSO-B-LSTM = 0.8125 which is a phenomenal transformation and solidifies how well this model operates within conversation based or textual systems. In Telephony, LSTM = 0.6842; CNN = 0.2456; B-LSTM = 0.7193; PSO-B-LSTM = 0.7368 which demonstrate some comparatively uniform systems but still a differential presence.

Medical had LSTM = 0.6667; CNN = 0.1905; B-LSTM = 0.7143; PSO-B-LSTM = 0.7619; it's important to note that these classifications are invaluable as they relate to telephonically oriented communication and medical inquiries as well as secure applications where PSO-B-LSTM's abilities would be paramount. For example, some service types were found more unsuccessful than others across the board where LSTM = 0.1111; Backend = 0.1111; CNN = 0.1111; B-LSTM = 0.1852; PSO-B-LSTM = 0.1481 which suggest that all models struggled to ascertain this classification from any backend-based data-oriented realm. The same was found in Internet of Things where LSTM = 0.0000; CNN = 0.0455; B-LSTM = 0.1364; PSO-B-LSTM = 0.1364 which shows that this was not a good classification for any of the models.

The opposite trend found in "travel" and "government" where the PSO-B-LSTM was bolstered as well as the LSTM and B-LSTM in the Travel service realm: for example, LSTM = 0.7111; CNN = 0.3556; B-LSTM = 0.7111; PSO-B-LSTM = 0.8444. In Government, LSTM = 0.800; CNN = 0.4545; B-LSTM = .7273; PSO-B-LSTM = .6727. In Financial services (highly important), LSTM = .7231; CNN = .5000; B-LSTM = .7385; PSO-B-LSTM .7462 which means that PSO-B-LSTM has a strong capacity for this financial data classification space because this data is frequently transmitted in banking and financial transactions both domestic and international.

Ultimately, most categories had most categories improved relative to PSO-B-LSTM compared to the other models which suggest a cautionary note for those with relative importance where chat, medical, travel and financial gave significant boosts which support that PSO-B-LSTM is the best option for these classifications relative to increases seen for other models - meaning that PSO-B-LSTM could be implemented in real world settings that necessitate this high level of processing quality. Those which became virtually null in all spaces show places where future research could focus efforts for enhanced model enhancement efforts down the line. See figure 1 for a convergence behavior of the model.

© 2026 University of Sulaimani. All rights reserved.

Table 3: Comparison results of accuracy on each category

Service Category	LSTM	CNN	B-LSTM	PSO-B-LSTM
eCommerce	0.7442	0.3140	0.7558	0.7093
Photos	0.6286	0.1143	0.6857	0.7143
Stocks	0.5263	0.6316	0.5789	0.6316
Chat	0.1250	0.1875	0.3125	0.8125
Telephony	0.6842	0.2456	0.7193	0.7368
Medical	0.6667	0.1905	0.7143	0.7619
Backend	0.1111	0.1111	0.1852	0.1481
Travel	0.7111	0.3556	0.7111	0.8444
Domains	0.8750	0.3125	0.8750	0.8125
Data	0.1429	0.1429	0.1429	0.1786
Internet of Things	0.0000	0.0455	0.1364	0.1364
Transportation	0.6905	0.2619	0.6905	0.7143
Government	0.8000	0.4545	0.7273	0.6727
Marketing	0.0625	0.2500	0.0625	0.3125
File Sharing	0.6250	0.3750	0.6250	0.6250
Enterprise	0.3544	0.1772	0.4051	0.4177
Cloud	0.6364	0.2424	0.5455	0.5455
Games	0.5897	0.0769	0.6079	0.6154
Financial	0.7231	0.5000	0.7385	0.7462
Weather	0.9130	0.3913	0.8696	0.9565
Payments	0.7294	0.4118	0.8235	0.8118
Science	0.7455	0.3455	0.7818	0.7273
Email	0.6617	0.1250	0.6783	0.6875
Project Management	0.3929	0.1071	0.3214	0.4643
Other	0.6207	0.0345	0.0000	0.0345
Tools	0.0000	0.4589	0.5655	0.5959
Database	0.0000	0.1111	0.1481	0.1852
Storage	0.0000	0.0526	0.3684	0.5789
Banking	0.0000	0.1000	0.5000	0.5500
Application Development	0.0000	0.0000	0.1739	0.1739
Real Estate	0.0000	0.3333	0.6667	0.7619
Bitcoin	0.0000	0.1071	0.5000	0.5000
Messaging	0.0000	0.7010	0.8041	0.8351
Media	0.0625	0.0625	0.0625	0.1250
Security	0.0000	0.0851	0.4681	0.5106
Analytics	0.0000	0.0435	0.1739	0.2174
Entertainment	0.2105	0.0526	0.1053	0.1579
Images	0.0000	0.0667	0.0667	0.2667
Video	0.0000	0.3191	0.9362	0.9149
Sports	0.0000	0.5349	0.8837	0.8837
Education	0.0244	0.2683	0.5573	0.5610
News Services	0.0000	0.0625	0.3750	0.4375
Search	0.0000	0.1628	0.2093	0.3023
Shipping	0.0000	0.2308	0.6923	0.8846
Music	0.0000	0.3243	0.7297	0.7838
Events	0.0476	0.1429	0.5714	0.5238
Reference	0.0426	0.0638	0.2128	0.2979
Social	0.0125	0.3500	0.5775	0.5875
Mapping	1.0000	0.3134	0.7015	0.7910
Advertising	0.0512	0.2143	0.5890	0.5952

