International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 1081-1105
https://doi.org/10.70917/ijcisim-2026-3520
Article

Design and Development of an Efficient AI Chatbot for Customer Service Systems

Sachin Babulal Jadhav[1]*, B. Kshirsagar[2]

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

Abstract: A chatbot is a virtual character with interactive text capabilities that can successfully communicate with any human. Conversational chatbots, sometimes referred to as chatbots or dialogue systems, are computer programs created to mimic human-user dialogue, particularly on the Internet. These chatbots can be utilized in a variety of settings, including entertainment, information gathering, customer service, and instructional help. Hence, in this research proposed a novel AI chatbot for handling complex problems in comprehensive analysis of contextual response generation and intent classification for customer support systems. Initially, the data is are collected from dataset, and text data processed through the Multiscale Transformer Vector Conversion phase (MTVC) approach which used for converting the text data into vectorization form that reduce dimensionality issues. The feature extraction approach uses Improved Osprey Optimization Algorithm (IOOA) which is used to effectively optimize feature weights. Then, the classification model is called Dilated Deep Temporal Convolutional Networks with Novel Loss Function (DDTCN-NLF) which effectively utilized for intent classification, which optimizes decision boundaries while reduce computational complexity and increased accuracy. Finally, response generation method Hybrid Cross Attention assisted Recurrent Bidirectional Encoder Representations from Transformer (HCAR-BERT) which is used to enhance the authentication of context through cross attention and recurrent memory and this method provide answer of user query. Experimental findings of this research showed that the effectiveness of the proposed method which attained highest accuracy value of 98.55%, an F1-score of 98.23%, respectively. In this research indicate that the suggested approach has improve performance, efficiency and complex problem solving.

Keywords: Chatbot, Transformer, Osprey Optimization, Novel Loss Function, Cross Attention.

1. Introduction

For many years, the industrial segment has experienced tremendous revolution through the advent of the Industry 4.0 paradigm [1]. This process represent as smart factory that contribute on implementing advanced and intelligent technological procedures like Artificial Intelligence (AI) and Machine Learning (ML) to enhance production processes automation and effectiveness in education and customer queries [2-3]. Natural Language processing (NLP) and ML models have developed and offered AI chatbot methodologies to act as human conversational agents, reshaping standard communication channels for industries [4]. As chatbots collect significant information with user communications that can deliver insights into customer behaviour as preferences that can be applied to present business process [5]. Chatbots can be categorized into different class like Intent and Rule based chatbots [6].

Here, rule-based process determine the chatbots reaction based on particular conditions or rule, especially applied for narrowly defined conversation protocols. Although, rule-based chatbots can't adequately response beyond protocol rules and conditions [7-8]. In contrast, Intent-based process depend on chatbot capability to learn and collect data. This process might be trained in NLP process by using different datasets that include conversation dialogs to extract interaction of conversation process that contains entity, intent and context [9]. Nowadays, ChatGPT is a vital AI chatbot creation, which is based on Large Language Model (LLM) that utilize transformer structure of generative pre-trained transformers (GPT) [10-11].

This process provide more benefits of chatbots, which present counselling services more accessible through eliminating location and time constraints [12]. Currently, recent research introduced hybrid chatbot methodology for addressing limitation with integrating rule, retrieval and generative strategies. Here, Multinomial naïve bayes classifier to intelligently route user queries to most appropriate component, providing effective performances [13]. Moreover, the mixed integer linear programming and meta-heuristic are used to increase significant in decision-making from textual information's [14]. Further, Neural Machine Translation (NMT) model that is enhancement on sequence-to-sequence process that provide vital improvement over the standard models. In the meantime, generative chatbots powered with advanced language process, which provide flexibility but often offer inaccurate or contextually irrelevant reaction [15].

1.1 Motivation

In today's environment, business is continuously seeking novel solutions to improve customer approval, constancy and practices. One vital techniques in this realm is usage of Artificial Intelligence (AI)-powered chatbots. These innovative approach have converted customer service with permitting actual relations and prompt personalized. The development of chatbots from essential automated responders to refined virtual assistants that is driven with development in ML, big analyst and NLP. These technologies have collectively empowered chatbots to provide unprecedented condition of customer assistance and engagements. Different application focused to growing reliance on AI-powered chatbots in customer services. The increasing demand for 24/7 customer assistance have concentrated standard customer service model inadequate. Currently, AI based chatbots deliver scalable solution and able of providing round-the-clock support without human interaction, therefore meeting customer belief for instant reactions. Additionally, AI based methodologies enable chatbots to offer very personalized practices. By considering customer information, chatbots can alter reaction based on own inclination, behavior strategies, fostering constancy and purchase history. However, maintaining and emerging refined AI based chatbot essential require significant cost and technical expertise. Ongoing improvement in AI methodologies and adoption rates, these challenges are motivated, paving the pathway for further widespread and efficiency utilize of AI chatbot in customer services. This research aims to explore the impact of AI-powered chatbot seeks for delivering the deeper understanding of trends and patterns in this developing arenas. The findings of this research will be focus to literature body of knowledge through highlighting key factors influencing efficiency of AI chatbots in complex problems.

1.2 Objectives

- To develops AI-based chatbot system for resolving the complex problem in customer related queries and improving response performances.
- To introduce Multiscale Transformer Vector Conversion phase (MTVC) mechanism for converting the text data into vectorization form that reduce dimensionality issues.
- To present Improved Osprey Optimization Algorithm (IOOA) model for selecting relevant features, which can decrease computational complexity.
- To develops Dilated Deep Temporal Convolutional Networks with Novel Loss Function (DDTCN-NLF) model for intent classification and present more realistic results.

This research paper is structured as follows: Section 1 contains introduction, which shows the basic information, section 2 and 3 demonstrates the related work and an elaborate overview of the proposed architecture. Section 4 illustrates result and discussion. Entire research work is concluded in section 5 by including the future scopes.

2. Related Works

Babayigit et al. [16] suggested TrBot for Turkish chatbot. Initially the gathered data will be pre-processed by removing special characters, lower casing, and tokenization. The TrBot model is used stacked LSTM layer to encode the input context vector to output features. Also used attention mechanisms for extract the high level features and long range dependency. For experimental evaluation model used QA and dialog datasets, the TrBot model attained the high accuracy of 80% at QA dataset. The model had scalability challenges this reduce the model efficiency.

Mikael et al. [17] presented Naïve Bayes for Chatbot Administrative Support. The collected data will pre-processing by text cleaning which remove the unnecessary elements and white space. The model utilized AIML (Artificial Intelligence Markup Language) mechanism it will compare input patterns with user queries. In the result

analysis the Naïve Bayes model attained the accuracy of 97.57%. The model had generalizability issues which may decrease the real-time performance.

Kathole et al. [18] presented AA-DTCN-SC model for educational chatbot system. Hybrid Averaging-based Driving Training-Barnacles Mating Optimizer (ADT-BMO) optimization algorithm was applied to enhance global exploration and local exploitation. The model was utilized three NLP models like Text Convolutional Neural Network (CNN), Bidirectional Encoder Representations from Transformer (BERT), and TransformerNet. The AA-DTCN-SC model attained the high accuracy of 92.68%. The model had computationally complexity which may raise the processing time.

Nageshwaran et al. [19] analysed Hybrid Convolutional Long Short-Term Memory (HCLSTM) for education Chatbot. Initially the collected data were pre-processed by tokenization, stop words removal, lemmatization, and stemming. The model utilized CNN layer extracting significant patterns from the images and LSTM layer capture the long-term dependency. The model attained the accuracy of 97%. Scalability challenges may arise to handling large numbers users.

Taiwo et al. [20] presented psychological first aid generative pre-trained transformer (PFA-GPT) for mental health support Chatbot. The PFA-GPT model utilizing BERT for detect the distress emotional detection. The experimental validation PFA-GPT model attained the accuracy rate of 93%. The limitation of this model was low efficiency in training. The overview of the existing methods are given in the Table 1.

Table 1: Overview of existing methods with their limitations

Authors & References	Year	Method	Performance	Limitations
Babayigit et al. [16]	2025	TrBot	Accuracy of 80%	The model had scalability challenges this reduce the model efficiency.
Mikael et al. [17]	2025	Naïve Bayes	Accuracy of 97.57%	The model had generalizability issues which may decrease the real-time performance.
Kathole et al. [18]	2025	AA-DTCN-SC	Accuracy of 92.68%	The model had computationally complexity which may raise the processing time.
Nageshwaran et al. [19]	2025	HCLSTM	Accuracy of 97%	Scalability challenges may arise to handling large numbers users.
Taiwo et al. [20]	2025	PFA-GPT	Accuracy of 93%	The limitation of this model was low efficiency in training.

Table 1: Overview of existing methods with their limitations

2.1 Problem Statement

The existing AI chatbot system faced complex problems such as, Babayigit et al. [16] suggested a TrBot model, however this research faced scalability challenge and reduced the efficiency of the model. Similarly, Naïve Bayes was developed by Mikael et al. [17], however model had generalizability issues which may decrease the real time performance. Likewise, Kathole et al. [18] introduced an AA-DTCN-SC model, nevertheless the model had computational complexity which may raise the processing time. Similarly, HCLSTM model was suggested by Nageshwaran et al. [19], however, this system faced scalability issues, it raised to handling large number of users. Finally, PFA-GPT was developed by Taiwo et al. [20], nevertheless limitation of this model was low efficiency in training. However, to overcome these limitations, this proposed research utilized a DDTCN-NLF model and HCAR-BERT for developing efficient chatbot system.

3. Proposed methodology

The rapid evolution of conversational AI and its integration into automated customer support systems has enabled seamless real-time interaction, intelligent query resolution and efficient service delivery across seamless real-time interaction, intelligent query resolution and efficient service delivery across diverse domains such as e-commerce, banking and corporate helpdesks. However, these systems continue to face problems with maintaining contextual accuracy, handling high-dimensional linguistic features, and resolving uncertain user intent in complex domain-specific dialogues. Traditional chatbot architectures typically optimize either keyword matching or basic intent classification, but rarely address feature optimization and deep temporal dependencies in a unified and adaptive

manner. To address this gap, this paper proposes an AI-based chatbot system utilizing optimized feature selection and advanced transformer-based response generation. Initially, the AI-based chatbot system is initialized with customer support datasets and domain-specific parameters to establish a baseline for conversional context. Figure 1 demonstrates the overall workflow diagram.

