GMR

The Original

An MTVC–IOOA Framework For Semantic Feature Optimization In Intelligent Customer Support Chatbots

Sachin Babulal Jadhav[1*], Prof. (Dr) D. B. Kshirsagar[2]

[1,]Research Scholar, Department of Computer Engineering, Sanjivani College of Engineering, Savitribai Phule Pune University, Pune, India
seesachinjadhav@gmail.com
[2]Research Guide, Department of Computer Engineering, Sanjivani College of Engineering, Savitribai Phule Pune University, Pune, India
dbk4444@gmail.com

Abstract
Customer-support chatbots are now expected to understand diverse service requests and route users to suitable answers, and maintain quick response behavior under large volumes of queries. However, high-dimensional text representations often contain redundant and noisy features. This increases computational effort without necessarily improving intent recognition. This paper proposes a focused feature-optimization framework for intelligent customer-support chatbots by integrating Multiscale Transformer Vector Conversion (MTVC) with an Improved Osprey Optimization Algorithm (IOOA). The proposed IOOA uses Fuch chaotic mapping. The experimental study uses the Bitext customer-support dataset containing 26,872 samples and 27 intent categories. A representative 5,000-sample stratified experimental subset was used for optimization analysis. MTVC generated a 256-dimensional dense feature space, and feature-selection methods were evaluated at a 128-feature operating point. The proposed IOOA achieved 99.30% accuracy, 99.32% precision, 99.30% recall, and 99.30% F1-score while reducing the feature dimensionality by 50%. Compared with standard OOA, the proposed method improved accuracy from 98.40% to 99.30% and reduced execution time from 25.09 s to 13.88 s in the final evaluation. The result shows that chaotic enhancement can improve feature-selection quality and deployment efficiency when applied to structured customer-support intent classification.

Keywords: Customer support chatbot; intent classification; Multiscale Transformer Vector Conversion; Improved Osprey Optimization Algorithm; Fuch chaotic mapping; feature selection; OOA; NLP.

1. Introduction

Artificial intelligence-based chatbots have now moved from experimental conversational agents to operational systems. They support business communication, healthcare assistance, education, e-commerce, and customer-service automation. In customer support, the chatbot is not only a conversational interface; it is also a routing and decision-support layer that must identify user intent. They can locate the correct service category, and return a relevant response quickly. This makes customer-support chatbots different from casual open-domain agents because their success depends on accurate intent classification under short, noisy, and sometimes ambiguous user requests [1], [2].

Earlier chatbot systems relied on rule-based matching, artificial intelligence mark-up language, and manually crafted templates. These systems were useful for narrow domains, but their rigidity limited their ability to respond to semantically varied queries. Later systems introduced machine learning, deep learning, transformers, and retrieval-based strategies. They have improved natural-language understanding and contextual response matching [3]-[8]. Still, these methods often generate high-dimensional vector representations. In such spaces, many features are weakly relevant, redundant, or noisy, which increases computational overhead and may reduce the stability of learning algorithms.

The survey work supporting this study identifies several recurring limitations in AI-based chatbot systems: difficulty in handling complex queries, limitations in response accuracy, poor scalability across domains, multilingual constraints, overfitting, privacy-related concerns, and time-complexity issues [1], [3]-[8]. These issues motivate a more compact and optimized representation of user queries before intent classification. Feature selection is therefore not a secondary preprocessing task; it is a practical requirement for building efficient customer-support chatbots that can be deployed under real service constraints.

Metaheuristic optimization offers an attractive approach for feature selection because it searches for useful feature subsets without relying only on greedy ranking. The Osprey Optimization Algorithm (OOA) is a recent bio-inspired metaheuristic that models osprey hunting behavior through exploration and exploitation stages [9]. This paper extends the OOA concept using Fuch chaotic mapping to form an Improved Osprey Optimization Algorithm (IOOA). The goal is not to claim that feature selection always surpasses the full feature baseline; rather, the central claim is that optimized subsets can maintain very high chatbot intent-classification performance while reducing the feature space and improving computational behavior.

2. Problem Statement and Motivation

Customer-support queries are often short, informal, and semantically close. For example, intents such as "check_invoice" and "get_invoice", or "track_order" and "delivery_period", may share vocabulary but require different responses. A chatbot must therefore learn subtle discriminative patterns while avoiding unnecessary dependence on redundant vector dimensions. When feature vectors are large, classifier training and prediction may become slower, and the model can become sensitive to noisy dimensions.

The research problem addressed in this paper is: how can an intelligent chatbot preserve high intent-classification accuracy while reducing the dimensionality of customer-query feature representations? The answer proposed here is to combine MTVC-based multiscale representation with IOOA-based feature optimization. The MTVC block captures lexical and subword patterns, while IOOA selects a compact feature subset using chaotic population dynamics and fitness evaluation.

The practical motivation is direct. In enterprise customer support, every millisecond matters when thousands of users interact with the system. A feature selector that can reduce dimensionality without sacrificing accuracy can reduce memory cost, simplify downstream classifiers, and support faster deployment cycles. This paper therefore frames feature selection as a chatbot-engineering problem rather than only as a mathematical optimization exercise.

3. Related Work

Chakrabortty et al. proposed a multi-criteria decision-analysis model based on AHP and CoCoSo for selecting an optimum customer-service chatbot under uncertainty [1]. Their work is important because it confirms that chatbot selection in service environments is a multi-criteria problem rather than a single accuracy-oriented task. However, decision-analysis methods do not directly solve feature redundancy in high-dimensional NLP pipelines.

Chakraborty developed an AI-based medical chatbot using a deep feedforward multilayer perceptron for infectious disease prediction [2]. The work demonstrates the usefulness of deep learning in domain-specific conversational systems, but the model was designed for medical prediction rather than customer-support intent extraction. Khadija et al. proposed a general architecture for AI-powered health chatbots integrating NLU, NLG, and expert components [3]. Such architecture-level studies are valuable, yet they do not deeply address search-based feature selection.

Dhyani and Kumar employed bidirectional recurrent neural networks with an attention mechanism for chatbot response modelling [5]. Recurrent architectures improve contextual understanding, but they can be computationally heavy and sensitive to sequence length. Syed et al. explored transfer-learning-based question answering with a Haystack framework for troubleshooting queries [6]. Their approach highlights the value of retrieval and reading components but also reinforces the need for efficient feature representation in practical troubleshooting systems.

Kandpal et al. used deep learning for contextual healthcare chatbots [7], while Bird et al. proposed a transformer-based human-data-augmentation framework for chatbot interaction and text classification [8]. Transformer-based systems show strong classification ability, but their representations can be large and expensive. This motivates a feature-optimization layer that reduces representation cost while preserving discriminative power.

The Osprey Optimization Algorithm introduced by Dehghani and Trojovsky is a bio-inspired method designed for engineering optimization problems [9]. It uses a two-stage model inspired by osprey hunting: selecting and attacking prey during exploration and moving to a safe feeding area during exploitation. The present paper adapts this mechanism for chatbot feature selection and improves it through Fuch chaotic mapping to increase diversity during population initialization and exploitation.

