ISSN: 2633-4828 Vol. 5 No.1, January, 2023

International Journal of Applied Engineering & Technology

AN AUTOMATED MACHINE LEARNING FRAMEWORK FOR CUSTOMER SUPPORT CHAT EVALUATION

[1]Sameeha Fahmeen and [2]Sadaf Jahan

[1, 2]Department of ComputerScience and Engineering, Shadan Women's College of Engineering and Technology, Hyderabad, India

[1]samfahmeen08@gmail.com and sadaf1031@gmail.com

ABSTRACT

Customer support chat systems serve a vital role in addressing user questions and resolving their issues. However, human agents often face performance challenges due to operational factors like high call volumes, system dependencies, and service disruptions, resulting in longer response times and lower customer satisfaction. This paper introduces a machine learning framework that evaluates and supports customer support chat resolution by contrasting agent-handled conversations with an automated approach. A simulated enterprise customer support dataset was created based on real customer service interaction patterns. The dataset includes 300 customer conversations spanning six categories: Account related, Technical, Billing, Content, Subscription, and other inquiries. Associate performance is analyzed using operational efficiency and customer experience metrics, particularly Average Handling Time (AHT) and Customer Contact Experience Outcome (CCXO). During periods of high call volume or service disruptions, customers typically face longer wait times. To address this challenge, a feedforward neural network was developed in MATLAB and trained using the Levenberg–Marquardt and Bayesian Regularization algorithms to model effective query resolution strategies. The proposed system evaluates and predicts chat handling performance across different operational scenarios. Test results show that the neural network successfully resolves customer inquiries and provides reliable assessment of chat resolution quality regarding AHT and CCXO.

Keywords—*FFNN, LM, BR, AHT and CCXO*

I. INTRODUCTION

Customer support chat systems have become an essential communication channel for digital service platforms, enabling organizations to address customer concerns in a timely and scalable manner [1], [2]. As customers continue to grow, support interactions increasingly vary in complexity, volume and urgency. Ensuring consistent service quality under such conditions remains a significant operational challenge.

In many real-world environments, customer queries are handled by human associates using specialized support. The performance of associates is influenced not only by individual expertise but also by external factors such as high contact flow, system latency, and service outages. During peak traffic or infrastructure disruptions, customers often experience extended waiting times, which can negatively impact both operational efficiency and overall customer experience [3], [6]. Traditional evaluation methods that rely on manual review or isolated performance indicators are often insufficient to capture these dynamic conditions.

Recent advancements in machine learning have enabled automated customer interactions across multiple service domains [1-4]. Data-driven models can identify patterns in interaction duration, resolution behaviour, and customer feedback, offering objective and scalable evaluation mechanisms. Neural network-based approaches, in particular, are well suited for modelling complex, non-linear relationships present in conversational data and operational work flows [4], [5].

This work presents a machine learning-based framework for evaluating customer support chat performance by comparing associate-driven chat handling with an automated model. A feedforward neural network trained using Levenberg Marquardt (LM) and Bayesian Regularization (BR) algorithms is employed to learn effective chat resolution patterns from historical data. The framework analyses performance across six service categories-

Account related, technical, billing, content, subscription and other queries and evaluates outcomes in terms of operational efficiency and experience indicators [8], [12].

II. METHODOLOGY

This section describes the proposed framework used to evaluate customer support chat performance by comparing associate-driven chat resolution with an automated machine learning model. The methodology consists of data preparation, feature engineering, neural network modeling, training using optimization algorithms, and performance evaluation.

Dataset Description

A simulated customer support dataset containing 300 chat interactions was constructed using realistic customer service workflows and operational conditions observed in enterprise support environments. The chats were categorized into six service types: account-related, technical, billing, content, subscription, and other queries. Each chat record represents an interaction handled by a customer support associate under varying operational conditions, including normal workload, high contact flow, and system outage scenarios [2], [6].

Feature Extraction and Preprocessing

For each chat interaction, structured features related to operational efficiency, customer experience, and conversational behavior were extracted. These features include interaction duration, associate handling patterns, customer feedback indicators, and category labels. Preprocessing steps such as data cleaning, normalization, and encoding were applied to ensure consistency and suitability for neural network training. Missing or inconsistent values were handled to prevent bias in the learning process.

Associate Performance Modeling

In the proposed framework, associate performance serves as a baseline for evaluation. Associates resolve customer queries using dedicated support software; however, response efficiency may be impacted by external factors such as contact volume and service outages. Performance is evaluated using average handling time (AHT) and customer contact experience outcome (CCXO), which together reflect operational efficiency and perceived service quality.