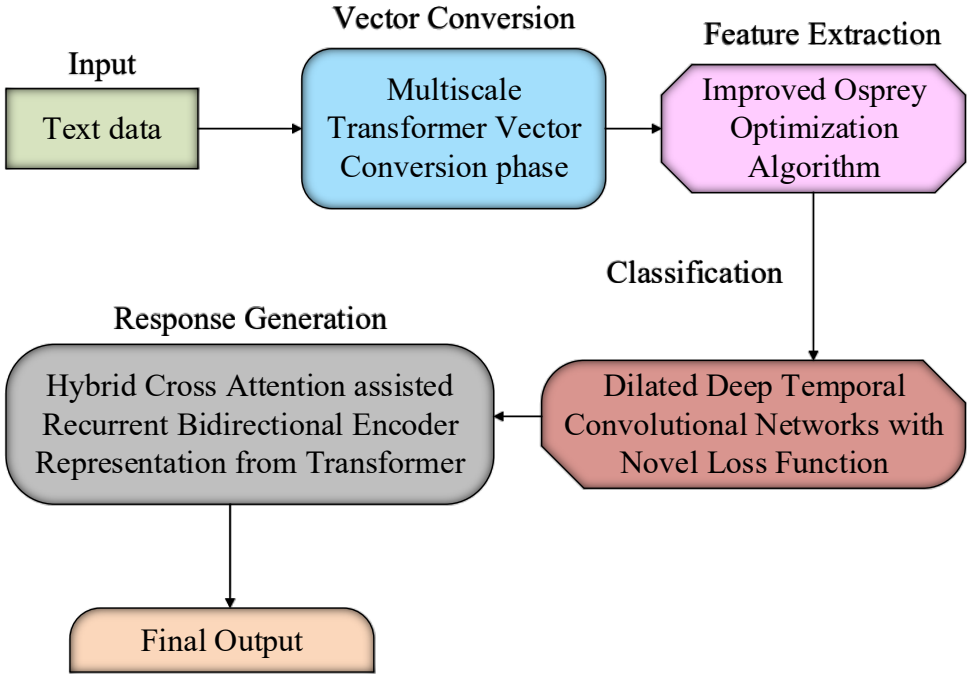


Figure 1: Complete workflow diagram of proposed methodology

Then, the raw textual data is processed through the Multiscale Transformer Vector Conversion phase, where the text is converted into dense vector forms to reduce the numerical representation of linguistic meaning. For these vectors, features such as semantic weight and contextual relevance are computed. Such values are aggregated into an optimization objective and the Improved Osprey Optimization Algorithm is used to determine the most salient optimal weighted features. After feature selection, dilated Deep Temporal Convolutional Networks with Novel Loss Function are employed to classify user intents across long-range text dependencies. After Intent classification, accurate response generation is guaranteed by the HCAR-BERT model, which ensures contextual alignments and immunity against uncertain information. Finally, performance is evaluated using Type I and Type II measures to ensure scalability and precision under real-world interactions.

3.1 Data Vectorization using MTVC

This paper introduces a Multiscale Transformer Vector Conversion to convert the text data into dense vector forms to reduce the numerical representation of meaning of the text. The multiscale format is obtained by extending the single scale format. The diagrammatic representation of the MTVC model is shown in figure 2.

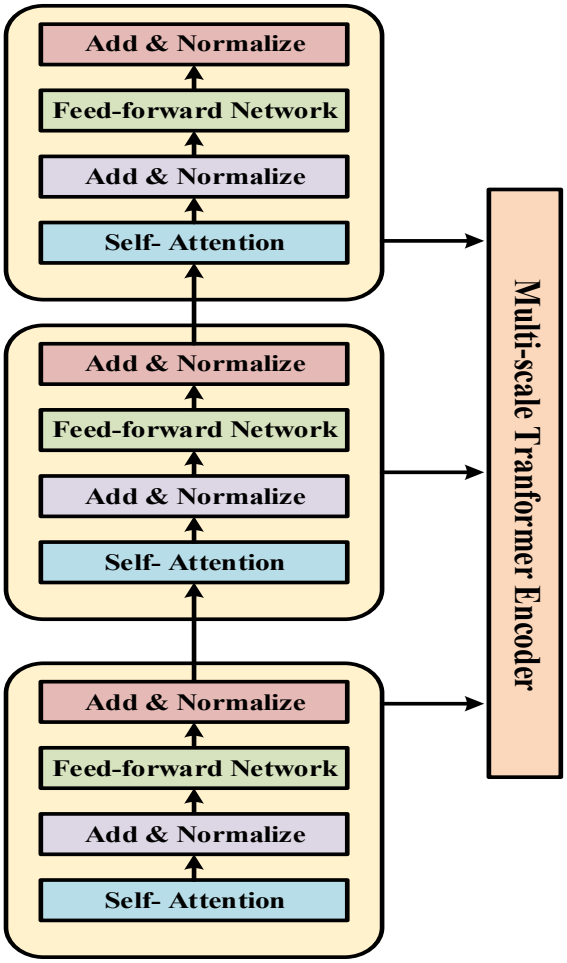


Figure 2: Structure of MTVC

In this multi-scale vector conversion, the index number of the single scale transformer is denoted by, m which ranges from 1 to R. Let assume the word as $w_{i.j}$ The output of the word from m single-scale transformer unit is represented as $ST_{i,j}^{(m)}$ and the output from multiscale transformer operator is given by the format of R representative vectors. The encoding representation can be represented as the following equation:

$$MTVC_{i,j} = \frac{1}{R}\sum_{r=1}^{R} \lambda_m . ST_{i,j}^{(m)} \tag{1}$$

In which, the represents the weight parameter for. Equation (1) shows the aggregation of deep representation from the single transformer units into a multiscale transformer operator. The word sequence for the piece of news is represented by the following equation:

$$S_i = \left\lfloor MTVC_{i,1} \oplus ... \oplus MTVC_{i,j} \oplus ... \oplus MTVC_{i,J} \right\rfloor \tag{2}$$

Here, the represents the final encoding representation of the word by Multiscale Transformer operator. The term denotes the vector form of data that helps to reduce the numerical representation of the text.

3.2 Feature Refinement through IOOA

This paper employs an Improved Osprey Optimization Algorithm, to improve the computational efficiency and remove noise of the vectorized data from MTVC. The IOOA [21] is the advanced optimization model which deploys chaotic mapping along with Optimization algorithm. The population is initialized using random initialization in OOA that causes uneven distribution of population initialization or reduction in the initialized population diversity. The chaotic mapping enhances algorithm performance by increasing initial population diversity and strengthening global search capabilities, ultimately leading to higher quality solutions. This paper integrates the Fuch chaotic mapping to initialize the osprey population. Flowchart of IOOA optimization are shown in figure 3.

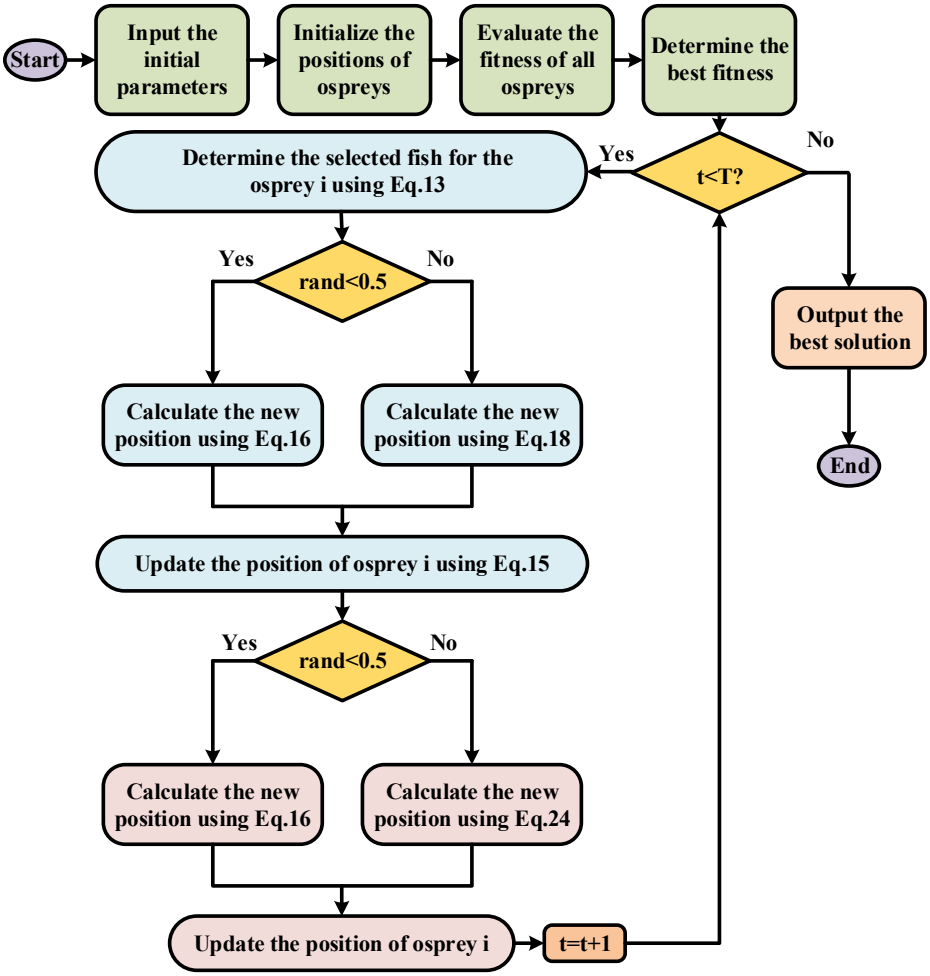


Figure 3: Flowchart of IOOA

This is infinitely collapsible chaotic mapping, which shows advantages like stronger chaotic properties and more balanced traversal than the traditional chaotic mapping and its chaotic sequence. The mathematical expression of Fuch chaotic mapping is,

$$h_{i+1} = \cos\left(\frac{1}{H_i^2}\right) XX) \tag{3}$$

Where, , and . The chaotic variables are generated using the equation (3). The OOA is a bio-inspired metaheuristic based on the hunting behaviour of ospreys, including the two-phase process of exploration and exploitation. The initialization of the osprey population in the search space is given by the equation:

$$Y_{i,j} = l_j + r_{i,j} \cdot \left(u_j - l_j\right), \quad i = 1,2,\ldots,N, \quad j = 1,2,\ldots,m \tag{4}$$

Where, the initial position of the th osprey in the th dimension is represented as, the and represents the upper and lower bounds of the th problem variable; is the random number in the range of [0, 1]; the osprey population size is given by and is the dimension of the problem. After adding chaotic variables the initialization formula is given by,

$$Y_{i,j} = l_j + h_i \cdot \left(u_j - l_j\right), \quad i = 1,2,\ldots,N, \quad j = 1,2,\ldots,m \tag{5}$$

The fitness value of the osprey is calculated using the function shown in equation (6).

$$FV(Y) = FV(Y_{1,}Y_{2,}\ldots,Y_N) \tag{6}$$

Where, the fitness value of the th osprey is denoted asand their position is given by. In this paper the minimum value ofhas to be solved. The smaller fitness value is better for the location of the osprey.

