Contents lists available at openscie.com
E-ISSN: 2961-7952
Open Global Scientific Journal
DOI: 10.70110/ogsj.v3i2.70
Journal homepage: https://openglobalsci.com

Performance Analysis of Fine-Tuned ChatGPT-3.5 Chatbot for Alni Accessories E-Commerce Service

Sugiarti[1], Ihwana As'ad[1]*, Al Hilaluddin[1]

[1] Faculty of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia

*Correspondence E-mail: ihwana.asad@umi.ac.id

ARTICLE INFO

Article History:
Received 22 September 2024
Revised 29 October 2024
Accepted 10 November 2024
Published 14 November 2024

Keywords:
Chatbot performance,
ChatGPT-3.5,
E-commerce service,
Fine-tunin,
Natural language processing.

ABSTRACT

Background: The rapid advancement of artificial intelligence (AI) technology, particularly in language modeling, has driven the adoption of chatbots to support e-commerce services. Alni Accessories store faces challenges related to limited product information and slow customer service, which negatively impact user satisfaction.
Aims: This study aims to analyze the performance of a fine-tuned ChatGPT-3.5-based chatbot to improve interaction quality and transaction efficiency on the e-commerce platform.
Methods & Results: The evaluation process was carried out through a series of experiments, where the 13th experiment achieved the best results with a Training Loss of 0.3894 and a Validation Loss of 0.5787. The Training Mean Token Accuracy of 0.8799 and Validation Mean Token Accuracy of 0.7971 indicate the model's ability to effectively learn language patterns. Furthermore, the Full Validation Loss of 0.6242 and Full Validation Mean Token Accuracy of 0.8122 demonstrate the model's strong generalization capability. The independent t-test produced a p-value of 0.0001, confirming a statistically significant difference in transaction response time between the old system and the new system. The findings of this study show that the fine-tuned ChatGPT-3.5 chatbot not only accelerates services and improves product information accuracy but also holds great potential for implementation in other sectors such as education and healthcare.

To cite this article: Sugiarti, As'ad, I., Hilaluddin, A. (2024). Performance analysis of a fine-tuned chatgpt-3.5 chatbot for alni accessories e-commerce service. *Open Global Scientific Journal*, 3(2), 83–93.

1. Introduction

The development of e-commerce technology has become a driving force behind the transformation of modern business. With its ease of access and broad market reach, e-commerce enables businesses to

conduct sales without geographical constraints, accelerate and improve the efficiency of transaction processes, and simultaneously serve as an effective promotional medium (Hairi *et al.*, 2020). Alni Accessories, located at Jalan Barukang Utara Lorong 8 No. 10, Cambayya, Makassar, is a small business specializing in titanium-based accessories. Its current sales strategy relies primarily on direct sales and simple online platforms such as WhatsApp and Facebook. However, challenges arise due to incomplete product information and slow customer service response times ranging from 6 minutes to 5 hours which consequently delay the ordering process and reduce customer satisfaction.

To address these challenges, a mobile-based e-commerce solution is required to deliver a more interactive and efficient shopping experience. One proposed approach is to integrate a fine-tuned ChatGPT-3.5 Turbo chatbot that can better understand the business context, provide accurate product information, and respond to customers more quickly. Previous studies have shown that fine-tuning large language models (LLMs) using company-specific datasets significantly improves response accuracy in corporate settings, as the model becomes more relevant to the specific business domain (Kishore, 2024). In addition, fine-tuning GPT-3.5 Turbo on e-commerce domain datasets has been proven effective in enhancing text-based product recommendation systems (Xu & Hu, 2023). A DevOps approach is also applied to ensure the system is developed in a sustainable, efficient, and scalable manner (Septiadi & Isnandar, 2024).

Earlier research has highlighted the effectiveness of chatbots in delivering information for example, in new student admission services (PMB) at STT-NF, where a chatbot successfully assisted prospective students in obtaining information via Telegram API and Python, and was proven effective through Black Box Testing, User Acceptance Testing (UAT), and questionnaire surveys (Herfian & Adriansyah, 2021). Moreover, studies on ChatGPT have demonstrated its ability to deliver accurate and fast information, improve time efficiency, and support user creativity, although critical evaluation of its outputs remains necessary (Misnawati, 2023).

In the e-commerce context, a reliable, responsive, and secure chatbot has been shown to improve customer satisfaction (Kappi & Marlina, 2023). The DevOps approach is widely adopted in software research because it accelerates the development and operations cycle without compromising system quality (Wibowo *et al.*, 2023). Fine-tuning language models such as ChatGPT has also been reported to significantly improve performance by adapting them to specific domains, resulting in more relevant outputs, efficient token usage, and reduced request latency (OpenAI, 2024).