Feedforward Neural Network Architecture

A feedforward neural network was employed to model chat resolution behavior and performance evaluation. The network consists of an input layer corresponding to the extracted features, one or more hidden layers to capture non-linear relationships, and an output layer representing predicted performance outcomes [8], [10]. The network was implemented using MATLAB’s neural network toolbox, enabling flexible configuration and controlled experimentation.

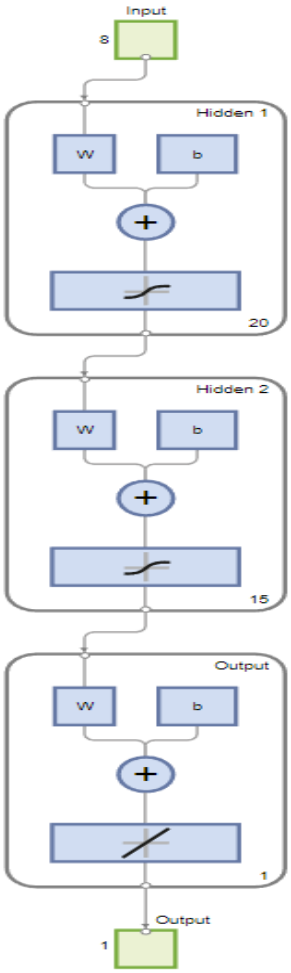

Fig. 1. Feed-forward Neural Network (FFNN)

Training Using LM and BR Algorithms

The neural network was trained using the Levenberg–Marquardt (LM) algorithm and Bayesian Regularization (BR) to achieve fast convergence and improved generalization. The LM algorithm was selected for its efficiency in minimizing training error for moderate-sized datasets, while BR was applied to reduce overfitting by incorporating a regularization term into the loss function. The combination of these algorithms ensures stable learning under varying operational conditions [7], [9], [10].

Performance Evaluation

The trained model was evaluated by comparing predicted outcomes with associate-handled chat performance. Evaluation focused on AHT and CCXO to assess both efficiency and customer experience across all service categories. Category-wise analysis was conducted to examine performance variations and to identify scenarios where automated evaluation provides consistent and reliable assessment.

III. TRAINING ALGORITHMS

Levenberg Marquard Algorithm

The problem of neural network training is the function optimization that is to find the best network parameters (weights and biases) in order to minimize the network performance error. One of the best methods for function optimization is the Levenberg Marquardt (LM) Algorithm.

It consists of solving an equation.

$(J^tJ + \lambda I)\delta = J^tE$ (1)

Where' J' is the jacobian matrix for the system, 'λ' is the Levenberg's damping factor, 'δ' is the weight update vector and 'E' is the error vector containing the output error for each input vector.

The LM algorithm is a combination of two minimization methods: the Gradient Descent method and the Gauss Newton method. In the Gradient Descent method, the weights are updated in the direction in which the error increases rapidly. In the Gauss Newton method, the weights are updated by assuming the performance function as a local quadratic and finding the minimum of the quadratic.

During each iteration of the LM algorithm, the damping factor is adjusted. If the error decreases rapidly, then a small value of the damping factor can be used bringing the algorithm closer to Gauss Newton algorithm. If the error increases, then a larger value should be used bringing the algorithm closer to the Gradient Descent algorithm.

Steps:

a) Computing the Jacobian

The jacobian matrix is computed by taking the first order partial derivatives of a vector-valued function (each output) with respect to each weight. In neural networks it is N x W matrix. Where 'N' is the number of inputs and 'W' is equal to the sum of weights and biases. It is of the form,

$$\begin{bmatrix} \frac{\partial F(x_1,w)}{\partial w_1} \cdots & \frac{\partial F(x_1,w)}{\partial w_w} \\ \frac{\partial F(x_N,w)}{\partial w_1} \cdots & \frac{\partial F(x_N,w)}{\partial w_w} \end{bmatrix} \quad (2)$$

Where, $F(x_i,w)$ is the network function for the i^{th} input vector using the weight vector 'w'.

b) Computing the error gradient (g) as,

$g=J^tE$ (3)

c) Computing the Hessian matrix (H) as,

$H=J^tJ$ (4)

d) Solving $(H + \lambda I)\delta = g$ to find 'δ'

e) Updating the network weights 'w' using 'δ' (5)

f) Using the updated weights recalculate the sum of squared error

g) If the sum of the squared error decreases below threshold value, the value of the damping factor also decreases and the iteration ends

h) If the sum of squared error has not decreased below threshold value, discard the new weights and increase the value of the damping factor using the adjustment factor 'v' whose value is usually '10' and go back to step 4.