Then the osprey enters the exploration phase or global exploration phase, after the population initialization. Other ospreys' positions with superior fitness values were selected as potential fish locations. The school of fish for each osprey is expressed as:

$$FL = \left\{ Y_z \middle| z \in \left\{ 1,2,....,N \right\} \cap FV_l < \left\{ FV_1, FV_2,..., FV_N \right\} \right\} \cup \left\{ Y_{best} \right\} \tag{7}$$

Where FL is the set of fish locations of the th osprey and is the ospreys location with best fitness value. The osprey randomly selects and attacks a fish in the search space. The new ospreys' position during the movement of the fish towards the fish is calculated using equations (8) and (9).

$$Y_{i,j}^{S1} = Y_{i,j} + h_i \cdot \left(FS_{i,j} - K_{i,j} \cdot Y_{i,j} \right), i = 1,2,...., N; j = 1,2,..., m \tag{8}$$

$$Y_{i,j}^{S1} = \begin{cases} Y_{i,j}^{S1}, l_j \le Y_{i,j}^{S1} \le u_j \\ l_j, Y_{i,j}^{S1} < l_j \\ u_j, Y_{i,j}^{S1} > u_j \end{cases} \tag{9}$$

Here, is the new position of the th osprey in theth dimension in the first stage; denotes the selected fish by the th osprey in the th dimension; is the random integer either 1 or 2. When the fitness value of the new position is better than the original fitness value, then the original position is replaced with the new position. If its value is lower, it does not replaces. This process is shown in equation (10):

$$Y_i = \begin{cases} Y_i^{S1}, FV_i^{S1} < FV_i \\ Y_i, FV_i^{S1} \ge FV_i \end{cases} \tag{10}$$

where is the new position of the th osprey after updating the first stage; is the fitness value of the new position of the th osprey after updating the first stage. In nature, soon after attacking a fish, the osprey will take it to a safe place and feed on it. This process in this paper is shown by equation (11) and (12), in which the new random position is calculated as the feeding position.

$$(Y_{i,j}^{S2} = Y_{i,j} + \frac{l_j + h_i \cdot \left(u_j - l_j \right)}{t}, i = 1,2,..., N; j = 1,2,..., m; t = 1,2,..., T \tag{11}$$

$$Y_{i,j}^{S2} = \begin{cases} Y_{i,j}^{S2}, l_j \le Y_{i,j}^{S2} \le u_j \\ l_j, Y_{i,j}^{S2} < l_j \\ u_j, Y_{i,j}^{S2} > u_j \end{cases} \tag{12}$$

Where is the new position of the th osprey in theth dimension in the second stage; is the current number of iterations and denotes the maximum number of iterations.

When the new positions' fitness value is better than the original fitness value, then the original position is replaced with the new position. If its value is lower, then it does not replaces. This replacement process is shown in equation (13):

$$Y_i = \begin{cases} Y_i^{S2}, FV_i^{S2} < FV_i \\ Y_i, FV_i^{S2} \ge FV_i \end{cases}$$

where, is the new position of the th osprey after updating the second stage; is the fitness value of the new position of the th osprey after updating the second stage.

3.3 Intent Classification using DDTCN-NLF

The optimized features from the IOOA is fed into the Dilated Deep Temporal Convolutional Network to handle long-range dependencies in dialog with a specialized loss function. This intent classification phase acts as the major decision making component of the chatbot which are responsible for mapping optimized user query features to specific functional categories. In this section the classification is performed using Dilated Deep Temporal Convolutional Networks integrated with Novel loss function based on Triplet Loss.

3.3.1 Dilated Convolution Layer

This paper used dilated convolution instead of normal convolutional layer to enhance the receptive field .Dilated convolutions is a powerful technique in deep learning designed to expand a neural networks effective receptive field. The traditional methods include CNN, in which the data is convolved and then pooled. Here, the pooling layer is used for increasing the receptive field and reducing the amount of calculation. Figure 4 illustrates the comparative structure of traditional and dilated convolutional layer.

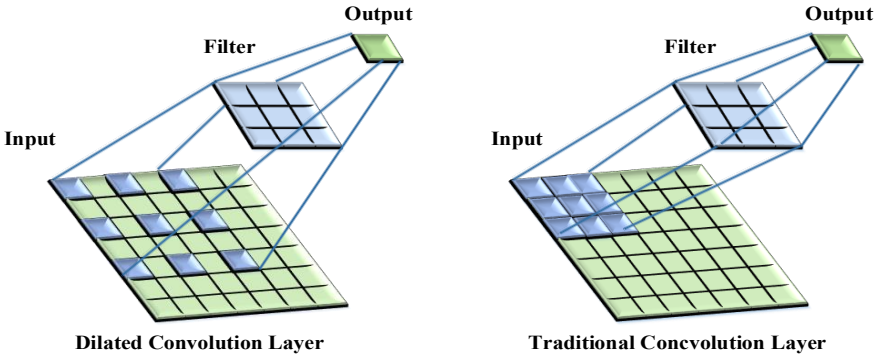


Figure 4: Comparative structure of Traditional and Dilated Convolutional Layer

The receptive field is the part of the data which is defined by the filter size of the layer in the CNN. The required features are extracted using the filter. The definition of the receptive field is shown in equation (14).

$$R_f = D(x-1)+1 \tag{14}$$

Here, denotes the size of the kernel; is the dilation rate, which is the space between each pixel in the convolution filter. In order to increase the feature resolution, quality of the training result and reduce computational costs, this paper integrates the dilation rate larger than one to the conv2D kernel through dilated convolution to the receptive field. The filter can grab more contextual information with an increase in receptive field . The output size can be calculated by using the equation (15):

$$\sigma = \left| \frac{a + 2P - R_f}{S} \right| + 1 \tag{15}$$

Where is the input including factors like dilation factor, padding and stride. Integrating the output of convolutions with different dilation rates becomes easier as the output features have same sizes. By utilizing diverse receptive fields of different sizes the features can be extracted effectively across various scales. Therefore, dilated convolutions are superior for exponentially widening the field of view without sacrificing image resolution or neglecting spatial convergence.

3.3.2 Deep Temporal Convolutional Network

The Deep Temporal Convolution network (DTCN) structure is developed by using many residual units. Each residual unit consists [22] of a Batch Normalization layer, convolutional layer, dropout layer and residual units. Figure 5 demonstrates the architecture of DDTCN diagram.

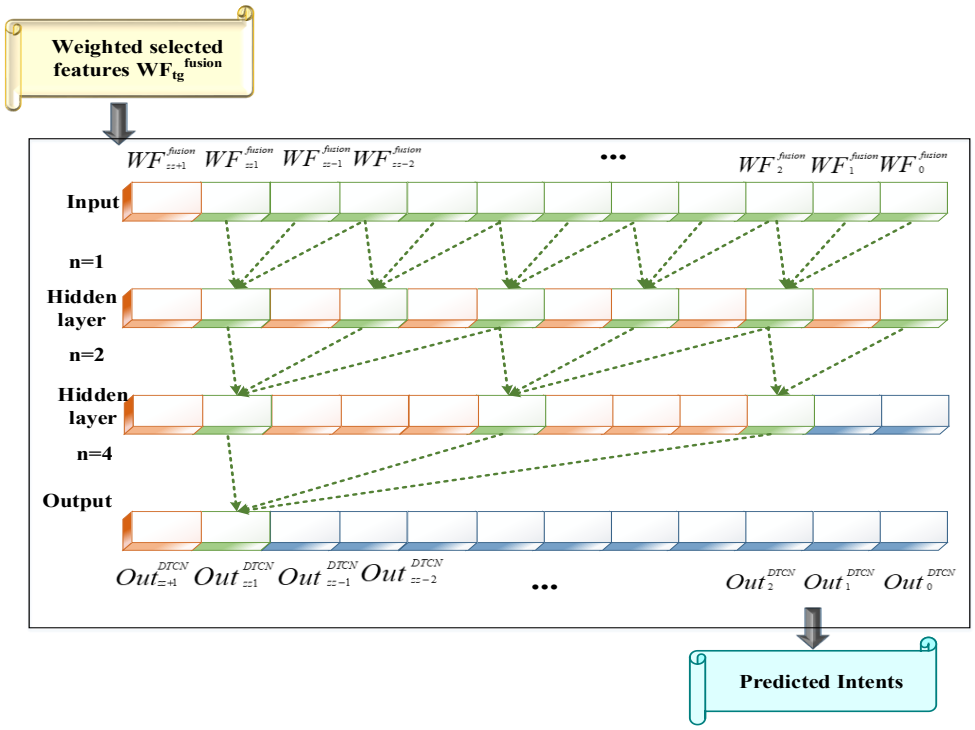


Figure 5: Architecture of DDTCN

The dilation factor can be given as 1, 2 and 4. The holes are inserted into kernels of the convolutional layer in the dilated convolutional layer. The dilation factors represents the holes within the neighbouring convolutional layers. The ordinary convolutional function is indicated by the dilation factor of value one. Then the other numbers of the dilation factors indicates the interior interval of the kernels in the convolutional layers. The kernel's size of the dilated convolutional layer is determined by using the equation (16).

$$x = rr.(KS - 1) + 1 \tag{16}$$

Where, is the kernel size of the dilated convolutional layer. The evaluation of the reception field is shown in equation (17).

$$R_f = \left(\sum_{R}^{tt} rr.(x - 1 \right) + 1 \tag{17}$$

The term in equation (17) represents the Reception field; R represents an array with rate of dilation and tt is the total count of the dilated convolutional layers. The reception field range is enhanced by increasing the rate of dilation or by increasing the kernel size. The use of normal convolution process makes the output solely dependent on the present value and the value from the former layer. This process can be mathematically represented as:

$$O_{zz}^{tt+1} = C(u_1^{tt}, u_2^{tt}, ..., u_{zz}^{tt}) \tag{18}$$

In the above equation (18), the term zz denotes the time; C denotes the convolution operation and represents the current output of the layer. The convolution operation does not depend on the upcoming time. By utilizing the residual connection the first and last layer of the residual units are connected, which assists in training the model effectively. Equation (19) represents the residual connection between the input and the output of the last residual units.

$$w = \sigma(U + V(U)) \tag{19}$$

Here, denotes the sigmoid activation function; is the last residual unit's output; denotes the residual unit's input; represents the residual unit's outcome. The input directly connects directly with the end residual units by skipping all the intermediate residual units. This helps to extract any dimensional attributes. This also helps to eliminate the disappearing gradient and explosion of the gradient issues and prevents the integrity of the provided information.

3.3.3 Triplet Loss Function

In this system, the triplet loss function is used as a novel loss function. This function was introduced to overcome the concept of [23] Siamese network for optimal classification. The architecture representation of the triplet loss function is given in figure 6.