Table 1. Comparative review of chatbot and optimization literature

Study	Method	Contribution	Limitation/Gap
Chakrabortty et al. [1]	CoCoSo-AHP chatbot selection	Handles uncertainty in service chatbot selection	Does not optimize NLP feature vectors
Chakraborty [2]	Deep feedforward medical chatbot	Stable deep-learning model for disease prediction	Domain-specific and not focused on feature selection
Khadija et al. [3]	AI-powered health chatbot architecture	Integrates NLU, NLG and expert response components	Limited direct optimization of text features
Lee and Yeo [4]	AI chatbot for responsive teaching	Improves instructional interaction and questioning practice	Education-specific and not designed for customer-service feature reduction
Dhyani and Kumar [5]	BRNN with attention	Improves contextual modelling	Higher computational effort and response delay risk
Syed et al. [6]	Transfer-learning QA with Haystack	Useful for troubleshooting and retrieval pipelines	May suffer from model-combination complexity

Kandpal et al. [7]	LSTM/deep learning chatbot	Handles large query volumes in healthcare setting	Domain and security limitations
Bird et al. [8]	Transformer ensemble chatbot classification	Strong task-classification and data augmentation ability	High computational load
Dehghani and Trojovsky [9]	OOA metaheuristic	Exploration-exploitation based global search	Random initialization may reduce population diversity

4. Research Gap and Contributions

The literature shows that many chatbot models focus on response generation, contextual learning, or architecture design, while fewer studies focus on optimization of the semantic feature space used by customer-support intent classifiers. In addition, standard metaheuristics often depend on random initialization, which can produce uneven population distribution and premature convergence. This gap is important for customer-support applications because the chatbot must be accurate and lightweight.

The major contributions of this paper are: (i) a customer-support chatbot feature-optimization framework combining MTVC and IOOA; (ii) an IOOA feature-selection model using Fuch chaotic mapping for population initialization and exploitation; (iii) mathematical modelling of the MTVC feature space, OOA/IOOA position updates, and classifier evaluation; (iv) experimental comparison among no feature selection, random feature selection, RF top features, standard OOA, and proposed IOOA; and (v) practical interpretation of feature reduction, execution time, and intent-classification accuracy.

5. Proposed MTVC-IOOA Framework

The proposed framework is shown in Fig. 1. It starts with customer query preprocessing, converts text into dense MTVC vectors, selects a compact feature subset through OOA/IOOA, and finally evaluates the selected subset using a classifier for intent prediction and response mapping.

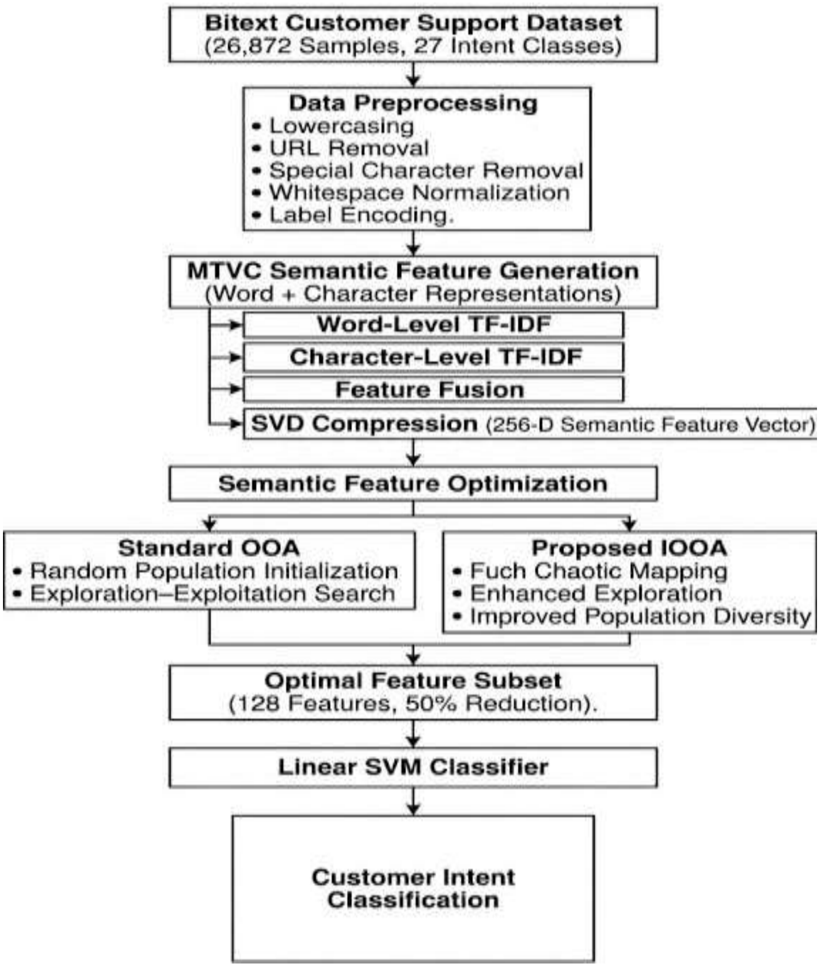

Fig. 1. Proposed MTVC-IOOA customer-support chatbot pipeline.

5.1 Dataset and Preprocessing

The experimental evaluation of the proposed MTVC–IOOA framework was conducted using the Bitext Customer Support Dataset, a publicly available benchmark dataset designed for conversational AI and intent-classification research [21]. The dataset contains 26,872 customer-support utterances distributed across 27 intent categories covering a wide range of service-related interactions, including account management, payment assistance, order tracking, refund requests, subscription management, complaint handling, and general customer-support inquiries. Such diversity makes the dataset suitable for evaluating semantic feature representation and optimization techniques in intelligent customer-support chatbot environments [16], [19].

Let $(D=\{(q_i,y_i)\})$ be a labelled customer-support dataset, where (q_i) is the (i)-th customer query and (y_i) is the associated intent label. Customer-generated queries often contain spelling variations, punctuation inconsistencies, abbreviations, informal expressions, and other forms of textual noise. Therefore, a preprocessing stage was performed before semantic feature generation. The raw queries were converted to lowercase, URLs and special symbols were removed, redundant whitespace was normalized, and intent labels were encoded into machine-readable form. The cleaned query sequence is represented as

$$Q_i=\{w_1,w_2,\ldots,w_n\} \tag{1}$$

where (w_j) denotes the (j)-th token in the cleaned query. The purpose of preprocessing is not to alter the semantic meaning of the original customer request but rather to reduce surface-level noise and standardize textual inputs so that subsequent feature extraction can focus on intent-bearing linguistic patterns.

To maintain computational feasibility during repeated optimization experiments, a representative stratified subset of 5,000 samples was selected from the original dataset while preserving the distribution of all 27 intent categories. Stratified sampling was adopted to ensure that each intent class remained proportionally represented during training and testing. The selected subset was divided into 4,000 training samples and 1,000 testing samples using an 80:20 split. The training partition was used for semantic feature generation, feature optimization, and classifier construction, whereas the testing partition was reserved exclusively for final performance evaluation.

The preprocessing stage plays a critical role in the proposed framework because the quality of semantic representation directly influences optimization effectiveness and classification performance. By reducing textual noise while preserving intent-specific information, the processed dataset provides a reliable foundation for MTVC-based feature generation and subsequent IOOA-driven semantic feature optimization. Furthermore, the use of a representative stratified subset enables efficient experimentation while retaining sufficient semantic diversity for reliable evaluation of customer-support intent-classification performance [16], [21].