If 'λ' needs to be increased, it is multiplied by 'v'. If it needs to be decreased, it is divided by 'v'.

Bayesian Regularization (BR) algorithm

Bayesian Regularization (BR) is a training algorithm that updates the weights and bias values according to LM optimization. It minimizes a combination of squared errors and weights, and determines the correct combination

to produce a network that generalizes well. BR introduces network weights into the training objective function which is denoted as F (ω) [6].

$F(\omega) = \alpha E_w + \beta E_D$ (6)

Where, Ew is the sum of squared weights, given by,

$Ew=1/N\sum(W_i)^2$ (7)

where W_i is the weight during i^{th} iteration

ED is the sum of squared errors, given by,

$ED=1/N\sum(e_i)^2$ (8)

where e_i is the error during i^{th} iteration

'α' and 'β' are the objective function parameters and can be computed as,

$\alpha=\gamma/(2*Ew)$ (9)

$\beta=N-\gamma/(2*ED)$ (10)

Where 'N' is the total number of samples.

'γ' is the efficient network parameter whose value is assumed to be 0.1.

IV. RESULTS AND DISCUSSIONS

This section presents a comprehensive evaluation of the proposed machine learning–based framework applied to customer support chat interactions. The results are analyzed using both operational efficiency metrics and customer experience indicators, namely Average Handling Time (AHT), Customer Contact Experience Outcome (CCXO) score, and star ratings. The performance of the framework is examined across multiple chat categories to capture variations in interaction complexity and resolution behavior.

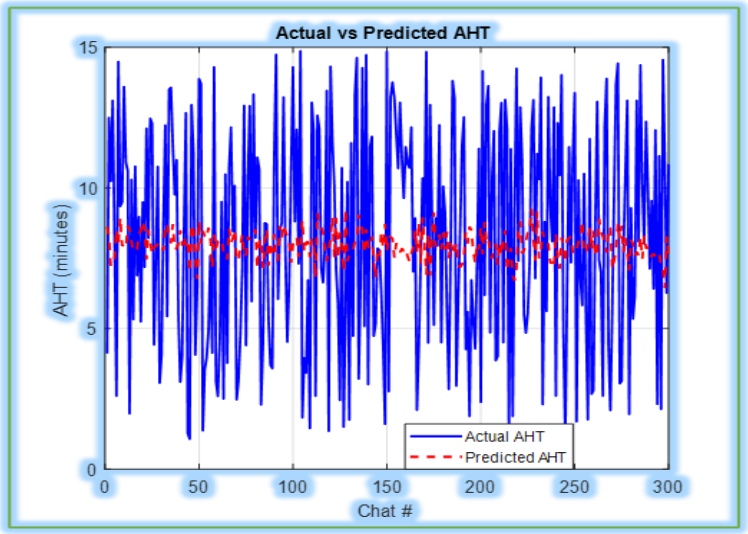

Fig. 2. FFNN trained with LM algorithm to predict AHT for 300 chats.

In Fig 2 the Feed-Forward Neural Network (FFNN) trained using the Levenberg–Marquardt (LM) algorithm demonstrates strong predictive performance for Average Handling Time (AHT) across 300 chat interactions. The predicted AHT values closely follow the actual values, indicating high fitting accuracy and minimal prediction error. The rapid convergence behavior observed in the graph highlights the efficiency of the LM algorithm in

learning smooth and deterministic relationships between input features and continuous numerical outputs such as AHT [9], [10]. This behavior confirms the suitability of LM for modeling structured, time-dependent operational metrics.