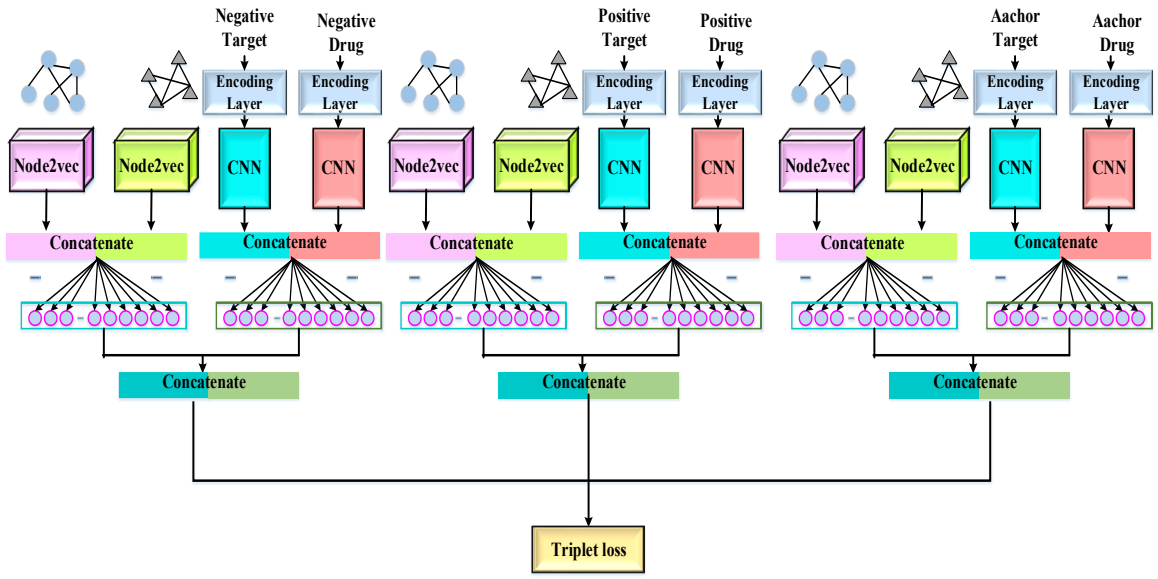


Figure 6: Architecture of Triplet Loss Function

The triplet loss function is a distance-based loss mechanism which trains a model for mapping data into multidimensional space where related items are clustered together and unrelated items are pushed away. This method operates by comparing three inputs simultaneously. The three inputs are taken as Anchor (), Positive () and Negative (), in which the Anchor represents the input user query, Positive represents the sample from same intent class and Negative denotes the sample from different intent class. The triplet loss is represented in equation (20):

$$\ell_T = \sum_{(x_a, x_p, x_n)} \max\left(0, f(x_a, x_p) - f(x_a, x_n) + \alpha\right) \quad (20)$$

Here, denotes the models output; is the parameter indicating distance between clusters. The triplet network will learn to achieve smaller distance between anchor-positive distance and anchor negative distance. This ensures considering multiple similar concepts than depending on one similar concept. This approach ensures high precision even when dealing with uncertain information, providing a robust intent label that guides the subsequent response generation phase.

3.4 Response Generation with HCAR-BERT

In this section the classified intent labels and the original input queries are integrated into the HCAR-BERT model for the final response generation. This HCAR-BERT model represents a sophisticated evolution of the standard BERT architecture, designed to handle the complexities of contextual dialogue and uncertain user queries in customer support systems.

3.4.1. Bidirectional Encoder Representation from Transformer

This model uses BERT to extract deep semantic features from the input query. This model acts as a base that understands the raw meaning of the customer's words. It is a pre-trained language representation model based on the evolving transformer architecture. The general purpose of BERT model is generating a "language understanding" model. By evaluating the context-appropriate words, the BERT model [24] generates numerical representations. The numerical vectors of words of similar meanings are aligned closely and the vectors for different meanings of same words are less similar. Both left and right contexts are considered simultaneously by BERT's architecture that

contributes to the effectiveness of interpreting meanings of the words even in complex language processing task. The BERT model consists of 12 layers of transformer structure. The architecture of the BERT model is presented in Figure 7.

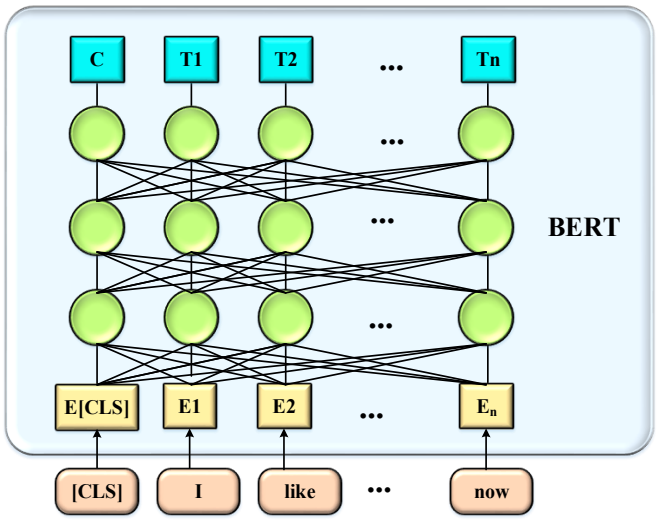


Figure 7: Structure of BERT model

First the BERT is trained through a large unlabelled text corpus. Then the model is refined for specific task called fine-tuning. BERT consists of 30000 tokens from WordPiece embeddings. A word which is not available in the vocabulary is divided into sub-words that exist in the vocabulary. The sub-words can even be a single character. The first 1000 tokens are reserved and the other tokens except the five tokens are in the form "[unused2]". The five tokens are:

- PAD: It is a token used for padding, Example: in a sequence of different sizes.
- UNK: "Unknown" token used to represent a token which is not in the vocabulary.
- CLS: "Classifier" token used in sequence classification instead of token classification. In the sequence of tokens, it is the first token.
- SEP: "Seperator" token, used to construct a sequence from a sub-sequences. Example, two sequences for sentence classification or question answering tasks. It is used as the last token in the sequence.
- MASK: This token is used to mask values and to replace the word that the model will attempt to predict.

The BERT training is classified into two stages:

i. Pre-Training: BERT is a pretrained model using two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM the percentage of the input tokens are selected at random and replaced with a MASK token. This task helps to predict the masked tokens. The NSP receives the sentences A and B to predict if the sentence B is next to A. This type of training is more important in order to understand the relationship between two sentences, which is important in tasks like Question Answering (QA) and Natural Language Inference (NLI).

ii. Fine-Tuning: It uses a trained model, which has been trained in a large text corpus and carries out training using domain application texts and labels. As compared with Pre-Training stage the Fine-Tuning is low cost.

The aspect extraction using contextual word embedding from BERT model shows better results for single-domain aspect extraction. The BERT model provides an effective classification of whether the given aspect belongs to a specific domain.

In order to handle the complexities of uncertain or evolving dialogue the Recurrent Neural Network (RNN) layer is introduced in this system. This maintains a temporal memory of the conversation flow which allows to generate

realistic and human-like responses. The diagrammatic representation of the structure of RNN can be shown in Figure 8.

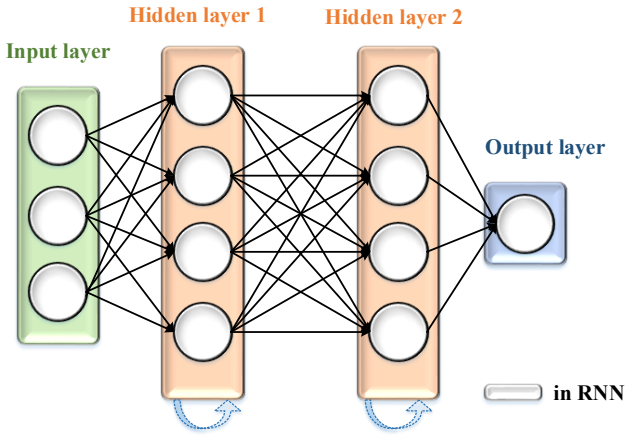


Figure 8: Structure of RNN

The algorithm steps of RNN can be described as follows:

Step 1: The input layer of the RNN receives the input sequence that contains a continuous word embeddings representing the words in a sentence using equation 21.

$$X_i = TEL(X_i - 1) \tag{21}$$

Where, in which is the input embedding and is the last hidden layer represents the sentence.

Step 2: The hidden layer of RNN captures the context of the input sequence. By placing a hidden state vector which is modified at every state based on the input data and the past hidden state. The hidden layer is represented by the following equations:

$$U_t = V_R Z_t + K_R h_{t-1} + b_R R_t \tag{22}$$

Where, and represents the update and reset gates, respectively. They are responsible for deciding the quantity of the previous hidden state that remains same and utilizing the current input for modifying the current hidden state. and are acquired throughout training and b denoted the bias vector.

Step 3: The output layer of RNN takes the final hidden state vector as input and produces the output score.

This architecture utilizes a Hybrid Cross Attention mechanism (CAM) to align the high-level intent features derived from the Intent classification [25] phase with the linguistic fine grained details of the query to ensure that the generated answer is contextually relevant and grounded in the identified user goal. Figure 9 shows the structure of Cross Attention Mechanism.

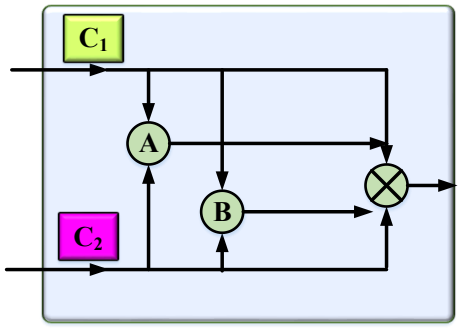


Figure 9: Structure of CAM

The novel cross attention mechanism is divided into three parts. The first part is an attention mechanism to generate weight features in horizontal and vertical directions. Then the second part is a multiplier to multiply two types of weight features to further enlarge the weight coefficients. The final part is used for matching the two types of weight features to obtain the maximum value that is used to supplement the weight coefficient results of the second part. The following equations shows the splicing and fusing of the weights of the three parts of the model.

$$Aw_i = \sum_{j=1}^{n} \frac{\exp(e_{i,j})}{\sum_{k=1}^{n} \exp(e_{ik})} h_j \tag{21}$$

$$mul = (Aw_I * Aw_{II}) \tag{22}$$

$$\max = (Aw_I, Aw_{II}) \tag{23}$$

$$CAM = CONC([Aw_I, Aw_{II}, mul, \max]) \tag{24}$$

Here, represents the weight coefficient assigned by the attention mechanism; is the feature sequence and is the feature of the moment. is the hidden layer information of feature sequence. The weight coefficient of the vertical attention mechanism in a feature sequence is represented by and the weight coefficients of the horizontal attention mechanism in a feature sequence is represented by. The multiplication of the weight coefficient is represented by and the maximum operation of the weight coefficient is given by. In this approach used to classified intents. Algorithm 1 demonstrates the overall proposed approach algorithm.