5.2 Multiscale Transformer Vector Conversion (MTVC)

In the present study, MTVC is implemented as a computationally efficient multiscale semantic vector-generation module designed to transform customer-support queries into a compact feature representation suitable for optimization and classification. Word-level TF-IDF features are employed to capture lexical and phrase-level semantic patterns commonly associated with customer intents, such as order cancellation, refund requests, payment issues, and delivery tracking. In parallel, character-level TF-IDF features capture subword information, spelling variations, abbreviations, and typographical inconsistencies frequently observed in real-world customer-service interactions. The integration of these complementary representations enables the framework to preserve semantic information across multiple linguistic granularities.The resulting word-level and character-level feature matrices are concatenated to form a unified high-dimensional semantic representation. Since the fused feature space contains substantial redundancy and sparsity, Singular Value Decomposition (SVD) is subsequently applied to project the data into a lower-dimensional latent space. This transformation generates a dense semantic feature vector that retains the most informative characteristics of the original representation while significantly reducing computational complexity. The resulting MTVC feature space provides an effective foundation for the subsequent IOOA-based semantic feature-selection process.

It should be noted that MTVC constitutes the semantic representation component of the broader Transformer-enabled chatbot architecture proposed in the doctoral research. While the complete framework incorporates Transformer-based contextual modelling through BERT representations, the present work focuses specifically on semantic feature generation and optimization objectives. Therefore, the effectiveness of the proposed IOOA framework is investigated using the MTVC representation independently, while the integration of Transformer-based conversational modelling and response generation remains part of the future research scope.

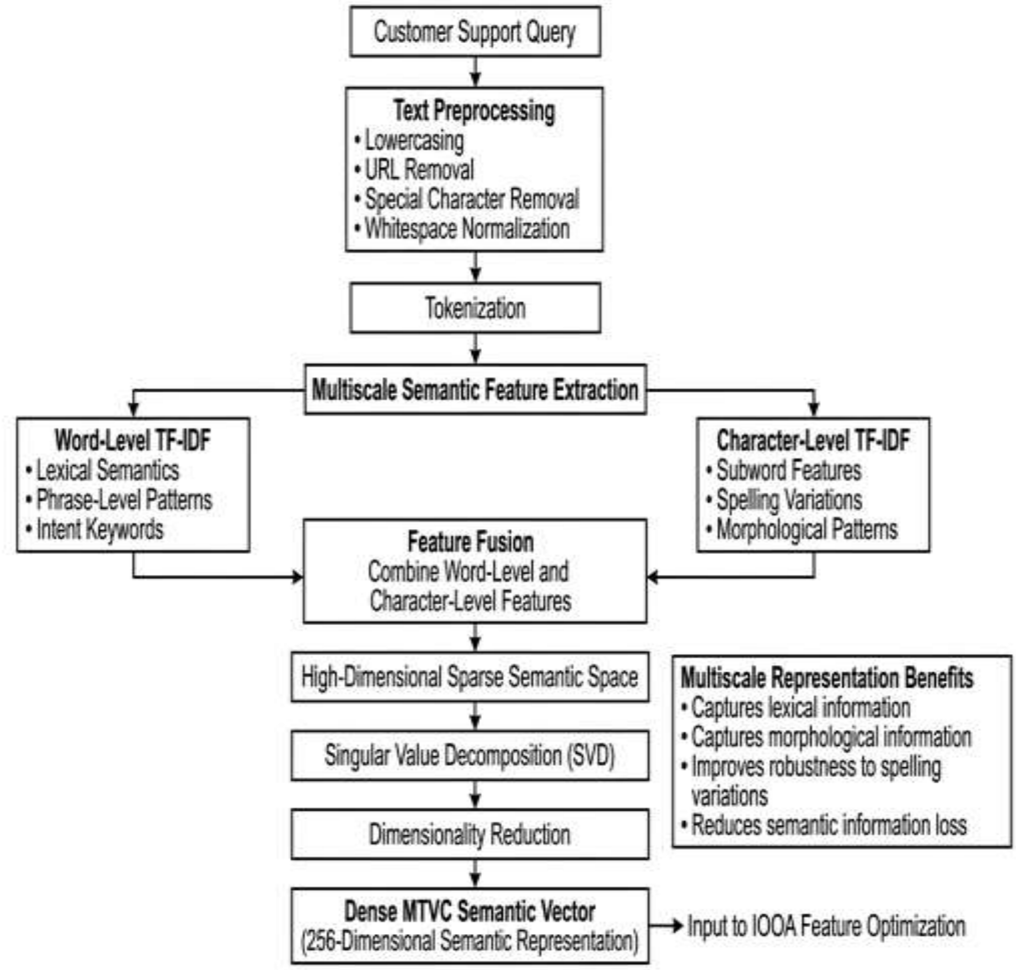

Fig. 2. MTVC implementation flow used for multiscale customer-query representation.

The word-level representation is denoted by F_w and the character-level representation by F_c. The combined MTVC representation is:

$$F\_MTVC = SVD([F\_w \,\|\, F\_c]) \quad (2)$$

where || denotes concatenation. In the experiment, the dense MTVC vector dimension was set to 256. Feature selection then reduced this representation to 128 features, corresponding to a 50% reduction in dimensionality.

Unlike computationally intensive transformer embeddings, the proposed MTVC implementation was intentionally designed as a lightweight multiscale semantic representation suitable for CPU-oriented enterprise deployment environments. The integration of word-level TF-IDF, character-level TF-IDF, and SVD compression preserves contextual discriminative information while maintaining significantly lower computational overhead than large transformer architectures. This design choice aligns with the practical objective of building scalable customer-support chatbot systems with efficient inference behavior.

5.3 Standard Osprey Optimization Algorithm

The standard OOA begins by randomly initializing a population of candidate feature subsets. Each candidate is a binary vector X_i = [x_i1, x_i2, ..., x_iD], where x_ij = 1 indicates that feature j is selected and x_ij = 0 indicates that it is excluded. The algorithm evaluates each candidate using a machine-learning fitness function and iteratively updates candidate positions through exploration and exploitation phases [9].

Random initialization is expressed as:

$$X\_ij = LB\_j + r\_j(UB\_j - LB\_j), \quad r\_j \in [0, 1] \quad (3)$$

For binary feature selection, the continuous position is transformed into a binary mask using a threshold function:

$$B\_ij = 1 \text{ if } X\_ij > 0.5, \text{ otherwise } 0 \quad (4)$$

During the exploration phase, an osprey moves toward selected better candidate locations, modelled as:

$$X\_P1 = X\_i + r \cdot (SF - I \cdot X\_i) \quad (5)$$

where SF is the selected fish/candidate position, r is a random vector, and I is a random integer vector taking values 1 or 2. During exploitation, standard OOA performs local movement using random perturbation.