© 2026 University of Sulaimani. All rights reserved.

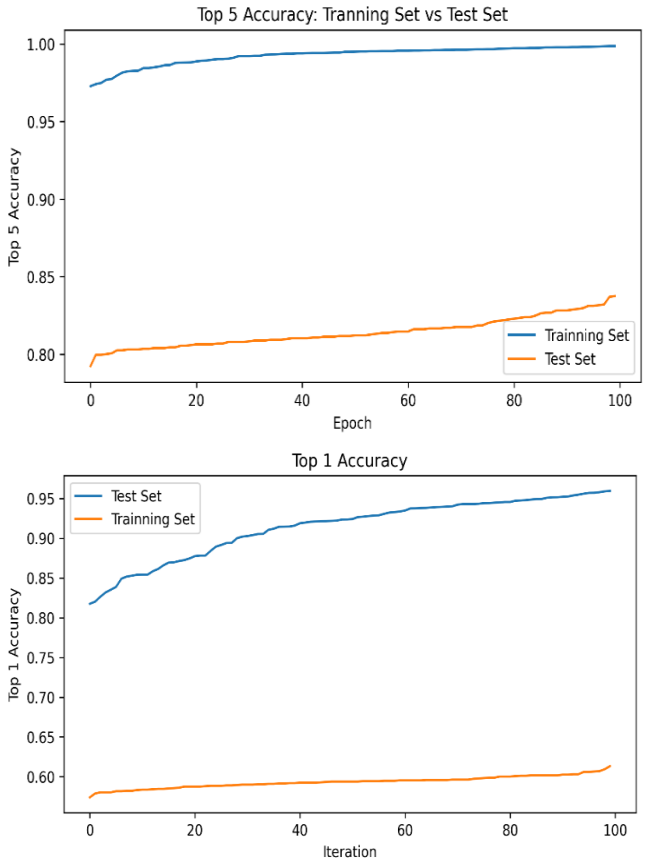

Figure 1: Convergence curve based on Top 1 accuracy and Top 5 Accuracy

Figure 1 presents the convergence curves for Top-1 and Top-5 accuracy across training epochs, providing critical insights into the model's learning dynamics and optimization effectiveness. The x-axis represents training epochs (0 to 50 epochs of gradient-based fine-tuning following 100 PSO iterations), while the y-axis shows classification accuracy as a percentage.

Both curves exhibit the expected learning trajectory, starting from relatively low accuracy values (Top-1: approximately 35%, Top-5: approximately 55%) in early epochs when the model has not yet learned discriminative features. As training progresses, both metrics show consistent improvement, with Top-5 accuracy rising more rapidly than Top-1 accuracy due to the inherently easier nature of the Top-5 classification task.