Algorithm 1: Overall proposed methodology algorithm

Start
Input: Raw Text data
Output: User output (accurate & contextual response)
Step 1: Vector Conversion
For all data in dataset do
MTVC
End for
Step 2: Feature extraction
For all converted data do
IOOA
End for
Step 3: Classification
For all extracted features do
DDTCN-NLF
End for
Step 4: Response Generation
For all classified images do
HCAR-BERT
End for
end

4. Experimental Result

The implementation of this proposed method is carried out in python. In this section the developed AI-based chatbots system using deep learning will be compared to existing models with two types of performance measures. This evaluation included type I positive measures and type II negative measures. The type I measures are evaluated with existing models like Inception V3, Long Short Term Memory-RNN (LSTM-RNN), Gated Recurrent Unit-CNN (GRU-CNN), Deep Neural Network (DNN). The system configuration used for implementing the proposed model is given in Table 2 and the Hyperparameter details are given in Table 3.

Table 2: System Configuration

Components	Details
Processor	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz 3.40 GHz
Installed RAM	16.0GB (15.9 GB usable)
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Table 3: Hyperparameter Details

Category	Hyperparameter	Value / Setting
Input	Input Length	512
Input	Input Channels	1
Model Architecture	Initial Conv Filters	64
Model Architecture	Kernel Size	3
Model Architecture	Padding	'same'
Residual Blocks	Number of Blocks	5
Residual Blocks	Dilation Rates	[1, 2, 4, 8, 16]
Residual Blocks	Filters per Block	64
Residual Blocks	Activation	ReLU
Residual Blocks	Normalization	Batch Normalization
Embedding Layer	Dense Units	128
Embedding Layer	Normalization	L2 Normalization
Classifier	Output Classes	27
Classifier	Activation	Softmax
Loss Functions	Classification Loss	Sparse Categorical Cross entropy
Loss Functions	Embedding Loss	Triplet Loss
Loss Functions	Triplet Margin	0.3
Loss Functions	Loss Weights	{1.0, 0.5}
Optimizer	Optimizer	Adam
Optimizer	Learning Rate	0.001 (1e-3)
Training	Epochs	300
Training	Batch Size	32
Pooling	Global Pooling	GlobalAveragePooling1D
Output	Outputs	Classifier + Embedding

4.1 Dataset Description

In the year 2023, Bitext Innovations released a synthetic dataset of 26,872 customer service QA pairs designed to train LLM-based [26] virtual agents. The collection covers 27 intents such as password resets and order tracking that are organised into categories like billing and account management. Each entry includes a high-quality response, entity slot annotations and stylistic variations. Because the data is synthetically generated and linguistically curates it offers a balanced accurate reflection of customer phrasing without the inconsistencies of raw data. This makes it an ideal benchmark for comparing architectural approaches. Developers can use it for fine tuning an LLM for end-to-end response generation or to train intent classifier for traditional pipeline systems.

4.2 Performance Metrics

For validating the performance of the proposed model, some performance metrics of positive measures such as accuracy, precision, recall, F-measure, specificity, Mathews Correlation Coefficient (MCC), Negative Prediction

value (NPV), Mean reciprocal rank (MRR), Bilingual Evaluation Understudy (BLEU), Recall Oriented Understudy for Gisting Evaluation (ROUGE) and Type II negative measures like False Positive rate (FPR), False Negative Rate (FNR) and False Discovery Rate (FDR). The mathematical description of these performance metrics are outlined in Table 4.

Table 4: Mathematical description of performance metrics

Performance metrics	Description
Accuracy	
Precision	
Recall	
F-Measure	
Specificity	
MCC	
NPV	
MRR	
BLEU	
ROUGE	
FPR	

4.3 Performance Evaluation

This research's performance is evaluated using various performance metrics in order to show the efficiency of the proposed model using different existing models including DNN [27], GUR_CNN [28], LSTM_RNN [29] and InceptionV3 [30]. BLEU, ROUGE-L, and MRR analysis utilized for different methods like BERT [31], RoBERTa-FNN [32], DistillBERT [33], ALBERT [35] and T5 [36].

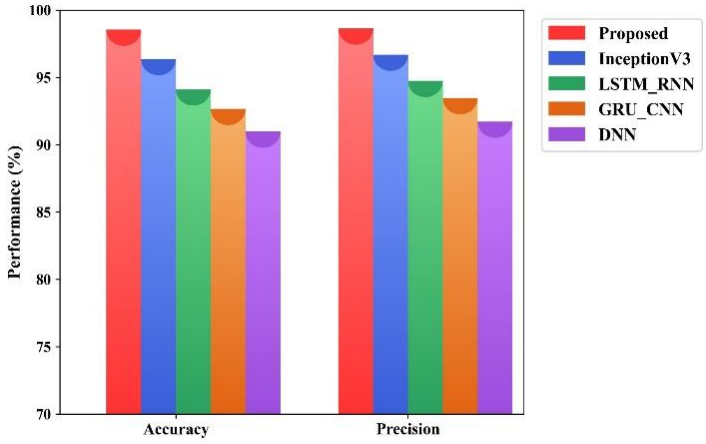


Figure 10: Accuracy and Precision analysis of proposed model with previous techniques

Figure 10 presents a comparative performance analysis of the proposed system against four baseline models like Inception V3, LSTM_RNN, GRU_CNN, DNN. The superior results of the proposed method over accuracy and precision id highlighted in this bar chart. The proposed model obtained 98.55% of accuracy and 98.64% of precision. The proposed model outperforms the existing models in which the InceptionV3 obtained 96.37% accuracy and 96.69% of precision; the LSTM-RNN model obtained 94.11% Accuracy and 94.74% Precision. Followed by the GRU_CNN model obtained 92.66% Accuracy and 93.46% Precision. And the DNN model shows lowest results across both metrics. This visual comparison confirms that the proposed architecture maintains high consistency and reliability, proving that the integration of IOOA for optimization and the DDTCN-NLF for classification effectively captures complex conversational patterns that conventional models fail to process with the same level of granularity.

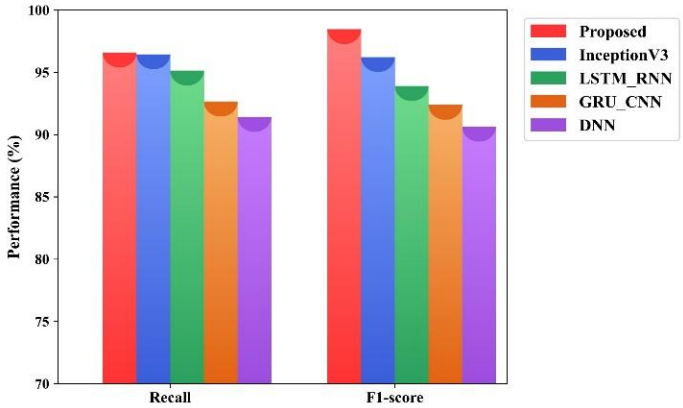


Figure 11: Performance validation of Recall and F1-score analysis

Figure 11 presents a comparative performance analysis of the proposed system against four baseline models, including InceptionV3, LSTM_RNN, GRU_CNN, and DNN, focusing on Recall and F1-score. The bar chart highlights the superior performance of the proposed method, which achieved a Recall of 98.55% and a top-tier F1-score of 98.23%. In comparison, the existing models show lower efficiency, with InceptionV3 obtaining 96.37% Recall and 96.22% F1-score, while the LSTM_RNN model reached 94.36% Recall and 93.98% F1-score. These are followed by the GRU_CNN model with 92.68% Recall and 92.43% F1-score, and the DNN model, which exhibited the lowest performance across both metrics. This visual data confirms that the proposed architecture maintains an exceptional balance between sensitivity and precision, validating that the combination of IOOA and the DDTCN-NLF classifier effectively handles the complexities of intent detection more robustly than conventional deep learning approaches.

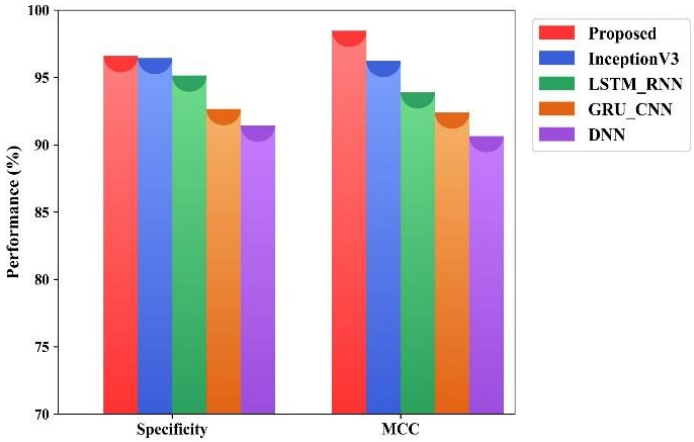


Figure 12: Performance evaluation of specificity and MCC analysis

The comparative performance analysis of the proposed system against four baseline models like InceptionV3, LSTM_RNN, GRU_CNN, and DNN, focusing on Specificity and MCC is shown in Figure 12. The superior results of the proposed method are clearly highlighted in this bar chart, with the proposed model obtaining 96.59% of specificity and 98.47% of MCC. In comparison, the existing models show lower efficacy, where InceptionV3 obtained 96.45% specificity and 96.23% of MCC, while the LSTM_RNN model achieved 95.14% specificity and 93.9% MCC. These are followed by the GRU_CNN model, which obtained 92.65% specificity and 92.39% MCC, and the DNN model, which shows the lowest results across both metrics. This visual comparison confirms that the proposed architecture maintains high predictive quality and reliability, proving that the integration of IOOA for optimization and the DDTCN-NLF for classification effectively balances true negative rates and overall correlation in ways that conventional models fail to achieve.

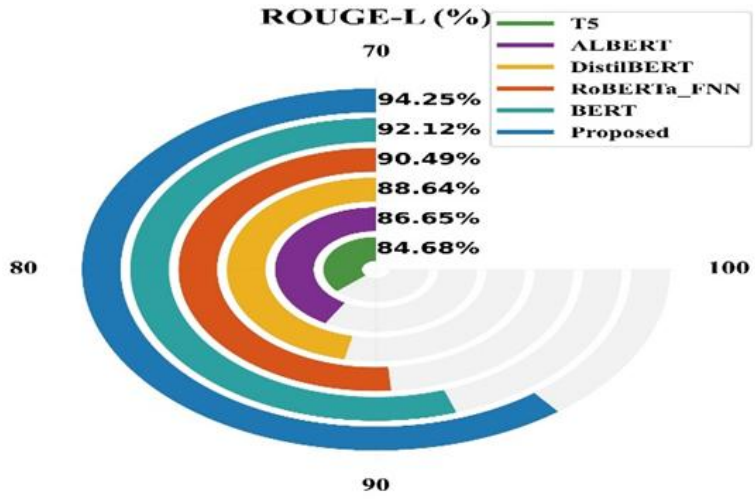


Figure 13: ROUGE-L analysis

Figure 13: presents a comparative performance analysis of the proposed system against five state-of-the-art language models, including T5, ALBERT, DistilBERT, RoBERTa_FNN, and BERT, specifically evaluating the ROUGE-L metric. The superior text generation capabilities of the proposed method are clearly highlighted in this circular bar chart, with the proposed model obtaining a top-tier score of 94.25%. In comparison, the existing models show lower overlap with reference responses, where the standard BERT model obtained 92.12% and the RoBERTa_FNN model achieved 90.49%. Then the DistilBERT achieved 88.64%, ALBERT at 86.65%, and the T5 model, which shows the lowest result of 84.68%. This visual comparison confirms that the proposed architecture maintains a high level of linguistic coherence and structural similarity to human responses, proving that the integration of the recurrent link and hybrid cross-attention in the generation phase significantly enhances the quality of the chatbot's output. Figure 14 demonstrates the ROC curve analysis of suggested method with different approaches.