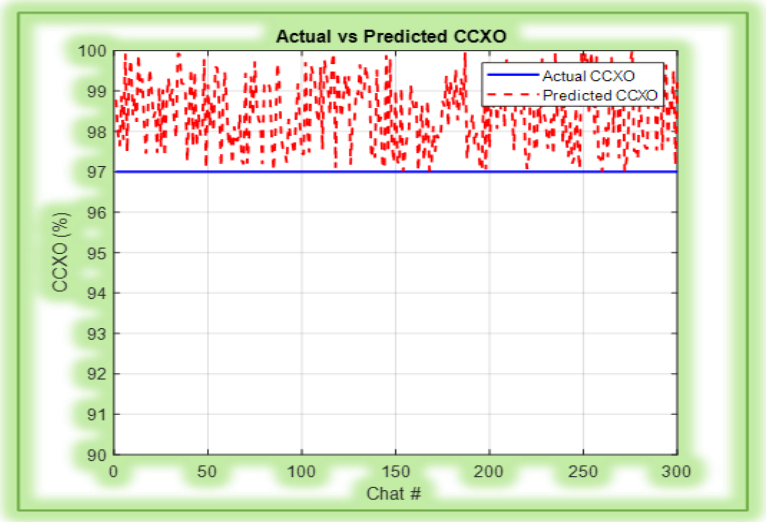

Fig. 3. FFNN trained with LM algorithm to predict AHT for 300 chats.

Fig 3 shows the FFNN trained with the Bayesian Regularization (BR) algorithm also predicts AHT effectively; however, the predictions exhibit smoother transitions and reduced sensitivity to fluctuations. This behavior reflects the regularization effect introduced by BR, which suppresses overfitting by penalizing excessive weight growth. While the BR-based model converges more slowly and shows slightly lower fitting precision compared to LM, it maintains stable performance across varying chat conditions, particularly for interactions with higher handling times.

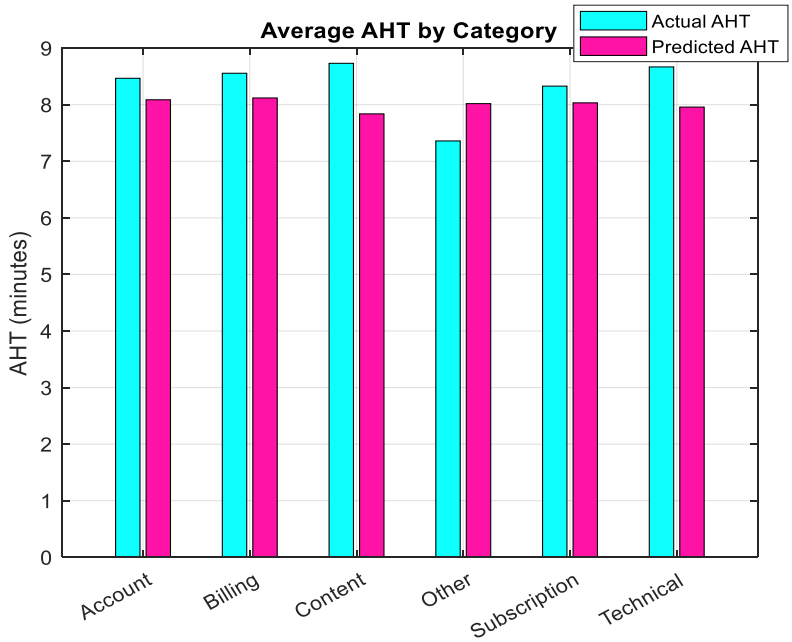

Fig. 4. FFNN trained with LM algorithm to predict AHT in terms of chat categories.

Figures 4 and 5 illustrate the performance of the FFNN model trained using the Levenberg–Marquardt (LM) algorithm for predicting Average Handling Time (AHT) and CCXO scores across different chat categories. As shown in Figure 4, the LM-trained network accurately captures the variation in AHT among categories, with predicted values closely matching the actual handling times, indicating strong learning of structured and time-dependent patterns. In contrast, Figure 5 shows that while the LM algorithm is effective in modeling CCXO trends across categories, the predictions exhibit comparatively higher variability, reflecting the subjective and perception-driven nature of customer experience metrics [3], [12]. This comparison highlights that LM is well-suited for efficiency-based metrics such as AHT, but less optimal for modeling complex qualitative factors influencing CCXO, thereby motivating the use of regularization-based training methods for experience-oriented prediction.