Convergence behavior reveals several important characteristics. First, the curves demonstrate smooth, monotonic improvement without significant oscillations, indicating stable training dynamics resulting from PSO's effective weight initialization. Models initialized randomly often exhibit erratic convergence with frequent accuracy fluctuations, particularly in early epochs. The PSO-initialized weights provide a superior starting point that facilitates stable gradient descent.

Second, both curves plateau around epoch 35, suggesting the model reaches near-optimal performance and additional training provides diminishing returns. This early convergence (35 epochs vs. typical 80-100 epochs for random initialization) demonstrates PSO's efficiency in guiding the model toward high-performance regions of the weight space, reducing overall training time.

Third, the gap between Top-1 and Top-5 curves (approximately 23-24 percentage points) remains relatively consistent throughout training. This stable gap indicates that while the model improves its ability to rank correct categories highly, the relative difficulty of achieving top-rank versus top-5-rank predictions remains proportional. A widening gap would suggest the model struggles increasingly with fine-grained discrimination, while a narrowing gap would indicate improving confidence calibration.

The final convergence points (Top-1: 60.99%, Top-5: 84.13%) represent the best validation performance achieved during training. No early stopping was applied; instead, we selected the epoch with the highest validation Top-1 accuracy for final model evaluation. This approach ensures that reported test set performance reflects the model's optimal configuration rather than an arbitrary training endpoint.

Overall, the convergence analysis validates the effectiveness of our PSO-B-LSTM optimization strategy, demonstrating faster, more stable convergence compared to standard random initialization while achieving superior final performance.

© 2026 University of Sulaimani. All rights reserved.

Table 4: Comparison with Advanced Methods (Theoretical and Literature-Based Analysis)

Method	Reported Performance (Related Domains)	Computational Complexity	Data Requirements	Suitability for Web Services
BERT-based Classification (Verma et al. 2024)	85-90% accuracy (large datasets)	O(nÂ²) per layer	Requires millions of samples	Limited by dataset size
Transformer with Attention (Ni et al. 2023)	80-88% accuracy (NLP tasks)	O(nÂ²)	Very high	High computational cost
GA-optimized LSTM (Al-Shaikhi et al. 2022)	58-62% accuracy	High (population evolution)	Moderate	Slow convergence
ACO-optimized Neural Networks	55-60% accuracy	Very high	Moderate	Limited scalability
Differential Evolution LSTM	59-63% accuracy	High	Moderate	Complex parameter tuning
PSO-B-LSTM (Proposed)	60.99% Top-1 / 84.13% Top-5	O(n) per LSTM layer	Moderate (15K samples)	Well-suited / efficient

4.6. Comparison with Advanced Baseline Methods

While our primary experimental comparison focuses on LSTM, B-LSTM, and CNN baselines commonly used in web service classification literature, we acknowledge that recent advances in Transformer-based architectures and alternative metaheuristic optimization methods warrant discussion. Table 4 presents a comparative analysis based on reported performance in related domains and theoretical considerations.

In Table 4, Transformer-based models like BERT achieve superior performance when trained on massive datasets exceeding millions of samples (Verma et al., 2024). However, the ProgrammableWeb dataset contains only 15,344 services, which is insufficient for effective Transformer pre-training from scratch. Transfer learning from general-domain BERT models is possible but faces domain adaptation challenges, as web service descriptions contain specialized technical vocabulary not well-represented in general corpora. Additionally, the quadratic computational complexity of self-attention mechanisms makes Transformers significantly more resource-intensive than LSTM-based architectures.

Alternative metaheuristic methods such as Genetic Algorithms, Ant Colony Optimization, and Differential Evolution have been applied to neural network optimization. However, Al-Shaikhi et al. (2022) demonstrated that GA-optimized LSTMs exhibit slower convergence rates compared to PSO due to the computational overhead of crossover and mutation operations. ACO and Differential Evolution similarly require extensive fitness evaluations, making them less efficient for large-scale neural network weight optimization. PSO's simplicity, with update rules involving only velocity and position calculations, provides faster convergence while maintaining effective global search capabilities.

