AI Based Sentiment Analysis of Incoming Calls on Helpdesk

[1]T. Rani, [2]Komarneni Veera Venkata Sai, [3]Rangisetti Jagadish, [4]Duddu Avinash, [5]Komarabathina Praharsh

[1]Assistant Professor, Dept of CSE-AI&ML, St. Ann's College of Engineering and Technology Chirala-523187, India.

[2,3,4,5]U.G. Students, Dept of CSE-AI&ML, St. Ann's College of Engineering and Technology Chirala-523187, India.

ABSTRACT- Customer helpdesks handle a large volume of incoming calls every day, and the emotional tone of these conversations is one of the strongest indicators of service quality and customer satisfaction. Manually monitoring every call for tone and sentiment is impractical, time-consuming, and highly subjective, which often causes dissatisfied customers to go unnoticed until they escalate their complaints or discontinue the service. This paper presents an AI-Based Sentiment Analysis of Incoming Calls on Helpdesk system that converts helpdesk call transcripts into a machine-readable form and classifies the underlying customer sentiment as Positive, Neutral, or Negative using a fine-tuned Bidirectional LSTM deep learning model. Before classification, the call text undergoes NLP preprocessing such as text cleaning, tokenization, stop-word removal, and sequence padding using NLTK to improve prediction accuracy. In addition to sentiment classification, the system estimates an escalation severity level (Low, Medium, High) based on the intensity of the detected negative sentiment, enabling supervisors to intervene in real time before a dissatisfied customer disconnects the call. The proposed Flask-based system with an SQLite backend enables transcript submission, dataset management, model training, and sentiment monitoring. The Bidirectional LSTM model achieves 87–90% prediction. The system provides a scalable, user-friendly, and cost-effective solution for automated helpdesk call quality monitoring.

***Keywords:** Sentiment Analysis, Deep Learning, Long Short-Term Memory (LSTM), Natural Language Processing, Helpdesk Analytics, Escalation Management, Flask, Customer Service.*

INTRODUCTION

Customer support and helpdesk services form the backbone of modern organizations, acting as the primary channel through which customer concerns, complaints, and queries are resolved. Every incoming call carries not only factual information but also an emotional tone that reflects how the customer truly feels about the service being provided, and this tone has a direct influence on customer satisfaction, retention, and brand reputation. Traditionally, gauging this emotional tone has depended on manual call monitoring by supervisors, a process that is slow, resource-intensive, and able to cover only a small fraction of the total call volume. As a result, many dissatisfied or frustrated customers go unnoticed until they escalate their complaints or discontinue using the service altogether.

With the advancement of Artificial Intelligence (AI), Natural Language Processing (NLP), and Deep Learning, it has become possible to automatically analyze the sentiment of a conversation in real time. This work presents an AI-Based Sentiment Analysis of Incoming Calls on Helpdesk system that uses a Long Short-Term Memory (LSTM) based deep learning model to classify the sentiment expressed in helpdesk call transcripts as Positive, Neutral, or Negative. The call audio is first converted into text using an Automatic Speech Recognition (ASR) engine. The transcribed text is cleaned and preprocessed using NLP techniques such as tokenization, stop-word removal, and padding before being passed to the trained LSTM model for classification. Based on the severity of the negative sentiment detected, the system automatically raises an escalation alert on the supervisor's dashboard, enabling proactive intervention.

The developed system is implemented as a Flask-based web application in which agents can register, log in, submit call transcripts or audio, and instantly receive sentiment predictions along with confidence scores and suggested response actions. An administrator module is also provided to manage datasets, preprocess data, train the model, manage agents, and monitor overall system performance and sentiment trends.

RELATED WORK

Sentiment analysis, also referred to as opinion mining, has been an active area of research at the intersection of Natural Language Processing and Machine Learning. Earlier approaches to sentiment classification depended largely on lexicon-based and rule-based methods, which classify text by counting positive and negative keywords. Although computationally simple, these approaches fail to capture context, negation, and sarcasm, which are common in conversational speech.