Based on the challenges faced by Alni Accessories and the findings of previous research, this study aims to develop a mobile e-commerce application integrating a fine-tuned ChatGPT-3.5 Turbo chatbot with a DevOps approach. This combination is expected to enhance service quality, accelerate the ordering process, and increase customer satisfaction, thereby maximizing overall e-commerce performance.

2. Methods

a. DevOps Method

The DevOps method is a developer-oriented principle designed to effectively and efficiently coordinate collaboration between development teams and operations teams (Kole & Sugeng, 2021). DevOps aims to enhance collaboration between these teams, from planning to the successful delivery of applications or features to end users. This approach helps minimize communication barriers between teams, thereby accelerating and standardizing the software development cycle.

Furthermore, DevOps emphasizes the implementation of Continuous Integration (CI) and Continuous Delivery (CD), which enable development, testing, and deployment processes to run automatically and continuously. This not only shortens product release cycles but also improves software quality by allowing potential errors to be detected and resolved more quickly.

In addition, the DevOps approach supports organizational agility, enabling companies to be more adaptive to market demands and technological changes (Fitriani *et al.*, 2023). Thus, DevOps is not merely a technical methodology but also a work culture that emphasizes collaboration, transparency, and continuous improvement.

The key steps in implementing DevOps are as follows:

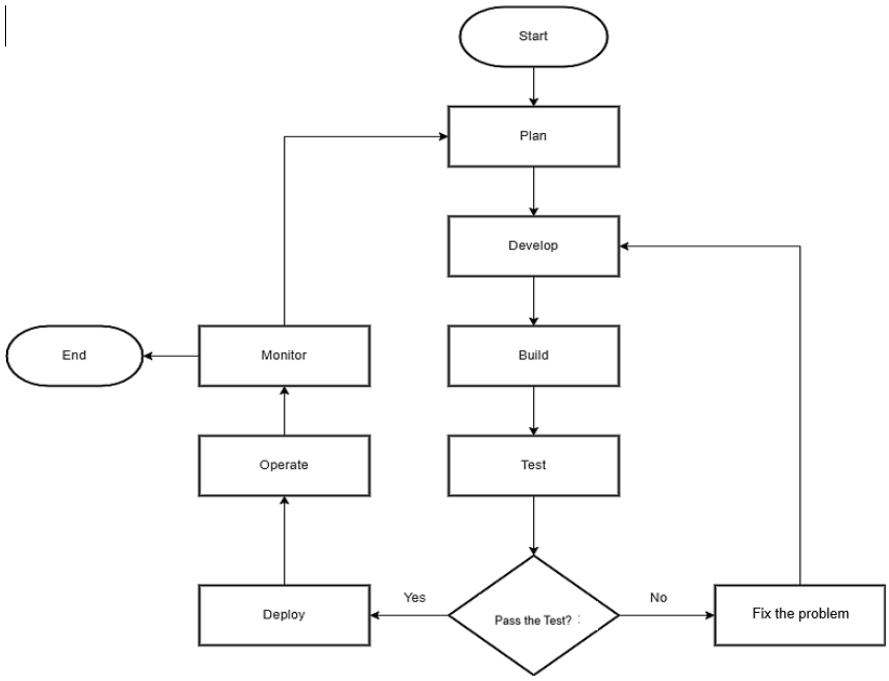


Figure 1. DevOps Method Stages

1. Plan

In the planning stage, system requirements are identified and designed using UML diagrams. Development management is handled through Google Cloud and GitHub for seamless collaboration. The data sources used at this stage include user requirement documents, e-commerce transaction records, and preliminary chatbot interaction datasets, which serve as the foundation for system design.

2. Develop

At this stage, the development team is divided into three groups: backend, frontend, and DevOps. The backend team is responsible for developing APIs and implementing business logic, while the frontend team focuses on building the user interface. The DevOps team manages cloud infrastructure and automates the deployment process. All development activities are carried out through close team collaboration, with version control and code collaboration managed via GitHub. The data sources include product catalog data, user profile information, and training datasets used to build and refine the chatbot model.

3. Build

This stage integrates the software modules for both the e-commerce platform and chatbot, with automated builds managed through GitHub Actions within the CI/CD pipeline. Each commit is tested before being merged into the main branch, producing software artifacts ready for testing. The data sources include integrated API endpoints, chatbot model checkpoints, and configuration files necessary for deployment.

4. Test

Testing is performed using the Independent T-Test method to compare response times between the manual system and the e-commerce platform. Google Cloud Monitoring is used to observe performance

and response time in real-time, while Postman validates APIs and chatbot functionality. If testing results indicate discrepancies such as suboptimal response times or API errors the team revisits the Develop stage, followed by Build and re-testing until the system meets performance requirements. The data sources include response time logs, API call records, and user interaction logs used to evaluate chatbot accuracy and system efficiency.