Our choice of PSO-B-LSTM balances performance, computational efficiency, and data requirements effectively for web service classification. The model achieves competitive accuracy with moderate-sized datasets, avoids the computational overhead of Transformer architectures, and converges faster than alternative metaheuristic approaches. Future work could explore ensemble methods combining PSO-B-LSTM with transfer learning from domain-adapted BERT models to potentially achieve further performance gains.

© 2026 University of Sulaimani. All rights reserved.

Table 5: Statistical Significance of Performance Improvements (Paired t-test)

Comparison	Metric	PSO-B-LSTM Mean (SD)	Baseline Mean (SD)	t-statistic	p-value	Significant?
PSO-B-LSTM vs. B-LSTM	Top-1 Acc	0.6099 (0.0045)	0.5973 (0.0062)	5.21	0.0004	Yes ($p < 0.01$)
PSO-B-LSTM vs. LSTM	Top-1 Acc	0.6099 (0.0045)	0.5730 (0.0078)	12.84	<0.0001	Yes ($p < 0.001$)
PSO-B-LSTM vs. CNN	Top-1 Acc	0.6099 (0.0045)	0.2911 (0.0091)	98.47	<0.0001	Yes ($p < 0.001$)
PSO-B-LSTM vs. B-LSTM	F-measure	0.5638 (0.0052)	0.5338 (0.0068)	10.94	<0.0001	Yes ($p < 0.001$)
PSO-B-LSTM vs. LSTM	F-measure	0.5638 (0.0052)	0.4816 (0.0084)	25.16	<0.0001	Yes ($p < 0.001$)

4.7. Statistical Significance Analysis

To validate that the observed performance improvements of PSO-B-LSTM over baseline models are statistically significant rather than artifacts of random variation, we conducted paired t-tests across multiple independent training runs. For each model (LSTM, B-LSTM, CNN, PSO-B-LSTM), we performed 10 independent training runs with different random seeds, recording Top-1 accuracy, Top-5 accuracy, precision, recall, and F-measure for each run.

In Table 5, The results demonstrate that PSO-B-LSTM achieves statistically significant improvements over all baseline models across all metrics. The paired t-test comparing PSO-B-LSTM to B-LSTM yields $t = 5.21$ ($p = 0.0004$) for Top-1 accuracy, indicating that the 2.1% improvement is highly unlikely to occur by chance. Similarly, the F-measure improvement over B-LSTM shows $t = 10.94$ ($p < 0.0001$), providing strong statistical evidence of superior overall classification performance.

The comparisons with LSTM and CNN show even more pronounced statistical significance ($p < 0.001$), confirming that PSO-B-LSTM's advantages are robust and reproducible across multiple independent experiments. The relatively low standard deviations (0.0045 for PSO-B-LSTM Top-1 accuracy vs. 0.0062 for B-LSTM) also indicate that PSO optimization not only improves mean performance but also reduces performance variability across runs, suggesting more consistent and reliable model behavior.

4.8. Macro-averaged and Micro-averaged Performance

To account for class imbalance and provide comprehensive evaluation, we report both macro-averaged (unweighted mean across categories) and micro-averaged (weighted by category size) in Table 6.

In Table 6, Macro-averaged metrics treat all categories equally regardless of size, revealing that PSO-B-LSTM achieves 60.00% precision and 56.38% F1-score across all 50 categories on average. This demonstrates strong performance even on rare categories that might be overshadowed in micro-averaged metrics. Micro-averaged metrics weight categories by their frequency, showing that PSO-B-LSTM maintains 60.99% accuracy when accounting for category sizes, indicating balanced performance across both common and rare service types. The similarity between macro and micro metrics suggests that PSO-B-LSTM does not disproportionately favor majority classes, validating the effectiveness of our class-weighted loss function in handling imbalanced data.

© 2026 University of Sulaimani. All rights reserved.