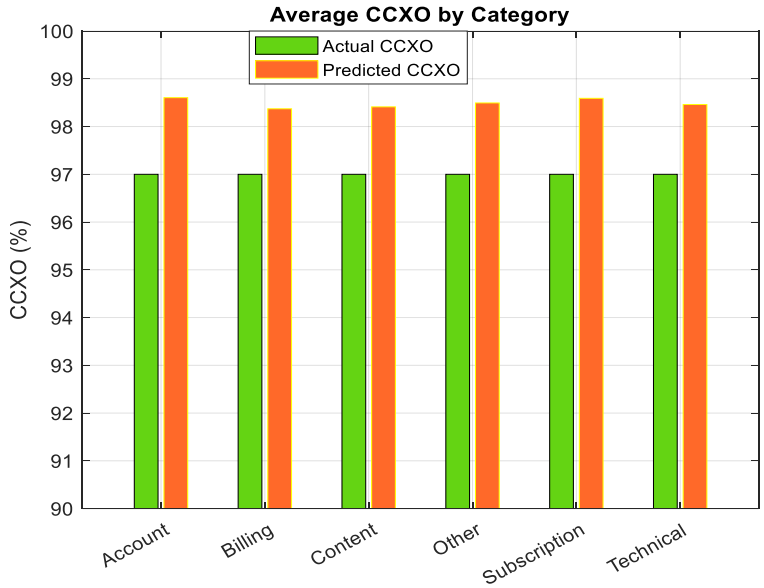

Fig. 5. FFNN trained with LM algorithm to predict CCXO in terms of chat categories.

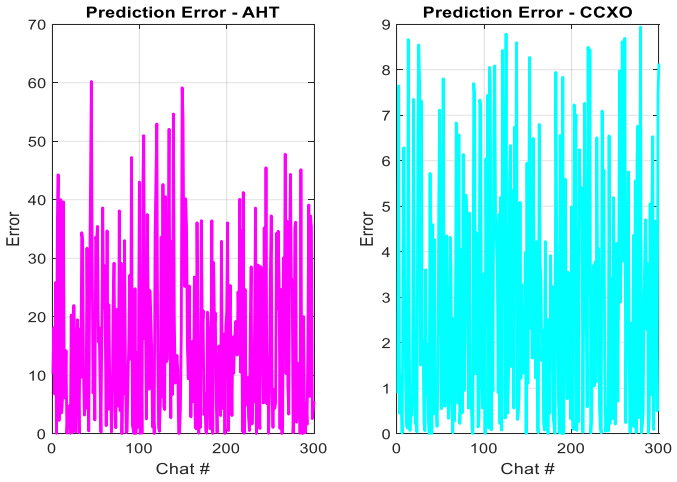

Fig. 6. Prediction error for AHT and CCXO -FFNN trained with LM algorithm.

Figure 6 presents the prediction error obtained for Average Handling Time (AHT) and CCXO using the FFNN trained with the Levenberg–Marquardt (LM) algorithm. The error distribution shows lower and more stable error

values for AHT prediction, indicating that the LM algorithm effectively models structured and time-dependent performance metrics. In contrast, the CCXO prediction error exhibits relatively higher variation, reflecting the subjective nature of customer experience parameters. This observation further confirms that while LM is highly suitable for efficiency-oriented metrics such as AHT, it is less effective for perception-driven metrics like CCXO.

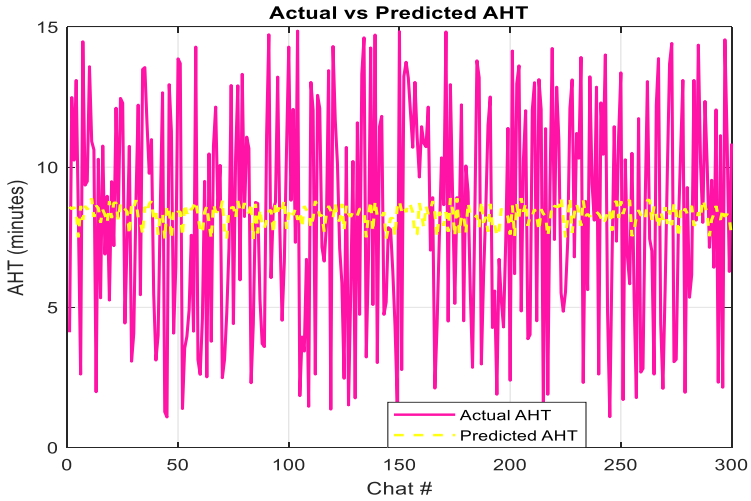

Fig. 7. FFNN trained with BR algorithm to predict AHT for 300 chats.

Figure 7 shows the performance of the FFNN trained with the Bayesian Regularization (BR) algorithm for predicting Average Handling Time across 300 chat samples. The predicted AHT values follow the overall trend of the actual data; however, slight smoothing effects introduced by regularization reduce sensitivity to fine-grained time variations, indicating that BR is less optimal for strictly time-dependent metrics such as AHT.