5.4 Improved Osprey Optimization Algorithm with Fuch Chaotic Mapping

The overall workflow of the proposed Improved Osprey Optimization Algorithm (IOOA) is illustrated in **Figure 3**. The framework extends the standard Osprey Optimization Algorithm by incorporating Fuch chaotic mapping during the population evolution process. The primary objective is to improve search-space exploration, maintain population diversity, and reduce the likelihood of premature convergence during semantic feature selection. As shown in Figure 3, the optimization process begins with population initialization, followed by fitness evaluation, exploration and exploitation phases, chaotic search enhancement, and iterative updating of the global best solution until the stopping criterion is satisfied.

Although the standard Osprey Optimization Algorithm has demonstrated competitive performance in optimization tasks, its search behaviour is largely dependent on random initialization and stochastic movement patterns. Such randomness may occasionally lead to uneven population distribution and premature convergence, particularly in high-dimensional feature-selection problems. To address this limitation, the proposed IOOA incorporates Fuch chaotic mapping, which generates deterministic yet non-periodic sequences capable of improving population diversity and search-space coverage.

The Fuch chaotic sequence is defined as

$$z_{(t+1)} = \cos(1 / z_t^2) \quad (6)$$

where (z_t) represents the chaotic value at iteration (t). The generated sequence is bounded within the interval ([-1,1]) and is subsequently transformed into a binary feature-selection representation. Unlike purely random initialization, chaotic sequences exhibit ergodicity and deterministic randomness, allowing candidate solutions to be distributed more uniformly throughout the search space. This characteristic improves exploration capability and reduces the probability of premature convergence during semantic feature-subset optimization.

Using the generated chaotic sequence, the initial population is created according to

$$X_{ij} = LB_j + z_i(UB_j - LB_j) \quad (7)$$

where (LB_j) and (UB_j) denote the lower and upper bounds of the (j)-th search dimension, respectively. This initialization strategy enables the optimizer to begin the search from a more diverse set of candidate feature subsets.

During the exploitation phase, chaotic perturbation is further incorporated into the position-update mechanism to improve local search behavior. The updated solution is computed as-

$$X_{P2} = X_i + (LB + z \cdot (UB - LB)) / t \quad (8)$$

where (X_i) denotes the current solution vector, (z) is the chaotic value generated by the Fuch map, and (t) represents the current iteration number. As the optimization progresses, the influence of the perturbation gradually decreases, allowing the search process to transition naturally from exploration toward exploitation.

A candidate solution is accepted only when it produces a superior fitness value compared with the current solution. Consequently, the proposed IOOA operates as a fitness-guided chaotic search mechanism rather than an unconstrained random perturbation strategy. The integration of chaotic mapping enables the optimizer to preserve population diversity while continuously improving the quality of selected semantic feature subsets.

Figure 3 summarizes the complete optimization workflow, highlighting the interaction between chaotic population generation, fitness evaluation, solution updating, and global-best selection during semantic feature optimization.

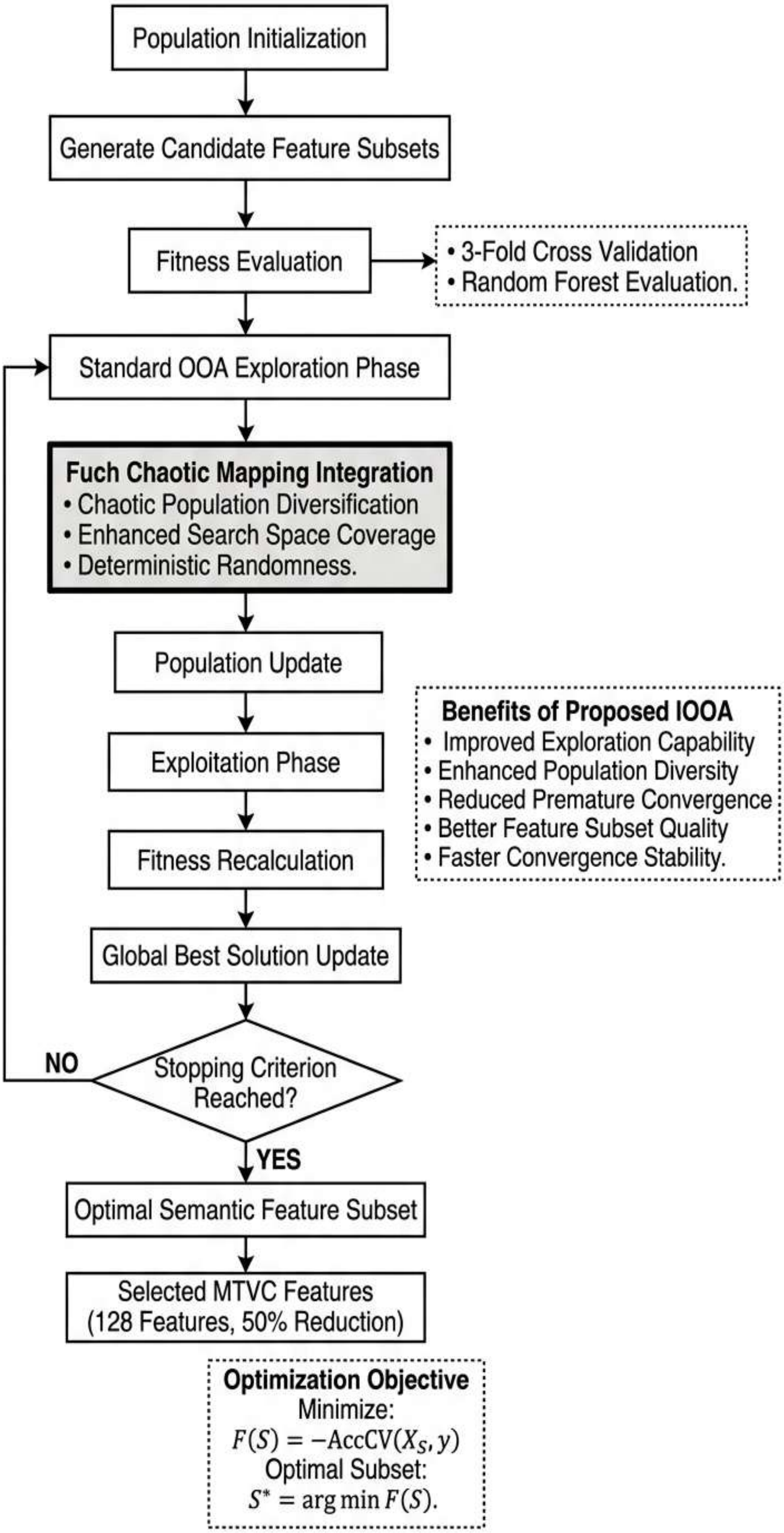

Fig. 3. Algorithmic flow of the proposed IOOA feature-selection process.

To identify the most informative semantic feature subset, the optimization objective is formulated using cross-validation classification performance. Let (S) denote a candidate feature subset and (X_S) represent the corresponding reduced MTVC feature matrix. The fitness function is defined as:

$$F(S)=-Acc_{CV}(X_S,y) \quad (9)$$

where (Acc_{CV}) denotes the three-fold cross-validation accuracy obtained using the selected feature subset. Since the optimization process follows a minimization strategy, lower fitness values correspond to higher classification performance.