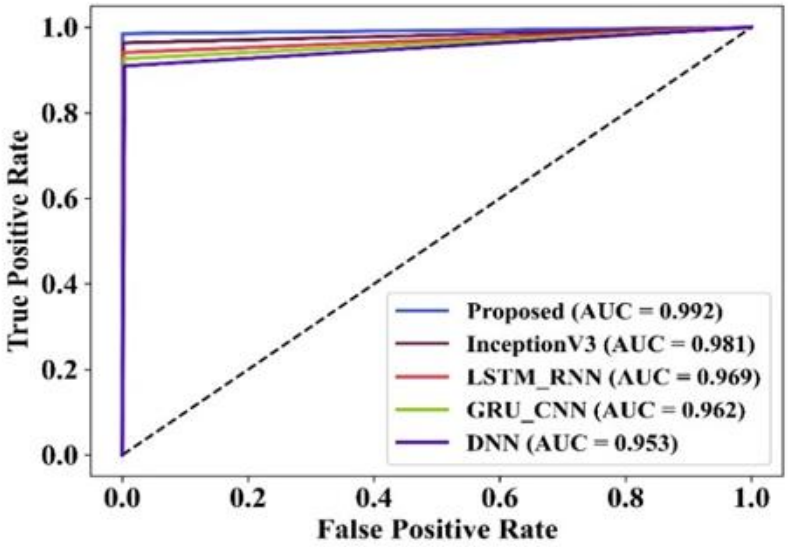


Figure 14: ROC curve analysis

The evaluation of the proposed model for Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC against four existing models like InceptionV3, LSTM_RNN, GRU_CNN, and DNN. The superior

diagnostic ability of the proposed method is clearly highlighted in this plot, with the proposed model obtaining a near-perfect AUC of 0.992. In comparison, the existing models exhibit lower classification efficiency, where InceptionV3 obtained an AUC of 0.981 and the LSTM_RNN model achieved 0.969. These are followed by the GRU_CNN model with an AUC of 0.962 and the DNN model, which shows the lowest result of 0.953. This visual comparison confirms that the proposed architecture maintains a significantly higher true positive rate while minimizing false positives, proving that the integration of IOOA for feature selection and the DDTCN-NLF for intent classification provides a much more robust and accurate decision-making boundary than conventional deep learning approaches.

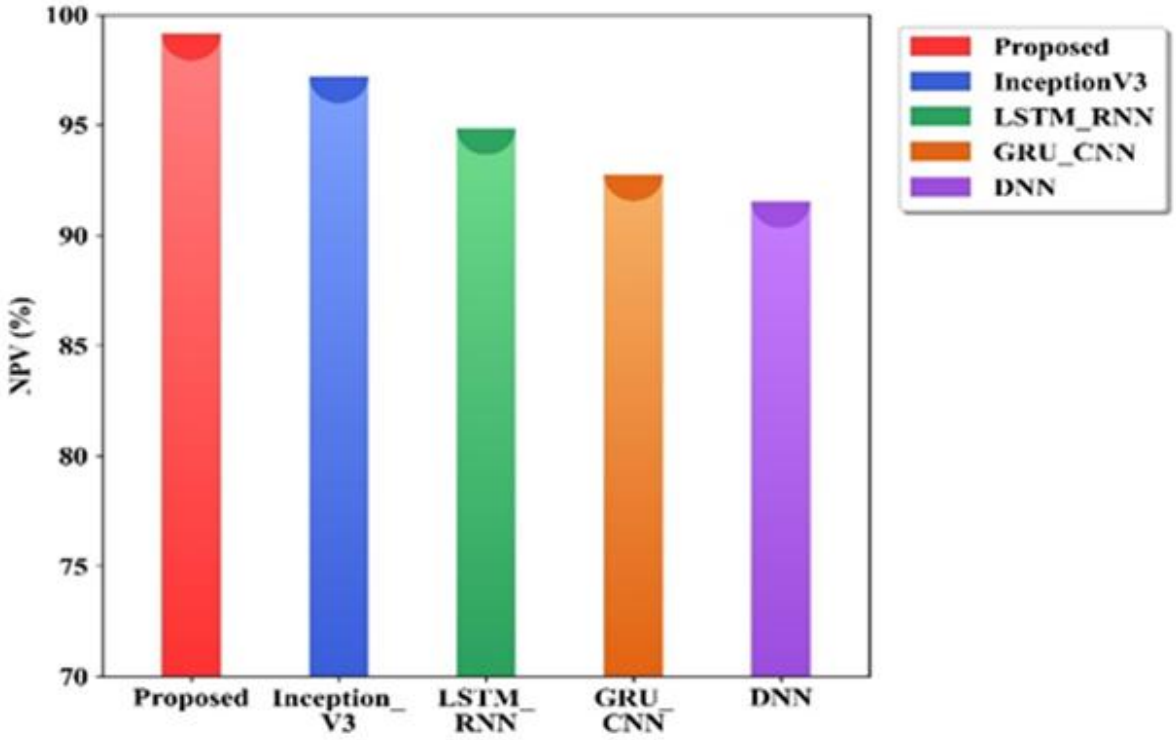


Figure 15: NPV analysis

Figure 15: presents a comparative performance analysis of the proposed system against four baseline models, including InceptionV3, LSTM_RNN, GRU_CNN, and DNN, specifically evaluating the Negative Predictive Value (NPV). The superior reliability of the proposed method is clearly highlighted in this bar chart, with the proposed model obtaining a peak NPV of 99.14%. In comparison, the existing models demonstrate a lower capacity for accurate negative predictions, where InceptionV3 obtained 97.2% and the LSTM_RNN model achieved 94.86%. These are followed by the GRU_CNN model at 92.76% and the DNN model, which shows the lowest result of 91.54%. This visual comparison confirms that the proposed architecture maintains an exceptional level of precision in identifying non-target intents, proving that the synergy between IOOA-based feature weighting and DDTCN-NLF classification significantly reduces false negative outcomes compared to conventional models.

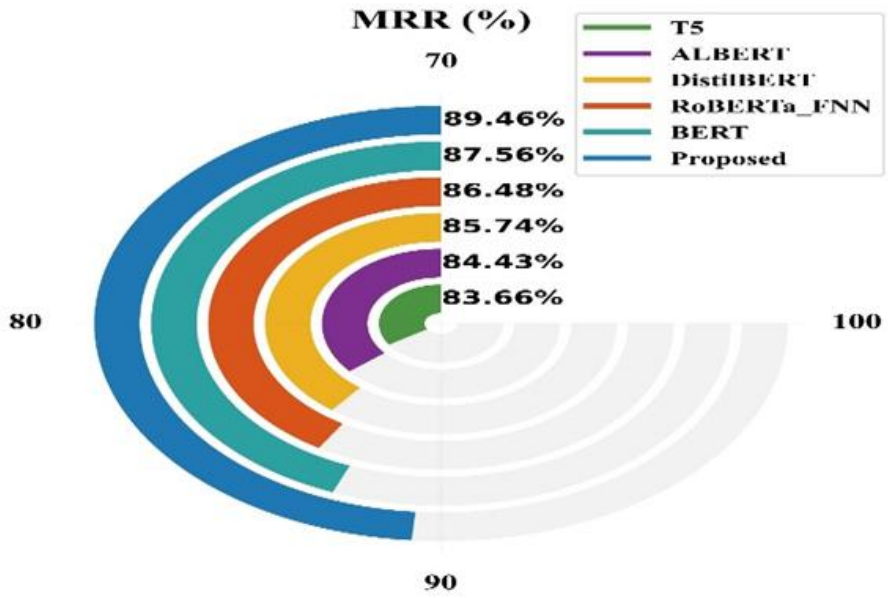
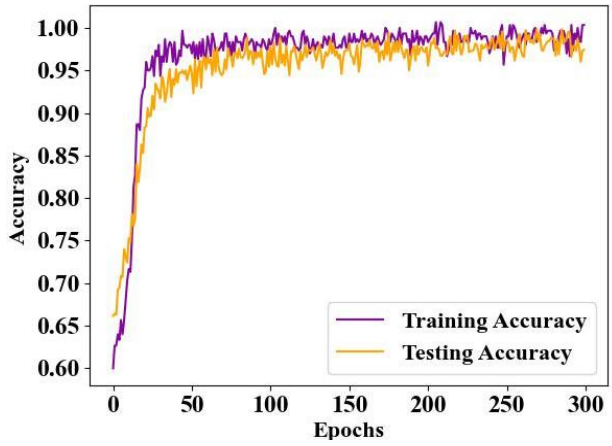
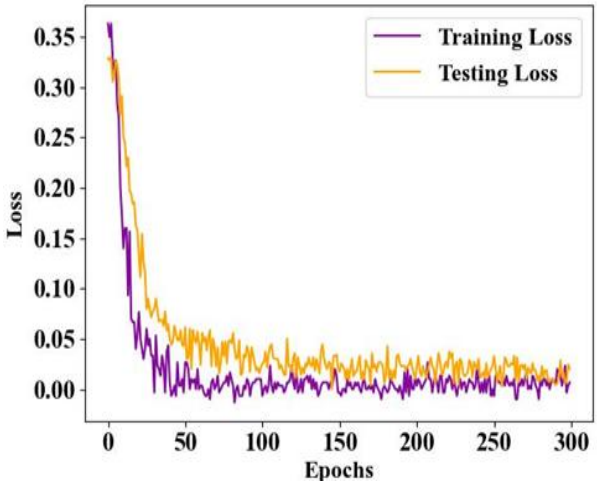


Figure 16: MRR analysis

The comparative analysis of the proposed system against five state-of-the-art language models, including T5, ALBERT, DistilBERT, RoBERTa_FNN, and BERT, specifically evaluating the Mean Reciprocal Rank (MRR) is shown in Figure 16. The superior retrieval efficiency of the proposed method is clearly highlighted in this circular bar chart, with the proposed model obtaining a top-tier score of 89.46%. In comparison, the existing models show lower ranking accuracy, where the standard BERT model obtained 87.56% and the RoBERTa_FNN model achieved 86.48%. These are followed by DistilBERT at 85.74%, ALBERT at 84.43%, and the T5 model, which shows the lowest result of 83.66%. This visual comparison confirms that the proposed architecture consistently places the most relevant responses at higher rank positions, proving that the integration of intent-aware contextual features significantly optimizes the response retrieval process compared to conventional transformer-based models. Figure 17 illustrates the training and testing accuracy and loss analysis of proposed with previous techniques.