Traditional Machine Learning techniques such as Naive Bayes, Support Vector Machine (SVM), and Logistic Regression improved upon lexicon-based methods by learning from manually engineered features such as TF-IDF and bag-of-words representations. However, their performance is closely tied to the quality of hand-crafted features and they struggle to generalize to informal, conversational call transcripts.

More recently, Deep Learning approaches, particularly Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and transformer-based architectures, have significantly improved sentiment classification accuracy by automatically learning sequential language patterns directly from raw text, without manual feature engineering. These models are able to capture long-range dependencies, negation, and contextual cues that are essential for accurate sentiment understanding in real conversations.

In parallel, there is a growing need for intelligent systems that go beyond offline sentiment labelling and provide practical, real-time support to helpdesk supervisors. Integrating sentiment prediction with automated escalation mechanisms can help organizations respond to dissatisfied customers before the relationship is damaged. Motivated by this, the present work combines deep learning-based sentiment detection with real-time escalation alerting through an easy-to-use web application.

LITERATURE SURVEY

Although the above studies achieve reasonably high sentiment classification accuracy, several gaps remain in existing literature. Most research focuses only on offline sentiment classification of generic text or product reviews rather than live helpdesk conversations. There is limited support for real-time, web-based helpdesk monitoring applications, and very few systems provide automated escalation alerts based on sentiment severity. In addition, existing models are typically evaluated on controlled, clean datasets rather than real call transcripts, and actionable agent-facing recommendations are rarely integrated into sentiment detection systems.

To address these gaps, the proposed system combines a deep learning-based sentiment classifier with a Flask web application, real-time escalation alerts, and supervisor analytics, providing a practical and user-friendly solution for modern helpdesk teams.

EXISTING METHOD

Customer helpdesks currently rely on a combination of manual and semi-automated approaches to assess call quality and customer sentiment. In manual call auditing, supervisors listen to a small, randomly sampled subset of recorded calls and manually score agent tone and customer satisfaction based on personal judgement. This process is slow, expensive, and can only cover a small fraction of the total call volume, meaning most dissatisfied interactions are never reviewed.

Rule-based and lexicon sentiment analysis tools classify text by counting the occurrence of positive and negative words from a predefined dictionary. While simple to implement, such systems fail to capture context, sarcasm, and negation, and their accuracy drops sharply on informal, conversational language typical of helpdesk calls. Traditional Machine Learning techniques, such as Naive Bayes, SVM, and Logistic Regression, classify sentiment using manually engineered features like TF-IDF and bag-of-words. These methods perform moderately well on clean datasets but require significant feature-engineering effort and do not scale well to the variability found in real call transcripts.

Some existing systems apply Recurrent Neural Networks or basic LSTM networks to learn sequential features automatically, generally achieving better accuracy than traditional ML methods. However, most of these implementations remain research prototypes focused purely on offline classification, and they typically lack a real-time, agent-facing web interface, escalation alerting, or supervisor analytics, which limits their practical usefulness in day-to-day helpdesk operations.

PROPOSED METHOD

The proposed AI-Based Sentiment Analysis of Incoming Calls on Helpdesk system automatically detects customer sentiment from call transcripts using a fine-tuned Bidirectional LSTM deep learning model, combined with an NLP preprocessing pipeline and a Flask-based web interface. The overall goal is to provide supervisors with fast, accurate, and automated sentiment insights while enabling proactive escalation of at-risk conversations, without requiring manual call auditing. When an agent submits a call transcript (or audio, which is converted to text using an ASR engine), the text is first cleaned by removing punctuation and special characters, converted to lowercase, tokenized, and filtered to remove stop-words using NLTK. The cleaned text is then converted into numerical sequences and padded to a fixed length before being passed to the trained sentiment model.

The classification model is built around a Bidirectional LSTM architecture. An Embedding layer first converts each word in the input sequence into a dense vector representation. The embedded sequence is then processed by a Bidirectional LSTM layer, which learns contextual dependencies from both past and future words in the sequence, allowing the model to capture negation, intensity, and context that are critical for accurate sentiment understanding. A Dropout layer reduces overfitting, and a Dense (fully connected) layer combines the extracted sequential features before a final Softmax layer predicts the probability of each sentiment class (Positive, Neutral, Negative).