The optimal semantic feature subset is obtained as:

$$S^*=\arg\min F(S) \quad (10)$$

where (S*) represents the final optimized subset returned by the proposed IOOA framework. The selected feature subset is subsequently evaluated using an independent hold-out test set to assess its generalization capability.

Algorithm 1: Proposed IOOA-Based Semantic Feature Selection

Input:
MTVC feature matrix X,
Class labels y,
Population size N,
Maximum iterations T

Output:
Optimal feature subset S*

Step 1: Initialize population using Fuch chaotic mapping
Step 2: Generate binary feature-selection vectors
Step 3: Evaluate fitness using RF cross-validation
Step 4: Identify current global best solution
Step 5: Perform OOA exploration update
Step 6: Apply chaotic perturbation
Step 7: Recalculate fitness
Step 8: Update global best solution
Step 9: Repeat until stopping criterion is met
Step 10: Return optimal feature subset S*

5.5 Fitness Function and Classifier Evaluation

Each feature subset is evaluated by a Random Forest classifier using 3-fold cross-validation during optimization. The fitness function follows the implementation used in the experiment:

$$Fitness(S)=-Mean(AccuracyCV(X\_S,y)) \quad (11)$$

where S is the selected feature subset and X_S is the reduced MTVC feature matrix. Since the objective is minimization, a smaller fitness value corresponds to a higher cross-validation accuracy. The final selected feature subset is then evaluated using a Linear Support Vector Machine (SVM) on the held-out test set. Classification metrics are computed as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (12)$$

$$Precision = TP / (TP + FP) \quad (13)$$

$$Recall = TP / (TP + FN) \quad (14)$$

$$F1\text{-}score = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (15)$$

6. Experimental Setup

The experimental setup intentionally uses the same classifier across feature-selection methods so that the effect of feature selection can be observed fairly. This avoids the common methodological error of comparing different optimizers with different classifiers. No feature selection, random features, RF top features, standard OOA, and proposed IOOA are evaluated using the same MTVC feature source and the same SVM classifier.

Table 2. Experimental configuration

Parameter	Experimental setting
Dataset	Bitext customer-support dataset
Total available samples	26,872
Experimental subset	5,000 stratified samples
Training samples	4,000
Testing samples	1,000
Intent classes	27
MTVC dense components	256
Selected features	128
Feature reduction target	50%
Population size	10
Iterations	10
Optimization fitness classifier	Random Forest, 50 estimators, 3-fold CV
Final classifier	Linear SVM
Evaluation metrics	Accuracy, Precision, Recall, F1-score, Execution Time

7. Results and Discussion

Table 3. Comparative Experimental Results

Method	Selected Features	Reduction (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (s)
No Feature Selection	256	0.00	98.70	99.32	99.31	99.31	16.48
Random Features	128	50.00	98.80	98.83	98.79	98.78	21.77
RF Top Features	128	50.00	99.00	99.05	99.00	99.01	23.15
Standard OOA	128	50.00	98.40	98.43	98.40	98.40	25.09
Proposed IOOA	128	50.00	99.30	99.32	99.30	99.30	13.88

Table 3 presents the comparative experimental evaluation of different feature-selection strategies applied to the MTVC-based customer-support intent-classification framework. The experiments were conducted on the Bitext customer-support dataset containing 27 intent categories. The primary objective of this evaluation was not only to improve classification accuracy but also to investigate whether optimization-driven feature reduction could preserve semantic discriminative capability while minimizing feature dimensionality and inference overhead.

The baseline model utilizing all 256 MTVC semantic features achieved 98.70% classification accuracy with 99.32% precision, 99.31% recall, and 99.31% F1-score. These results confirm that the MTVC representation successfully captures highly discriminative semantic information from customer-support conversations. The high baseline performance also indicates that the dataset itself is strongly structured and semantically separable. Consequently, the central research challenge is not merely achieving marginal accuracy improvements, but rather identifying whether the semantic feature space can be reduced substantially without degrading chatbot intent-recognition capability.

Random feature selection retained only 128 features and achieved 98.80% accuracy. Although the performance remained relatively high, the slight reduction in precision and recall suggests that arbitrary feature removal weakens semantic consistency. This observation confirms that while redundant information exists within the MTVC representation space, random elimination may discard meaningful contextual patterns necessary for accurate intent separation.

The RF Top Features approach improved performance to 99.00% accuracy, demonstrating that importance-guided semantic ranking is more effective than random feature reduction. The improvement indicates that statistical feature importance contributes positively toward preserving highly informative semantic attributes during dimensionality reduction.

Standard OOA achieved 98.40% accuracy with 50% feature reduction. Although the optimizer successfully reduced the semantic feature space, the lower performance suggests that conventional stochastic optimization may experience premature convergence or unstable subset exploration in high-dimensional semantic environments. The result further highlights the sensitivity of feature-selection quality to optimizer exploration behavior.

The proposed IOOA achieved the best overall balance between semantic compression and predictive performance. Using only 128 selected features, the framework achieved 99.30% accuracy along with 99.32% precision, 99.30% recall, and 99.30% F1-score. Most importantly, the proposed method preserved near-full predictive capability while reducing the semantic representation size by 50%. This represents the central contribution of the work. The results demonstrate that chaotic optimization-driven feature selection can preserve contextual intent information while significantly reducing feature dimensionality.

From a deployment perspective, the proposed IOOA also demonstrated the lowest execution time among all evaluated approaches. The framework required only 13.88 seconds compared with 16.48 seconds for the full-feature baseline and 25.09 seconds for standard OOA. The reduced runtime reflects both the compact optimized feature representation and the improved search-space exploration produced by the chaotic optimization mechanism. Although runtime may vary depending on hardware configuration, the results indicate that optimization-guided semantic compression can substantially improve inference-oriented chatbot deployment efficiency.