(a) Training and testing accuracy analysis



(b) Training and Testing loss analysis

Figure 17: (a-b): Training and Testing accuracy and loss analysis

Figure 17 (a-b): Training and Testing accuracy and loss analysis

In above graph illustrate the proposed approach, the model's accuracy improves as the number of epochs rises, as seen in the graph. Both training and testing accuracy are initially low, but they rapidly increase throughout the first few epochs, demonstrating the model's rapid learning. After around 50 epochs, the accuracy stabilizes and stays at a very high level between 97% and 99%. The model functioning effectively and is not overfitting because the training accuracy is somewhat greater than the testing accuracy. Overall, using both training and testing data, the model consistently produces results with excellent accuracy. For both training and testing data, the graph demonstrates how the loss drops as the number of epochs rises. The loss is high at first, but it rapidly decreases within the first few epochs, suggesting that the method is learning efficiently. The loss stabilizes and becomes extremely low after about 50 epochs. Although the testing loss is significantly higher than the training loss they are still rather close to one another. This slight variation demonstrates that the model is functioning properly on new data and is not overfitting. After enough training, the model generally achieves minimal loss and steady performance. Figure 18 demonstrates the confusion matrix analysis.

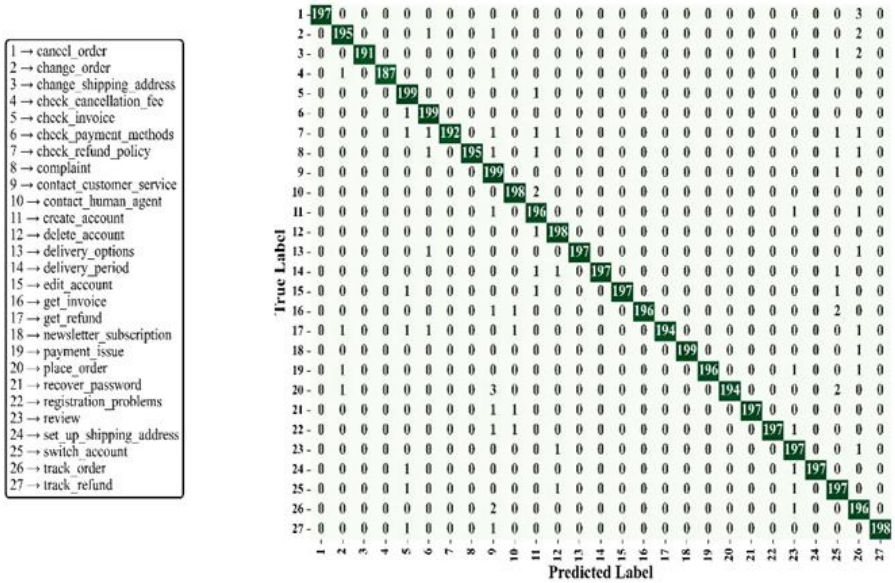


Figure 18: Confusion matrix analysis

Confusion matrix is displayed in this graphic and is used to assess a classification model's effectiveness. The anticipated labels are shown in each column, while the actual (true) labels are shown in each row. Correct predictions are indicated by the numbers along the diagonal, indicating that the model properly identified those classes. The fact that the majority of the data are concentrated on this diagonal suggests that the model is operating with excellent accuracy. Only a few misclassifications are visible in the other values outside the diagonal, which are extremely tiny. Overall, the method has very few errors and can accurately classify the majority of the categories. BLEU analysis are shown in figure 19.

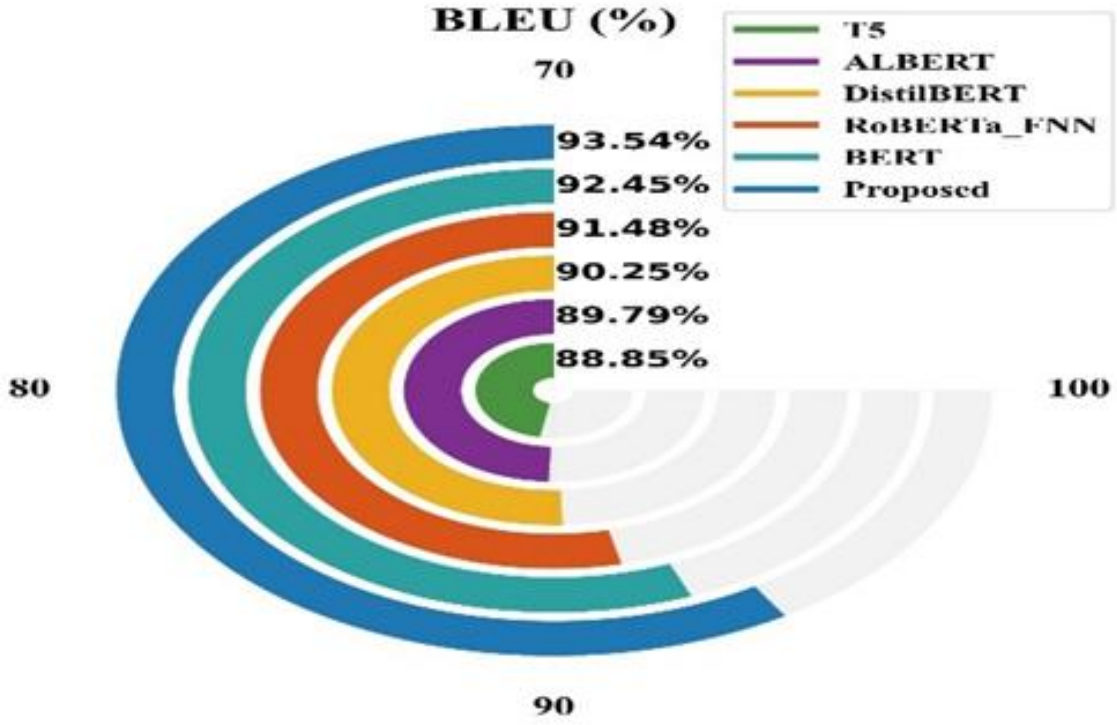


Figure 19: BLEU analysis

The proposed model defines the BLEU (Bilingual Evaluation Understudy) in artificial intelligence conversation bots is a metric that rates the quality of text produced with human-written reference responses on a scale of 0 to 1. The BLEU score performance of several models is compared in the Figure 18. A model is represented by each coloured ring, the percentage numbers show the model's performance. With the greatest score of 93.54%, the proposed method performs best of all. BERT model achieved BLEU value of 93.45%, respectively. RoBERTa-CNN model achieved BLEU value of 91.48%, respectively. DistilBERT model attained 90.25%, ALBERT model achieved BLEU value of 89.79%, respectively. T5 method achieved lowest score value of 88.85%. Overall, the graph unequivocally demonstrates that the suggested strategy performs in terms of BLEU score than the current approach. Table 5 illustrates the performance validation of the proposed approach with existing models. Table 6 demonstrates the BLEU, ROUGE-L and MRR analysis with proposed model with previous methods.

Table 5: Performance Evaluation of the proposed approach with existing models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)	NPV (%)	MCC (%)
InceptionV3	96.37	96.69	96.37	96.22	96.45	97.2	96.23
LSTM_RNN	94.11	94.74	94.36	93.98	95.14	94.86	93.9
GRU_CNN	92.66	93.46	92.68	92.43	92.65	92.76	92.39
DNN	90.98	91.72	91.01	90.74	91.42	91.54	90.64
Proposed	98.55	98.64	98.55	98.23	96.59	99.14	98.47

Table 6: Performance validation of the BLEU, ROUGE-L and MRR analysis with existing approaches

Model	BLEU	ROUGE-L	MRR
BART	0.9245	0.9212	0.8756
RoBERTa-FNN	0.9148	0.9049	0.8648
DistilBERT	0.9025	0.8864	0.8574
ALBERT	0.8979	0.8665	0.8443
T5	0.8885	0.8468	0.8366
Proposed	0.9354	0.9425	0.8946

5. Discussion

By this section, this research aims to provide further understanding about the research of advanced AI-based chatbot architectures for customer support. Instead of focusing on simple pattern matching or basic classification, this research focused on dimensionality reduction, optimal feature selection, and high-fidelity contextual response generation. For providing better generalizability and reliability, this research experimented on the Bitext GenAI Chatbot Dataset, which comprised diverse intent categories and conversational variations. The input user query was pre-processed for processing into the MTVC phase. The input query was tokenized and converted into a multiscale vector format for efficient representation. MTVC encodes these vectors to capture hierarchical semantic meanings and reduces their numerical dimensionality. Most researchers, such as Babayigit et al. and Nageshwaran et al. [19], utilized standard LSTM or CNN layers which often struggle with high-dimensional data and long-range dependencies. This MTVC framework enhances the presented research's outcomes by ensuring a dense and informative vector space.

Further, the research addresses the limitations of computational complexity and local optima found in models like AA-DTCN-SC [18] by introducing the IOOA. This optimization algorithm utilizes chaotic maps to refine feature weights, effectively filtering out noise and ensuring only the most significant representations are passed to the intent classification phase. Based on these optimized features, the DDTCN-NLF categorizes user intents. By utilizing dilated convolutions and Triplet Loss, the model captures long-term temporal dependencies while ensuring high inter-class separation, a significant improvement over the generalizability issues seen in Naïve Bayes [17] approaches.

Finally, instead of directly employing standard Large Language Models or basic GPT variants like PFA-GPT [20], the suggested research utilized the HCAR-BERT framework for answer generation. This model integrates a Hybrid Cross-Attention mechanism and Recurrent links to provide grounded and contextually aware responses. Compared with traditional BERT or transformer-based models, the HCAR-BERT approach effectively produces texts that are temporally coherent and highly accurate, achieving a superior Accuracy of 98.55%. The integration of these mechanisms contributed to the chatbot framework's robust and reliable performances across various customer support scenarios, effectively minimizing the vagueness and training inefficiencies found in existing state-of-the-art models. Table 7 illustrates the comparative analysis of proposed approach and previous techniques.

Table 7: Comparative analysis of proposed with previous approaches

Author & References	Methods	Accuracy (%)
Babayigit et al. [16]	TrBot	80
Mikael et al. [17]	Naïve Bayes	97.57
Kathole et al. [18]	AA-DTCN-SC	92.68
Nageshwaran et al. [19]	HCLSTM	97
Taiwo et al. [20]	PFA-GPT	93
Proposed	MTVC-IOOA based DDTCN-HCAR-BERT	98.55

Table 7: Comparative analysis of proposed with previous approaches

5.1 Rationale and Implications of work

In proposed model rationale of work using Retrieval-Augmented Generation (RAG) and LLMs are used to give context-aware, precise, multi-step reasoning when designing an effective AI chatbot for complicated challenges. Managing ambiguity is multi-turn, ambiguous questions are a challenge for traditional bots. Advanced AI can

comprehend the intent of users behind challenging issues. Scalability and Efficiency of complex tasks can be automated to save costs and boost efficiency. The implication of work is strategic decision support by evaluating enormous datasets to support intricate decision-making, these chatbots function as knowledgeable advisors. Ethical and security responsibilities for designing these systems necessitates proactive mitigation of AI hallucinations and data security issues, and ChatGPT frameworks.