The predicted sentiment and its confidence score are then used to compute an escalation severity level. If the predicted sentiment is Negative with a high confidence score, the call is flagged as High severity; a Negative prediction with moderate confidence is flagged as Medium severity; all other cases are treated as Low severity. Based on this severity level, the application displays suggested response actions to the agent and raises a real-time alert on the supervisor's dashboard for calls that require immediate attention.

SYSTEM ARCHITECTURE

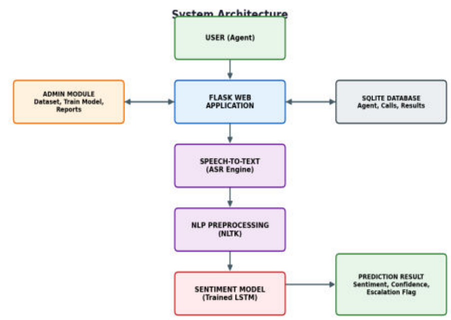

Fig. 1. System architecture of the proposed AI-based helpdesk sentiment analysis application.

Figure 1 illustrates the overall system architecture. Agents interact with the application through the Flask web interface; after authentication, submitted transcripts pass through the speech-to-text and preprocessing modules before being analyzed by the trained LSTM model. The prediction results, confidence scores, escalation severity, and recommended actions are then returned to the agent and stored in the SQLite database, while the administrator manages datasets, retrains the model, and monitors sentiment trends and escalation history through the admin dashboard.

A. LSTM Sentiment Model Architecture

Figure 2 shows the internal architecture of the sentiment classification model. The preprocessed transcript is converted into a fixed-length sequence and passed through an Embedding layer, a Bidirectional LSTM layer, a Dropout layer, a Dense layer, and finally a Softmax output layer that produces the sentiment class probabilities. This layered structure allows the model to automatically learn sequential language features directly from raw conversational text, without manual feature engineering.

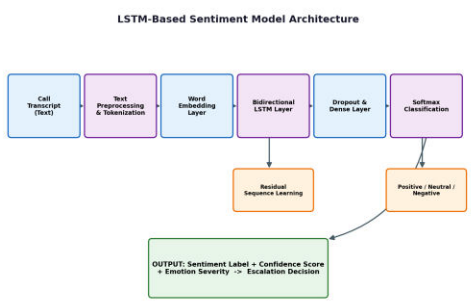

Fig. 2. Bidirectional LSTM sentiment classification model architecture.

MODULE DESCRIPTION

1. Agent Authentication Module

Provides secure registration and login for helpdesk agents. Credentials are stored in the SQLite database, ensuring that only authenticated agents can submit transcripts and view predictions.

2. Transcript / Audio Submission Module

Allows agents to submit a call transcript directly or upload call audio, which is converted to text using an Automatic Speech Recognition (ASR) engine before further processing.

3. Text Preprocessing Module

Cleans and prepares the transcript for classification through lowercasing, punctuation removal, tokenization, stop-word removal, and sequence padding using NLTK.

4. Sentiment Prediction Module

Passes the preprocessed sequence through the trained Bidirectional LSTM model to classify the transcript as Positive, Neutral, or Negative, along with a confidence score.

5. Escalation & Agent Assistance Module

Computes an escalation severity level (Low, Medium, High) from the predicted sentiment and confidence score, displays recommended response actions to the agent, and raises a real-time alert on the supervisor's dashboard for severely negative calls.

6. Administrator Module

Enables dataset upload and preprocessing, model training/retraining, agent management, and monitoring of organization-wide sentiment trends and escalation history through visual reports.

7. Database Module

Stores agent registration details, login credentials, call transcripts, prediction history, and escalation records using SQLite, ensuring secure and efficient data storage and retrieval.

RESULTS AND DISCUSSION

The proposed system was implemented using Python, Flask, TensorFlow/Keras, NLTK, and SQLite, and evaluated using a labelled helpdesk call transcript dataset. The application was tested for agent authentication, transcript submission, sentiment prediction, escalation alerting, and database operations, and all major functionalities performed as expected under unit, integration, system, and functional testing.