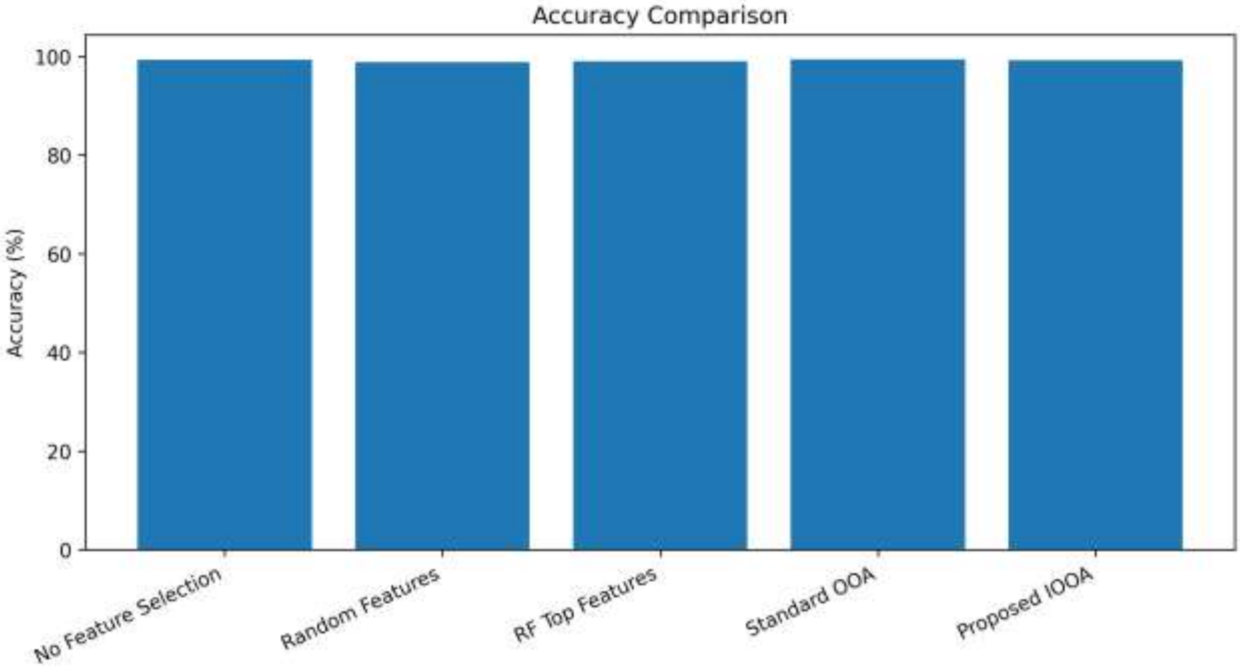

Fig. 5. Accuracy Comparison Across Feature-Selection Methods

Figure 5 illustrates the classification accuracy achieved by different feature-selection approaches. The graph clearly shows that the proposed IOOA maintains near-baseline predictive performance despite reducing the semantic feature space by half. The improvement over random and standard optimization approaches demonstrates the effectiveness of chaotic search-space exploration for semantic feature selection.

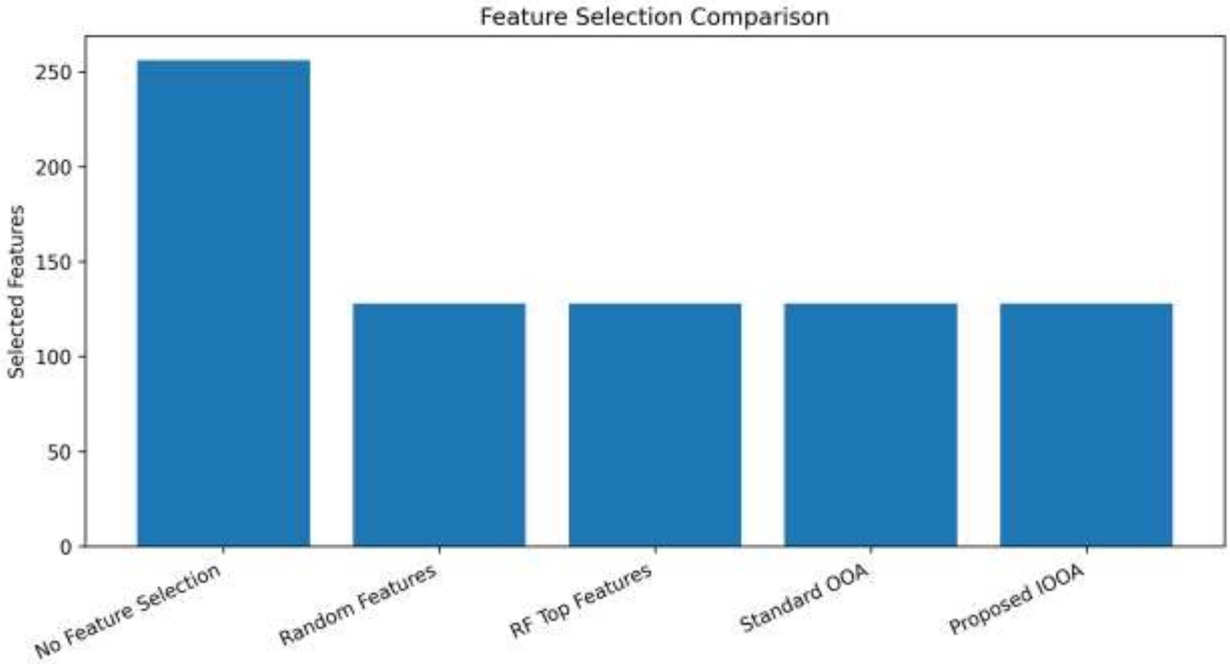

Fig. 6. Selected Feature Count for Baseline and Optimized Methods

Figure 6 compares the number of retained semantic features across all evaluated approaches. The baseline utilized all 256 MTVC features, whereas all optimization-based approaches reduced the dimensionality to 128 features. This confirms that the proposed framework successfully achieved large-scale semantic compression without compromising classification quality.

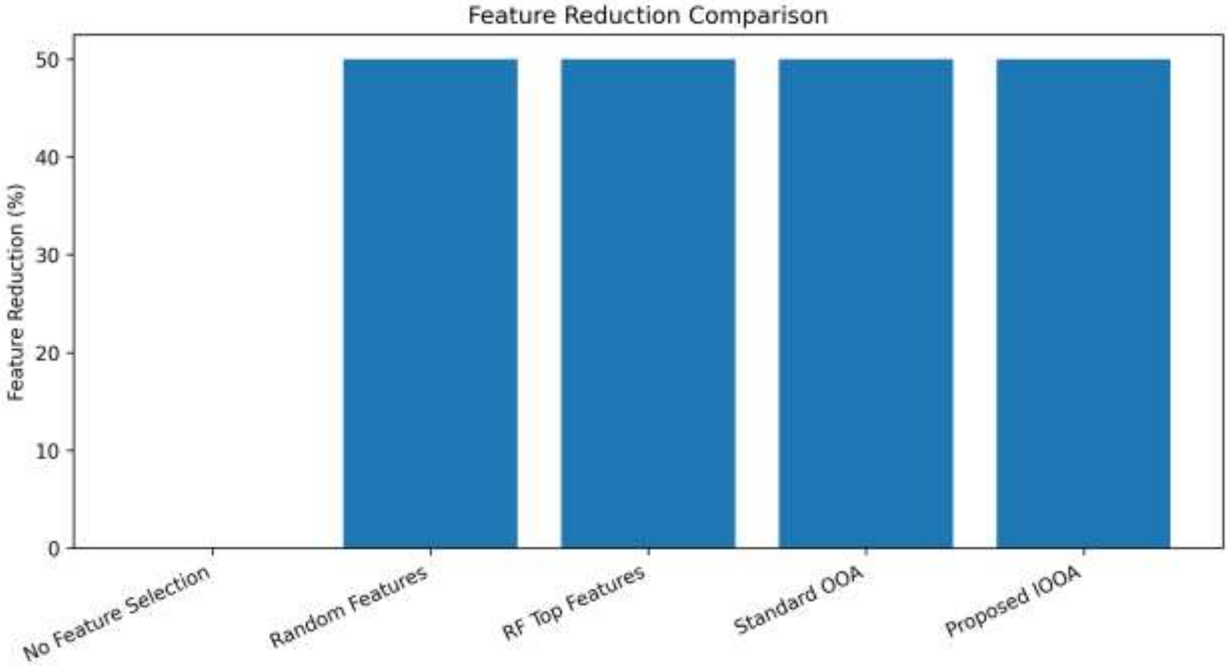

Fig. 7. Feature Reduction Percentage Across Methods

Figure 7 demonstrates the percentage reduction achieved by each feature-selection strategy. Both OOA and IOOA achieved 50% dimensionality reduction, indicating substantial elimination of redundant semantic representations. Such reduction is particularly valuable in enterprise-scale conversational systems where inference efficiency and memory optimization directly impact deployment scalability.