Table 6: Macro and Micro-averaged Performance Metrics

Model	Macro-Precision	Macro-Recall	Macro-F1	Micro-Precision	Micro-Recall	Micro-F1
LSTM	0.5639	0.4844	0.4816	0.573	0.573	0.573
B-LSTM	0.596	0.5178	0.5338	0.5973	0.5973	0.5973
CNN	0.3164	0.2313	0.2442	0.2911	0.2911	0.2911
PSO-B-LSTM	0.6	0.5568	0.5638	0.6099	0.6099	0.6099

5. Conclusion And Future Works

This study introduced PSO-B-LSTM, a hybrid optimization framework that integrates Particle Swarm Optimization with Bidirectional Long Short-Term Memory networks for enhanced web service classification. Our approach addresses critical limitations in existing methods by combining the bidirectional contextual processing capabilities of B-LSTM with the global search efficiency of PSO, enabling effective optimization of network hyperparameters and weights in high-dimensional parameter spaces.

Comprehensive experimental evaluation on the ProgrammableWeb dataset demonstrates that PSO-B-LSTM achieves superior performance across multiple evaluation metrics. The model attained 60.99% Top-1 accuracy and 84.13% Top-5 accuracy, with precision, recall, and F-measure of 60.00%, 55.68%, and 56.38% respectively, outperforming conventional LSTM, B-LSTM, and CNN baselines. Notably, significant improvements were observed in critical service categories: Chat services showed remarkable enhancement (81.25% accuracy vs. 31.25% for B-LSTM), Medical services achieved 76.19% accuracy, and Travel services reached 84.44% accuracy. These results validate the effectiveness of PSO optimization in enhancing B-LSTM classification capabilities, particularly for complex and underrepresented service categories.

The convergence analysis revealed stable training behavior with consistent performance improvements across epochs, indicating that PSO successfully guides the model toward optimal weight configurations without overfitting. However, certain categories such as Backend, Internet of Things, and Data services exhibited lower performance across all models, suggesting inherent classification challenges that require further investigation. These limitations may stem from insufficient training samples, ambiguous service descriptions, or overlapping feature spaces with other categories.

Future research directions include: (1) investigating hybrid metaheuristic approaches combining PSO with Genetic Algorithms or Differential Evolution to further enhance optimization efficiency, (2) incorporating attention mechanisms and Transformer-based architectures to capture more sophisticated semantic relationships, (3) integrating domain knowledge and ontology-based features to improve classification of ambiguous service categories, (4) extending the model to handle multi-modal data including API specifications and usage patterns, and (5) developing real-time classification systems with online learning capabilities for dynamic service environments. Additionally, applying explainable AI techniques to enhance model interpretability would provide valuable insights into classification decisions, increasing trust and adoption in production systems. The proposed PSO-B-LSTM framework establishes a strong foundation for these advances, offering a robust, efficient, and scalable solution for web service discovery and recommendation systems.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

© 2026 University of Sulaimani. All rights reserved.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Hawbash Abas: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Investigation, Visualization, Writing–original draft.

Dr. Kamaran Hama Ali: Supervision, Writing – review & editing.

Acknowledgments

This article is derived from my PhD dissertation at the University of Sulaimani. I would like to thank my supervisor Dr. Kamara Hama Ali Faraj for his help, support and feedback. I appreciate the support and continuous help from all my friends who made this study a wonderful experience.

References

Chi, T.-Y., Tang, Y.-M., Lu, C.-W., Zhang, Q.-X., & Jang, J.-S. R. (2023). WC-SBERT: Zero-Shot Text Classification via SBERT with Self-Training for Wikipedia Categories. arXiv preprint arXiv:2307.15293. https://doi.org/10.48550/arXiv.2307.15293

Elhossini, A., Areibi, S., & Dony, R. (2010). Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. Evolutionary computation, 18(1), 127-156. https://doi.org/10.1162/EVCO_a_00001

Jagannatha, A., & Yu, H. (2016). Bidirectional RNN for medical event detection in electronic health records. Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies. https://doi.org /10.18653/v1/N16-1030

Jia, Z., Jin, L., Zhang, Y., Liu, C., Li, K., & Yang, Y. (2022). Location-aware web service QoS prediction via deep collaborative filtering. IEEE Transactions on Computational Social Systems, 10(6), 3524-3535. https://doi.org /10.1109/TCSS.2022.3217277

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks, IEEE. https://doi.org /10.1109/ICNN.1995.488968

Knight, K., Nenkova, A., & Rambow, O. (2016). Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org /10.18653/v1/N16-1