6. Conclusion

In conclusion, this research framework addresses a comprehensive analysis of contextual response generation and intent classification for customer support systems. Initially, data is processed through the MTVC phase to reduce numerical complexity, followed by the IOOA, which effectively optimizes feature weights. Afterwards, the DDTCN-NLF is utilized for intent classification, which optimizes decision boundaries and handles uncertain information. This model avoids misclassification in domain-specific queries and increases the accuracy of user intent detection. Finally, the HCAR-BERT enhances the authentication of context through cross-attention and recurrent memory. This proposed model provides high precision analysis and effective response generation while it optimally balances semantic meaning between the dialogue process. The proposed model achieves a high accuracy of 98.55%, an F1-score of 98.23%, a BLEU score of 0.9354 and an MRR of 0.8946 for customer support dataset. This proposed model's performance analysis is better than existing models, while it states their efficiency and generalization performances in real-time applications. However, the proposed model performs well under dynamic customer support environments, making it suitable for real-time applications like e-commerce, banking, and technical helpdesks. In future, the proposed model will be introduced to multimodal data integration to evaluate sentiment scores between users and increase the personalization process. This process will increase consistency and reliability between the bot and user as well as enhance performances in real-time global enterprise applications.

Abbreviations

Abbreviations	Full form
AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
LLM	Large Language Model
GPT	Generative Pre-trained Transformers
NMT	Neural Machine Translation
MTVC	Multiscale Transformer Vector Conversion
IOOA	Improved Osprey Optimization Algorithm
DDTCN-NLF	Dilated Deep Temporal Convolutional Networks with Novel Loss Function
HCAR-BERT	Hybrid Cross Attention assisted Recurrent Bidirectional Encoder Representation from Transformer
AIML	Artificial Intelligence Markup Language
ADT-BMO	Averaging-based Driving Training-Barnacles Mating Optimizer
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformer
HCLSTM	Hybrid Convolutional Long Short-Term Memory
PFA-GPT	Psychological First Aid Generative Pre-trained Transformer
DTCN	Deep Temporal Convolution network
MLM	Masked Language Model
NSP	Next Sentence Prediction
QA	Question Answering
NLI	Natural Language Inference
RNN	Recurrent Neural Network

Abbreviations	Full form
CAM	Cross Attention mechanism
LSTM-RNN	Long Short Term Memory-Recurrent Neural Network
GRU-CNN	Gated Recurrent Unit-Convolutional Neural Network
DNN	Deep Neural Network
RAG	Retrieval-Augmented Generation

Reference

1. Kiangala, Kahiomba Sonia, and Zenghui Wang. "An experimental hybrid customized AI and generative AI chatbot human machine interface to improve a factory troubleshooting downtime in the context of Industry 5.0." The International Journal of Advanced Manufacturing Technology 132, no. 5 (2024): 2715-2733.
2. Pergantis, Pantelis, Victoria Bamicha, Charalampos Skianis, and Athanasios Drigas. "Ai chatbots and cognitive control: Enhancing executive functions through chatbot interactions: A systematic review." Brain Sciences 15, no. 1 (2025): 47.
3. Balasubramaniam, S., A. Prasanth, K. Satheesh Kumar, and Seifedine Kadry. "Artificial Intelligence-Based Hyperautomation for Smart Factory Process Automation." Hyperautomation for Next-Generation Industries (2024): 55-89.
4. Davar, Narius Farhad, M. Ali Akber Dewan, and Xiaokun Zhang. "AI chatbots in education: challenges and opportunities." Information 16, no. 3 (2025): 235.
5. Izadi, Saadat, and Mohamad Forouzanfar. "Error correction and adaptation in conversational AI: a review of techniques and applications in chatbots." Ai 5, no. 2 (2024): 803-841.
6. Gunnam, Ganesh Reddy, Devasena Inupakutika, Rahul Mundlamuri, Sahak Kaghyan, and David Akopian. "Assessing performance of cloud-based heterogeneous chatbot systems and a case study." IEEE Access 12 (2024): 81631-81645.
7. Halvoník, Dominik, and Jozef Kapusta. "Large language models and rule-based approaches in domain-specific communication." IEEE Access (2024).
8. Jain, Lavish, Rahul Ananthasayam, and Uditi Gupta. "Comparison of Rule-based Chat Bots with Different Machine Learning Models." Procedia Computer Science 259 (2025): 788-798.
9. Wyawahare, Medha, Shreya Roy, and Sarang Zanwar. "Generative vs intent-based chatbot for judicial advice." In 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), vol. 2, pp. 1-6. IEEE, 2024.
10. Antonie, Ninel Iacobus, Gina Gheorghe, Vlad Alexandru Ionescu, Loredana-Crista Tiucă, and Camelia Cristina Diaconu. "The role of ChatGPT and AI Chatbots in optimizing antibiotic therapy: a comprehensive narrative review." Antibiotics 14, no. 1 (2025): 60.
11. Zhang, Chaobo, Jian Zhang, Yang Zhao, and Jie Lu. "Automated data-driven building energy load prediction method based on generative pre-trained transformers (GPT)." Energy 318 (2025): 134824.
12. Attigeri, Girija, Ankit Agrawal, and Sucheta V. Kolekar. "Advanced NLP models for technical university information chatbots: Development and comparative analysis." IEEE Access 12 (2024): 29633-29647.
13. Mikael, Kanaan, Ö. Z. Cemil, Tarik A. Rashid, and Goran S. Nariman. "A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization." IEEE Access (2025).
14. Rema, Catarina, Armando Sousa, Heber Sobreira, Pedro Costa, and Manuel F. Silva. "Exploring the Potential of LLM-based Chatbots for Task Scheduling in Robot Operations." In 2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 45-51. IEEE, 2025.
15. Alam, Khorshed, Mahbubul Haq Bhuiyan, Mohammad Shafiqul Islam, Abul Hossain Chowdhury, Zaheed Ahmed Bhuiyan, and Suman Ahmmed. "Co-Pilot for Project Managers: Developing a PDF-Driven AI Chatbot for Facilitating Project Management." IEEE Access (2025).
16. Babayigit, Bilal, Habibelahi Rahmani, and Mohammed Abubaker. "TrBot: A Turkish Deep Learning Chatbot Utilizing Seq2Seq Model." IEEE Access (2025).
17. Mikael, Kanaan, Ö. Z. Cemil, Tarik A. Rashid, and Goran S. Nariman. "A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization." IEEE Access (2025).
18. Kathole, Atul, Suvarna Patil, Devyani Jadhav, Hirkani Pathak, and Amita Sanjiv Mirge. "Development of student intent-based educational chatbot system with adaptive and attentive DTCN on symmetric convolution approach." MethodsX (2025): 103542.
19. Nageshwaran, Vinoth, and Sathish Nageshwaran. "Leveraging LLM-Powered Chatbot for Learning and Optimization in Databases and Information Systems Education." In 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), pp. 268-273. IEEE, 2025.

20. Taiwo, Olajumoke, and Baidaa Al-Bander. "Emotion-aware psychological first aid: Integrating BERT-based emotional distress detection with Psychological First Aid-Generative Pre-Trained Transformer chatbot for mental health support." Cognitive Computation and Systems 7, no. 1 (2025): e12116.
21. Ismaeel, Alaa AK, Essam H. Houssein, Doaa Sami Khafaga, Eman Abdullah Aldakheel, Ahmed S. AbdElrazek, and Mokhtar Said. "Performance of osprey optimization algorithm for solving economic load dispatch problem." Mathematics 11, no. 19 (2023): 4107.
22. Lara-Benítez, Pedro, Manuel Carranza-García, José M. Luna-Romera, and José C. Riquelme. "Temporal convolutional networks applied to energy-related time series forecasting." applied sciences 10, no. 7 (2020): 2322.
23. Zhou, Xiaoyan, Tao Tang, Yuting Cui, Linbin Zhang, and Gangyao Kuang. "Novel loss function in CNN for small sample target recognition in SAR images." IEEE Geoscience and Remote Sensing Letters 19 (2021): 1-5.
24. Aftan, Sulaiman, and Habib Shah. "A survey on bert and its applications." In 2023 20th Learning and Technology Conference (L&T), pp. 161-166. IEEE, 2023.
25. Li, Hui, and Xiao-Jun Wu. "CrossFuse: A novel cross attention mechanism based infrared and visible image fusion approach." Information Fusion 103 (2024): 102147.
26. https://www.kaggle.com/datasets/bitext/bitext-gen-ai-chatbot-customer-support-dataset
27. Dzaky, Azmi Abiyyu, Junta Zeniarja, Catur Supriyanto, Guruh Fajar Shidik, Cinantya Paramita, Egia Rosi Subhiyakto, and Sindhu Rakasiwi. "Optimization chatbot services based on DNN-BERT for mental health of university students." Journal of Applied Informatics and Computing 8, no. 1 (2024): 13-21.
28. Ghildiyal, Vaibhav. "An intelligent chatbot using deep learning with bidirectional GRU and CNN model." International Journal of Mathematics in Operational Research 29, no. 2 (2024): 258-270.
29. Choudhary, Prakash, and Sumit Chauhan. "An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism." Decision Analytics Journal 9 (2023): 100359.
30. Ormeño-Arriagada, Pablo, Gastón Márquez, Carla Taramasco, Gustavo Gatica, Juan Pablo Vasconez, and Eduardo Navarro. "Early Oral Cancer Detection with AI: Design and Implementation of a Deep Learning Image-Based Chatbot." Applied Sciences 15, no. 19 (2025): 10792.
31. Babu, Arun, and Sekhar Babu Boddu. "BERT-based medical chatbot: enhancing healthcare communication through natural language understanding." Exploratory research in clinical and social pharmacy 13 (2024): 100419.
32. Hazim, Layth Rafea, and Oguz Ata. "Textual authenticity in the AI era: evaluating BERT and RoBERTa with logistic regression and neural networks for text classification." In 2024 International Symposium on Electronics and Telecommunications (ISETC), pp. 1-6. IEEE, 2024.
33. Hussain, Muhammad, Caikou Chen, Muzammil Hussain, Muhammad Anwar, Mohammed Abaker, Abdelzahir Abdelmaboud, and Iqra Yamin. "Optimised knowledge distillation for efficient social media emotion recognition using DistilBERT and ALBERT." Scientific Reports 15, no. 1 (2025): 30104.
34. Hwang, Myeong-Ha, Jikang Shin, Hojin Seo, Jeong-Seon Im, Hee Cho, and Chun-Kwon Lee. "Ensemble-nqg-t5: Ensemble neural question generation model based on text-to-text transfer transformer." Applied Sciences 13, no. 2 (2023): 903.