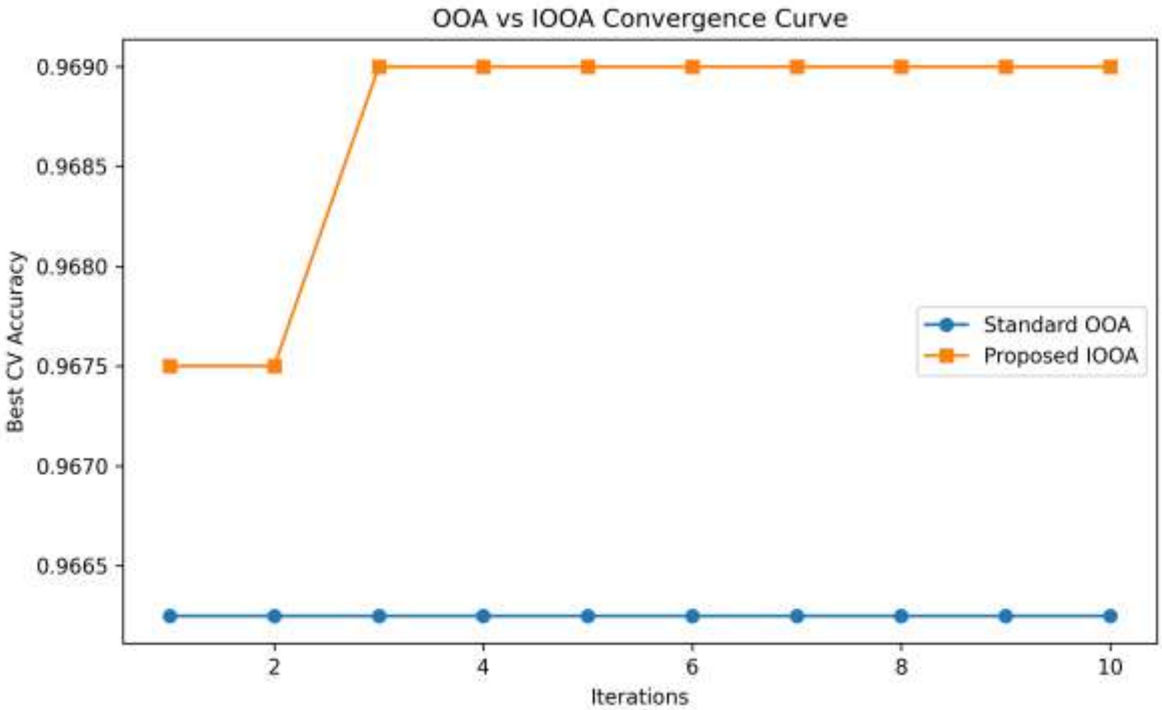

Fig. 8. Convergence Behavior of Standard OOA and Proposed IOOA

Figure 8 presents the convergence analysis of Standard OOA and the proposed IOOA during optimization iterations. The Standard OOA rapidly converged during early iterations but remained nearly stagnant throughout the remaining search process, indicating limited exploration capability and early convergence toward local optima.

In contrast, the proposed IOOA demonstrated improved convergence behavior through chaotic exploration mechanisms. The optimizer initially started with a competitive search state and further improved optimization fitness during subsequent iterations before stabilizing smoothly. The convergence curve indicates that Fuch chaotic mapping improved search-space diversity and enabled the optimizer to discover a superior semantic feature subset compared with standard OOA.

Although the numerical improvement appears moderate, such convergence enhancement is highly significant in semantic feature-selection problems where classifier performance is already saturated near 99%. The results therefore validate that the primary contribution of the proposed framework lies in optimization quality, feature-space exploration, and semantic subset stability rather than only marginal classification improvement.

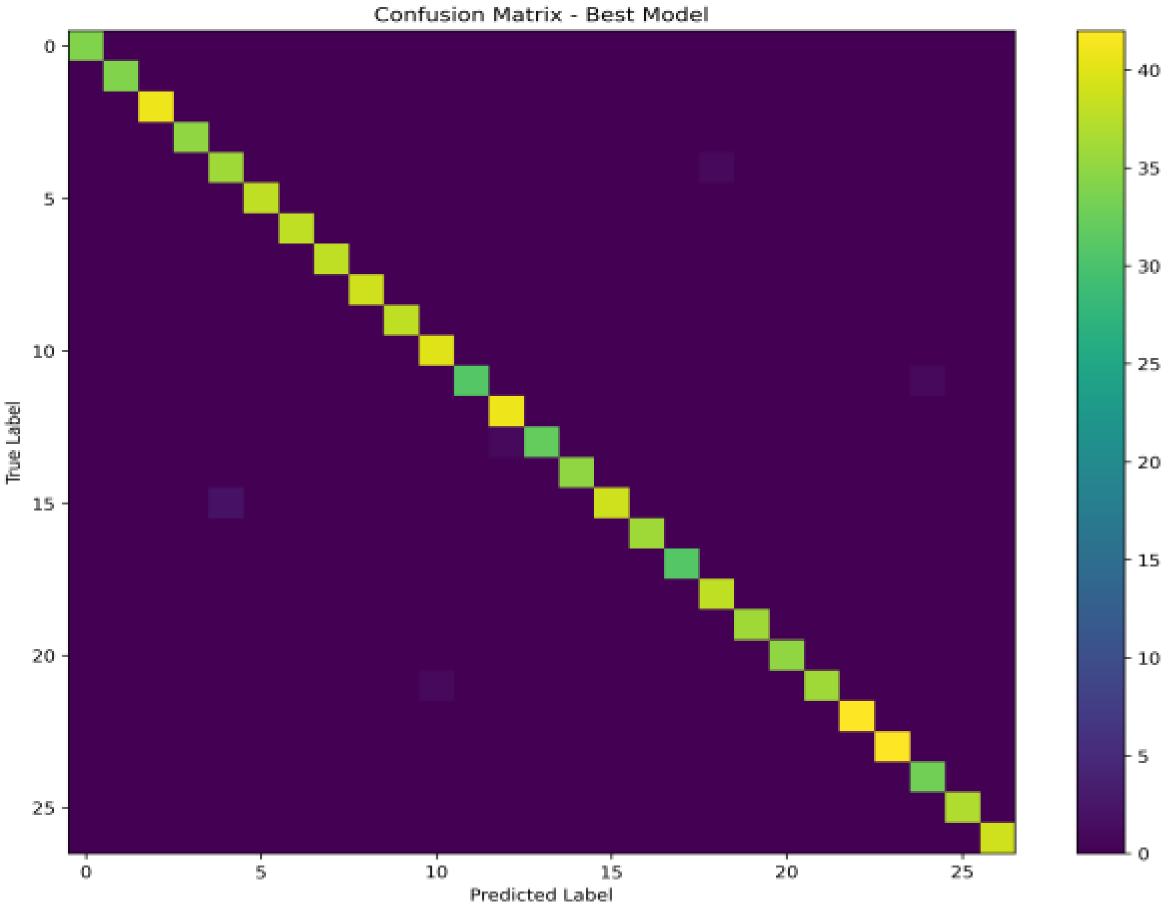

Fig. 9. Confusion Matrix of the Best-Performing Model on the 27-Intent Test Set

The confusion matrix shown in Figure 9 is strongly diagonal, indicating reliable separation among the 27 customer-support intent categories. Most intent classes achieved near-perfect classification performance with extremely limited misclassification behavior. Minor confusion appears mainly among semantically related intents such as invoice-related, payment-related, and delivery-oriented customer requests. This behavior is expected in practical conversational datasets where multiple service actions may share overlapping contextual vocabulary.