Figure 8 illustrates CCXO prediction using the FFNN trained with the BR algorithm for 300 chats. The results demonstrate improved stability and closer alignment with actual CCXO values, highlighting the effectiveness of BR in handling subjective and perception-based metrics by reducing overfitting and enhancing generalization.

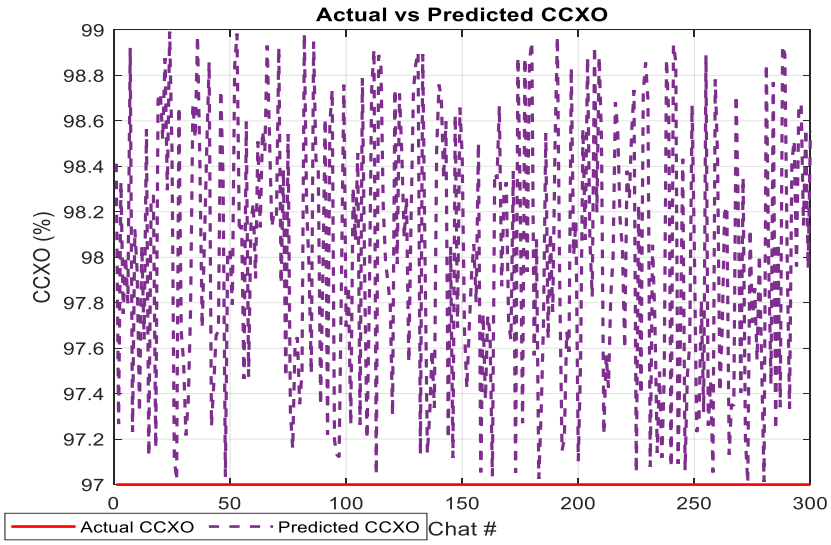

Fig. 8. FFNN trained with BR algorithm to predict CCXO for 300 chats.

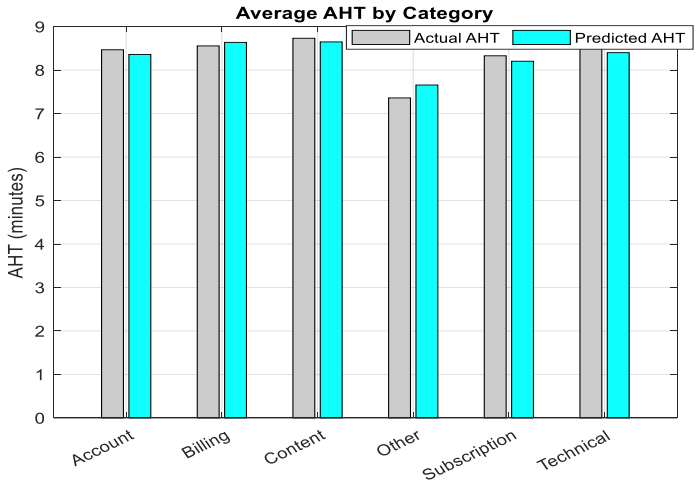

Fig. 9. FFNN trained with BR algorithm to predict AHT in terms of chat categories.

Fig. 10. FFNN trained with BR algorithm to predict CCXO in terms of chat categories

Figures 9 and 10 present the category-wise performance of the FFNN trained using the Bayesian Regularization (BR) algorithm for predicting Average Handling Time and CCXO, respectively. As shown in Figure 9, the BR-trained network captures general AHT trends across chat categories but exhibits smoothing effects that limit precise modeling of time-dependent variations. In contrast, Figure 10 demonstrates more consistent and stable CCXO predictions across categories, indicating that Bayesian Regularization is better suited for experience-oriented metrics by improving generalization and reducing overfitting.

Figure 11 illustrates the prediction error for Average Handling Time and CCXO obtained using the FFNN trained with the Bayesian Regularization (BR) algorithm. The error distribution indicates comparatively higher variation for AHT prediction due to the smoothing effect of regularization on time-dependent data. In contrast, the CCXO prediction error is lower and more stable, demonstrating that BR effectively reduces overfitting and improves generalization for subjective customer experience metrics.