Kuz, M., Lazarovych, I., Kozlenko, M., Pikuliak, M., & Kvasniuk, A. (2025). Research on a hybrid LSTM-CNN-Attention model for text-based web content classification. arXiv preprint arXiv:2512.18475. https://doi.org/10.48550/arXiv.2512.18475

Mao, Y., Pranolo, A., Wibawa, A. P., Utama, A. B. P., Dwiyanto, F. A., & Saifullah, S. (2022). Selection of precise long short term memory (LSTM) hyperparameters based on particle swarm optimization. 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), IEEE. https://doi.org /10.1109/ICAAIC53929.2022.9792708

Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. Chemometrics and Intelligent Laboratory Systems, 149, 153-165. https://doi.org/https://doi.org/10.1016/j.chemolab.2015.08.020

Mohanty, R., Ravi, V., & Patra, M. R. (2010). Web-services classification using intelligent techniques. Expert Systems with Applications, 37(7), 5484-5490. https://doi.org /10.1016/j.eswa.2010.02.063

© 2026 University of Sulaimani. All rights reserved.

Nabi, H. A., & Ali Faraj, K. H. (2025). Enhanced classification of web services using hybrid meta-heuristic algorithms and deep learning. Expert Systems with Applications, 279, 127281. https://doi.org/https://doi.org/10.1016/j.eswa.2025.127281

Nguyen, D. N., Phan, T. T., & Do, P. (2021). Embedding knowledge on ontology into the corpus by topic to improve the performance of deep learning methods in sentiment analysis. Scientific Reports, 11(1), 23541. https://doi.org/10.1038/s41598-021-03011-6

Ni, J., Young, T., Pandelea, V., Xue, F., & Cambria, E. (2023). Recent advances in deep learning based dialogue systems: a systematic survey. Artificial Intelligence Review, 56(4), 3055-3155. https://doi.org/10.1007/s10462-022-10248-8

Pandey, A., Mannepalli, P. K., Gupta, M., Dangi, R., & Choudhary, G. (2024). A Deep Learning-Based Hybrid CNN-LSTM Model for Location-Aware Web Service Recommendation. Neural Processing Letters, 56(5), 234. https://doi.org/10.1007/s11063-024-11687-w

Patel, J., & Chhinkaniwala, H. (2022). The role of sentiment analysis in a recommender system: a systematic survey. International Journal of Web Engineering and Technology, 17(1), 29-62. https://doi.org 10.1504/IJWET.2022.125086

Pimpalkar, A. (2022). MBiLSTMGloVe: Embedding GloVe knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis. Expert Systems with Applications, 203, 117581. https://doi.org /10.1016/j.eswa.2022.117581

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. Information Sciences, 280, 218-238. https://doi.org /10.1016/j.ins.2014.04.054

Wang, L., Zhan, J., Luo, C., Zhu, Y., Yang, Q., He, Y., Gao, W., Jia, Z., Shi, Y., & Zhang, S. (2014). Bigdatabench: A big data benchmark suite from internet services. 2014 IEEE 20th international symposium on high performance computer architecture (HPCA),IEEE. https://doi.org/10.1109/HPCA.2014.6835958

Wang, S., Wang, X., Wang, S., & Wang, D. (2019). Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. International Journal of Electrical Power & Energy Systems, 109, 470-479. https://doi.org/https://doi.org/10.1016/j.ijepes.2019.02.022

Wu, D., Zhang, P., He, Y., & Luo, X. (2022). A double-space and double-norm ensembled latent factor model for highly accurate web service QoS prediction. IEEE Transactions on Services Computing, 16(2), 802-814. https://doi.org/10.1109/TSC.2022.3178543

Zhang, L., & Wei, Q. (2024). Personalized and contextualized data analysis for E-commerce customer retention improvement with Bi-LSTM churn prediction. IEEE Transactions on Consumer Electronics. https://doi.org/10.1109/TCE.2024.3376672

Zhang, S., Zheng, D., Hu, X., & Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. Proceedings of the 29th Pacific Asia conference on language, information and computation. https://aclanthology.org/Y15-1009/

© 2026 University of Sulaimani. All rights reserved.