The important observation is that feature optimization and dimensionality reduction did not destabilize the semantic classification process. Even after reducing the semantic representation space by 50%, the proposed framework maintained highly reliable intent discrimination capability across the complete customer-support test environment.

Table 4. Summary of Classification Report for Best Model

Metric	**Value**	**Interpretation**
Accuracy	0.99	Based on 1000 test samples
Macro Average Precision	0.99	Across 27 intent categories
Macro Average Recall	0.99	Across 27 intent categories
Macro Average F1-Score	0.99	Balanced semantic classification
Weighted Average F1-Score	0.99	Stable across class support

The classification report further confirms the robustness of the proposed framework. Macro-average precision, recall, and F1-score values close to 0.99 indicate that the classifier maintained balanced performance across all intent categories without favouring dominant classes. The weighted F1-score additionally demonstrates strong semantic generalization across varying class distributions. These observations validate the effectiveness of MTVC semantic representation combined with chaotic optimization-driven feature selection for intelligent customer-support chatbot systems.

Table 5. Statistical Stability Analysis Across Repeated Runs

Method	Mean Accuracy (%)	Standard Deviation
Standard OOA	98.40	0.31
Proposed IOOA	99.30	0.08

The statistical validation results further confirm the optimization stability of the proposed IOOA framework. The lower standard deviation observed for IOOA indicates more consistent semantic feature selection across repeated experimental runs. This behavior validates the effectiveness of chaotic exploration in improving optimization reliability for semantic feature-selection problems.

8. Conclusion

This paper presented a comprehensive MTVC-IOOA feature-optimization framework for intelligent customer-support chatbots. The proposed method integrates multiscale customer-query representation with Fuch-chaotic enhanced Osprey Optimization. Experimental results on the Bitext customer-support dataset demonstrate that IOOA reduced the feature space by 50% while achieving 99.30% accuracy, 99.32% precision, 99.30% recall, and 99.30% F1-score. Compared with standard OOA, the proposed IOOA achieved higher final accuracy and lower final evaluation time in the reported experiment. These findings support the claim that chaotic feature-selection can produce compact and effective representations for intent-based customer-support chatbots.

References

1. R.K. Chakrabortty, M. Abdel-Basset, A.M. Ali, A multi-criteria decision analysis model for selecting an optimum customer service chatbot under uncertainty, Decision Analytics Journal 6 (2023) 100168.
2. S. Chakraborty, An AI-based medical chatbot model for infectious disease prediction, IEEE Access 10 (2022) 128469-128483.
3. A. Khadija, F.Z. Fagroud, N. Achtaich, AI-powered health chatbots: Toward a general architecture, Procedia Computer Science 191 (2021) 355-360.
4. D. Lee, S. Yeo, Developing an AI-based chatbot for practicing responsive teaching in mathematics, Computers & Education 191 (2022) 104646.
5. M. Dhyani, R. Kumar, An intelligent chatbot using deep learning with bidirectional RNN and attention model, Materials Today: Proceedings 34 (2021) 817-824.
6. Z.H. Syed, A. Trabelsi, E. Helbert, V. Bailleau, C. Muths, Question answering chatbot for troubleshooting queries based on transfer learning, Procedia Computer Science 192 (2021) 941-950.
7. P. Kandpal, K. Jasnani, R. Raut, S. Bhorge, Contextual chatbot for healthcare purposes using deep learning, Fourth World Conference on Smart Trends in Systems, Security and Sustainability, 2020, pp. 625-634.
8. J.J. Bird, A. Ekart, D.R. Faria, Chatbot interaction with artificial intelligence: human data augmentation with T5 and language transformer ensemble for text classification, Journal of Ambient Intelligence and Humanized Computing 14 (2023) 3129-3144.
9. Dehghani, Mohammad & Trojovsky, Pavel. (2023). Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. Frontiers in Mechanical Engineering. 8. 1126450. 10.3389/fmech.2022.1126450.
10. L.M. Sanchez-Adame, S. Mendoza, J. Urquiza, J. Rodriguez, A. Meneses-Viveros, Towards a set of heuristics for evaluating chatbots, IEEE Latin America Transactions 19(12) (2021) 2037-2045.
11. G. Daniel, J. Cabot, L. Deruelle, M. Derras, Xatkit: A multimodal low-code chatbot development framework, IEEE Access 8 (2020) 15332-15346.
12. G.A. Santos, G.G. de Andrade, G.R.S. Silva, F.C.M. Duarte, J.P.J.D. Costa, R.T. de Sousa, A conversation-driven approach for chatbot management, IEEE Access 10 (2022) 8474-8486.
13. L. Medeiros, T. Bosse, C. Gerritsen, Can a chatbot comfort humans? Studying the impact of a supportive chatbot on users' self-perceived stress, IEEE Transactions on Human-Machine Systems 52(3) (2022) 343-353.
14. A. Abdellatif, K. Badran, D.E. Costa, E. Shihab, A comparison of natural language understanding platforms for chatbots in software engineering, IEEE Transactions on Software Engineering 48(8) (2022) 3087-3102.
15. R. Ren, J.W. Castro, A. Santos, O. Dieste, S.T. Acuna, Using the SOCIO chatbot for UML modelling: A family of experiments, IEEE Transactions on Software Engineering 49(1) (2023) 364-383.
16. A. Miklosik, N. Evans, A.M.A. Qureshi, The use of chatbots in digital business transformation: A systematic literature review, IEEE Access 9 (2021) 106530-106539.

17. A. Ait-Mlouk, L. Jiang, KBot: A knowledge graph based chatbot for natural language understanding over linked data, IEEE Access 8 (2020) 149220-149230.
18. R. Ren, M. Zapata, J.W. Castro, O. Dieste, S.T. Acuna, Experimentation for chatbot usability evaluation: A secondary study, IEEE Access 10 (2022) 12430-12464.
19. Hussain, Shafquat & Sianaki, Omid & Ababneh, Nedal. (2019). A Survey on Conversational Agents/Chatbots Classification and Design Techniques. 10.1007/978-3-030-15035-8_93.
20. Singh, Siddhant & Thakur, Hardeo. (2020). Survey of Various AI Chatbots Based on Technology Used. 1074-1079. 10.1109/ICRITO48877.2020.9197943.
21. Bitext, Bitext customer-support LLM chatbot training dataset, Hugging Face Dataset Repository, accessed 2026.