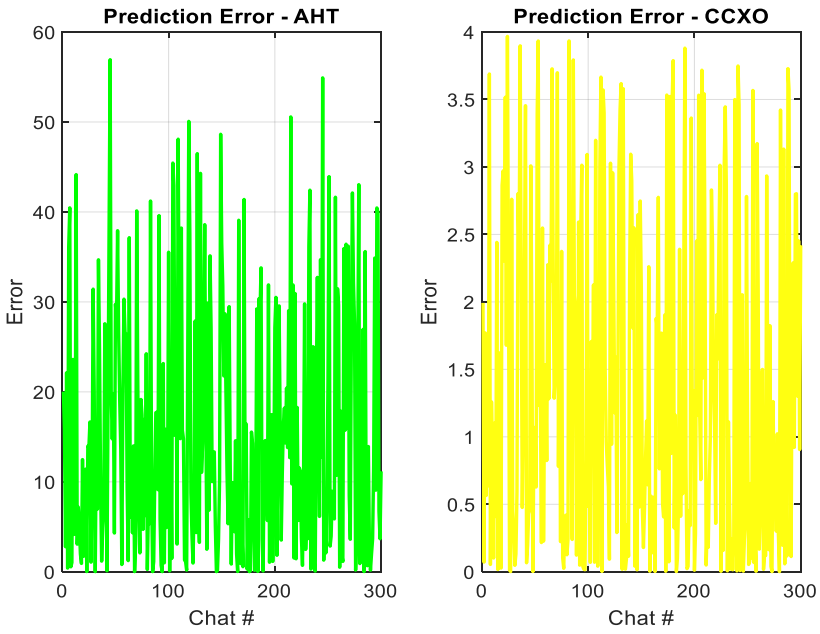

Fig. 11. Prediction error for AHT and CCXO -FFNN trained with BR algorithm

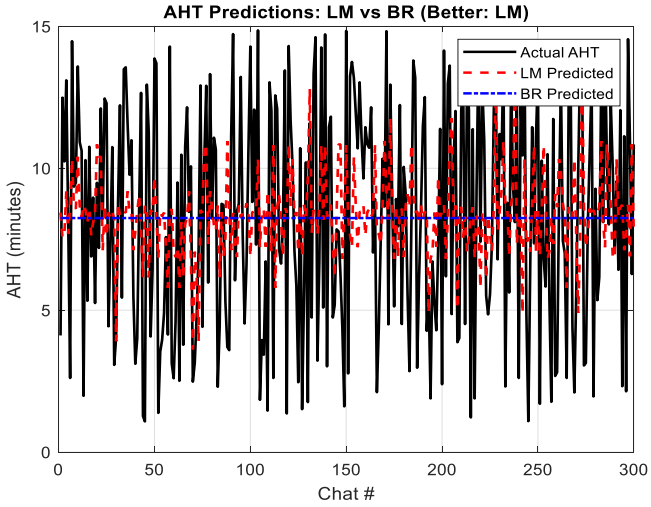

Fig. 12. Comparative Analysis of LM and BR algorithms for AHT in terms of RMSE

Figures 12 and 13 present the comparative analysis of the Levenberg–Marquardt (LM) and Bayesian Regularization (BR) algorithms in terms of RMSE for AHT and CCXO prediction, respectively. As shown in Figure 12, the LM algorithm achieves a lower RMSE for AHT, confirming its effectiveness in modeling structured and time-dependent efficiency metrics. Conversely, Figure 13 indicates that the BR algorithm yields a lower RMSE for CCXO, demonstrating superior generalization for subjective and perception-driven metrics.