The trained Bidirectional LSTM model successfully classified customer sentiment from submitted call transcripts and generated prediction results along with confidence scores in a few seconds, while maintaining a smooth and responsive user experience. Table II summarizes the observed performance of the proposed system.

Parameter	Observation
Prediction Accuracy	High (approximately 87-90% on test data)
Response Time	A few seconds per transcript
User Interface	User-friendly, web-based
Sentiment Detection	Accurate across Positive, Neutral, and Negative classes
Escalation Alerting	Real-time, based on severity level
Database Performance	Efficient (SQLite)
Overall System Performance	Satisfactory for practical helpdesk monitoring

TABLE I. PERFORMANCE ANALYSIS

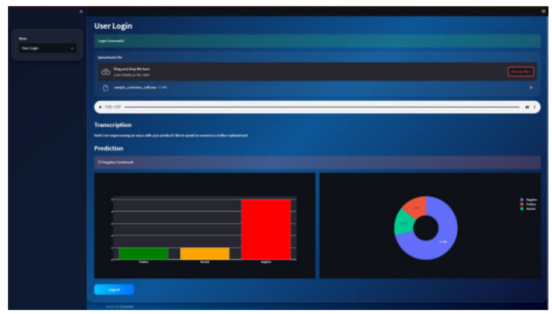

Fig. 3. Sentiment prediction, confidence score, and escalation-assistance result.

Figure 3 shows a sample sentiment prediction and escalation-assistance result generated by the application, where the predicted sentiment, confidence score, and severity level are displayed to the agent along with recommended response actions. The escalation module raises an immediate alert on the supervisor's dashboard whenever a severely negative sentiment is detected, allowing timely intervention before the customer disconnects the call.

The results confirm that the integration of Flask, NLTK, TensorFlow/Keras, and SQLite provides an efficient platform for real-time helpdesk sentiment detection. Compared to manual call auditing and lexicon-based approaches, the proposed Bidirectional LSTM-based system offers considerably higher coverage, faster turnaround, and an integrated escalation mechanism that existing approaches generally lack.

CONCLUSION

This paper presented an AI-Based Sentiment Analysis of Incoming Calls on Helpdesk system that assists helpdesk teams in identifying customer sentiment through deep learning. The system integrates Python, Flask, TensorFlow/Keras, NLTK, and SQLite to provide an efficient and user-friendly platform for real-time sentiment detection.

The trained Bidirectional LSTM model accurately classifies customer sentiment from submitted call transcripts and generates prediction results along with confidence scores. In addition, the system estimates escalation severity and provides real-time alerts, helping supervisors make informed decisions regarding call intervention and customer retention. The successful integration of natural language processing, deep learning, and web technologies demonstrates the effectiveness of Artificial Intelligence in modern customer service operations, and the developed system provides a reliable foundation for improving helpdesk quality management.

FUTURE SCOPE

Future enhancements include integrating transformer-based models like BERT, real-time streaming sentiment detection, and voice tone analysis (pitch, pace, stress) alongside mobile alert applications for supervisors. The system can also be expanded to support multiple regional languages, code-mixed speech, and integration with live CRM/IVR systems for automated, enterprise-scale deployment. Additionally, incorporating agent coaching recommendations based on recurring negative sentiment patterns would further enhance the system's practical and operational value.

REFERENCES

[1] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1-135, 2008.

[2] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural

Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[4] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.

[5] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.

[6] K. P. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[7] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. O'Reilly Media, 2022.

[8] T. Mikolov et al., "Distributed Representations of Words and Phrases and their Compositionality," Advances in Neural Information Processing Systems (NeurIPS), 2013.

[9] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," Proceedings of EMNLP, pp. 1532-1543, 2014.

[10] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), 2017.

[11] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 265-283, 2016.

[12] L. Devillers, L. Vidrascu, and L. Lamel, "Challenges in Real-Life Emotion Annotation and Machine Learning Based Detection," Neural Networks, vol. 18, no. 4, pp. 407-422, 2005.

[13] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. O'Reilly Media, 2018.

[14] NLTK Project, "Natural Language Toolkit Documentation," Available: https://www.nltk.org

[15] TensorFlow Team, "TensorFlow Documentation," Available: https://www.tensorflow.org