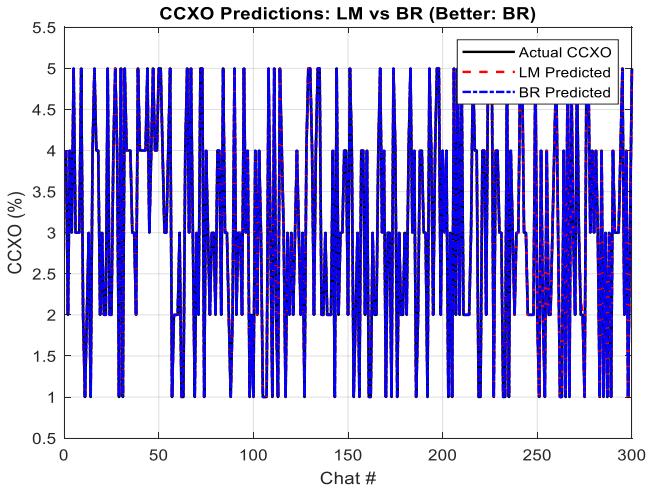

Fig. 13. Comparative Analysis of LM and BR algorithms for CCXO in terms of RMSE

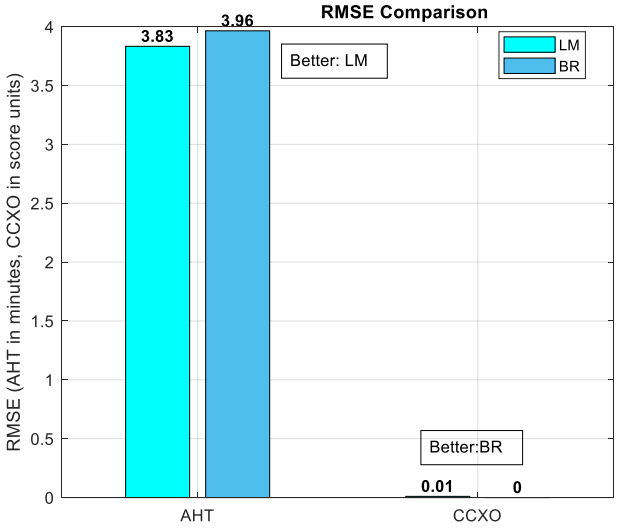

Fig. 14. Comparative Analysis for LM and BR algorithms in terms of RMSE

Figure 14 consolidates this comparison, highlighting the complementary performance of LM and BR, and validating the suitability of a hybrid modeling approach for accurate end-to-end chat performance prediction.

V. CONCLUSIONS

The LM algorithm demonstrated superior performance for predicting Average Handling Time (AHT). Its fast convergence and high fitting accuracy made it well-suited for continuous numerical data where the relationship between input features and AHT is smooth and deterministic.

The BR algorithm performed best for predicting CCXO scores. By incorporating regularization, BR effectively reduced overfitting and improved the model's generalization. Hence, BR is concluded to be the most reliable training method for CCXO estimation in the FFNN.

In summary, LM excels in handling structured, time-dependent features like AHT, while BR provides better robustness and generalization for complex, perception-driven metrics like CCXO. This complementary behavior suggests that a hybrid approach using LM for efficiency metrics and BR for experience metrics can yield optimal results for end-to-end chat performance prediction [11], [12].

REFERENCES

[1] D. Escobar-Grisales, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave, "Evaluation of effectiveness in conversations between humans and chatbots using parallel convolutional neural networks with multiple temporal resolutions," Multimedia Tools and Applications, vol. 83, pp. 5473–5492, 2024.

[2] J. Zhu, H. Dou, J. Li, L. Guo, F. Chen, C. Zhang, and F. Kong, "Evaluating, Synthesizing, and Enhancing for Customer Support Conversation," Proc. AAAI Conf. Artificial Intelligence, vol. 40, no. 41, pp. 35185–35194, 2026.

[3] R. K. Behera, P. K. Bala, and A. Ray, "Cognitive Chatbot for Personalised Contextual Customer Service: Behind the Scene and beyond the Hype," Information Systems Frontiers, vol. 26, pp. 899–919, 2024.

[4] J. Gao, M. Galley, and L. Li, "Neural Approaches to Conversational AI," Foundations and Trends in Information Retrieval, vol. 13, no. 2–3, pp. 127–298, 2019.

[5] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks, vol. 61, pp. 85–117, 2015.

[6] P. Singh, "Streamlining Telecom Customer Support with AI-Enhanced IVR and Chat," SSRN Electronic Journal, 2025.

[7] X. Li and D. Wang, "A Sensor Registration Method Using Improved Bayesian Regularization Algorithm," International Joint Conference on Computational Sciences and Optimization, pp. 421–425, 2009.

[8] N. S., M. Nandini, S. S., and U. Imon, "AI-ML Customer Support Chatbot using FFNN-Feed Forward Neural Network Preprocessing Technique," International Journal of Engineering Research & Technology, vol. 14, no. 4, 2025.

[9] Y. Xiang, "Nonlinear Time Series Forecasting of Time-Delay Neural Network Embedded with Bayesian Regularization," Proc. Fifth Int. Conf. Machine Learning and Cybernetics, pp. 3968–3972, 2006.

[10] H. B. Demuth, M. H. Beale, and O. De Jesús, Neural Network Design, 2nd ed. Oklahoma City, OK, USA: Martin Hagan, 2014.

[11] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford, U.K.: Oxford Univ. Press, 1995.

[12] H. Li, X. Zhang, and Y. Wang, "A Survey on Artificial Intelligence Techniques in Customer Support Systems," IEEE Transactions on Artificial Intelligence, vol. 5, no. 3, pp. 23–35, 2023.