5. Deploy

After successful testing, the software is deployed to the Google Cloud infrastructure. Deployment is executed through the CI/CD pipeline using Google Cloud Build to ensure a smooth and consistent release process. The data sources include deployment scripts, infrastructure configuration files, and monitoring dashboards for verifying a successful rollout.

6. Operate

Operational management of the production application is handled using Google Cloud Operations Suite to manage logs, monitor application health, and apply autoscaling when necessary. Data sources at this stage include system health metrics, error logs, and scaling activity records, which help maintain optimal system performance.

7. Monitor

Real-time monitoring is performed using Google Cloud Monitoring and Logging to detect issues early. Chatbot interaction data is analyzed to continuously improve customer service quality. Data sources include chatbot conversation logs, user feedback surveys, and performance metrics extracted from the production environment.

2.2 Fine-tuning

Fine-tuning a large language model (LLM) for specific tasks allows the model to fully utilize its capacity by adapting to a particular domain (J *et al.*, 2024). The benefits of fine-tuning include efficiency, as a pre-trained model requires less data and time for task-specific training compared to building a model from scratch (Brown *et al.*, 2020). Therefore, this study adopts a fine-tuning approach to enhance model performance in domain-specific tasks. The fine-tuning process in this research follows these steps:

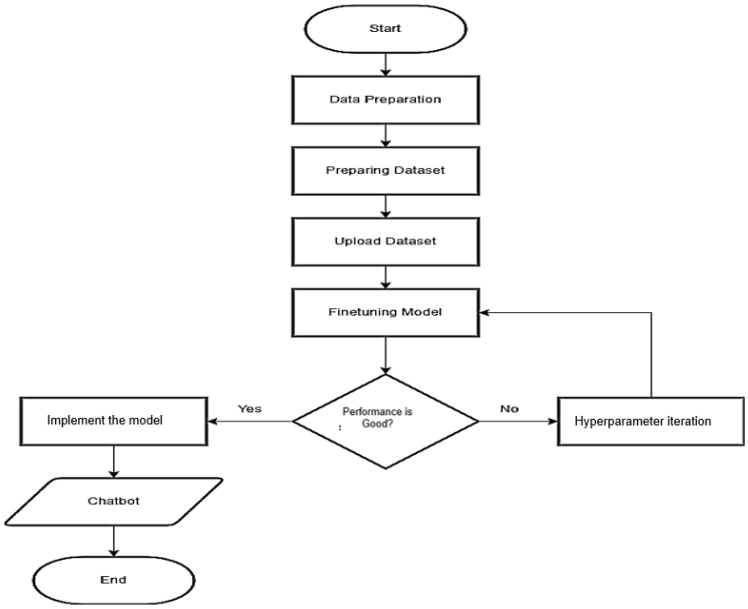


Figure 2. Fine-Tuning Process Flowchart

1. Data Preparation
   Data preparation is a crucial initial step in the fine-tuning process, which includes data collection and preprocessing. Data collection focuses on obtaining relevant question–answer conversations in the context of titanium accessory e-commerce, such as transaction histories from WhatsApp, Messenger, or external sources. Preprocessing is then performed to remove noise, correct typographical errors, and eliminate duplicates, ensuring high-quality data for model training.
2. Preparing the Dataset
   This stage involves prompt engineering, dataset splitting, and format adjustment for fine-tuning. In this study, the dataset consists of 70 samples, which are split into 80% for the training set and 20% for the validation set. Prompts are designed based on common interaction patterns, such as product inquiries and recommendations. The dataset is then divided into training and validation sets and converted into JSON Lines (JSONL) format to ensure compatibility with the fine-tuning platform.
3. Upload Dataset
   After formatting, the dataset is uploaded to the OpenAI platform for fine-tuning. The dataset files are structured to meet the platform's technical requirements, enabling successful training of the GPT-3.5 Turbo model.
4. Fine-tuning
   This process involves configuring training parameters such as epoch count, learning rate, and batch size to prevent overfitting, as well as validating the JSONL format and ensuring the alignment of prompt–completion pairs before execution.

3. Results and Discussion

3.1 Fine-tuning Results

This study conducted fifteen fine-tuning experiments, the results of which are summarized in Table 1.

Table 1. Hyperparameter Configurations

Experiment	Batch Size	Learning Rate	Epochs
Experiment 1	Auto	Auto	Auto
Experiment 2	8	0.05	10
Experiment 3	16	0.1	7
Experiment 4	4	0.01	10
Experiment 5	4	0.05	3
Experiment 6	3	0.07	3
Experiment 7	7	0.05	8
Experiment 8	8	0.05	12
Experiment 9	8	0.005	12
Experiment 10	8	0.06	10
Experiment 11	8	0.07	12

Experiment 12	4	2	8
Experiment 13	8	1	11
Experiment 14	8	1	16
Experiment 15	8	1.2	12

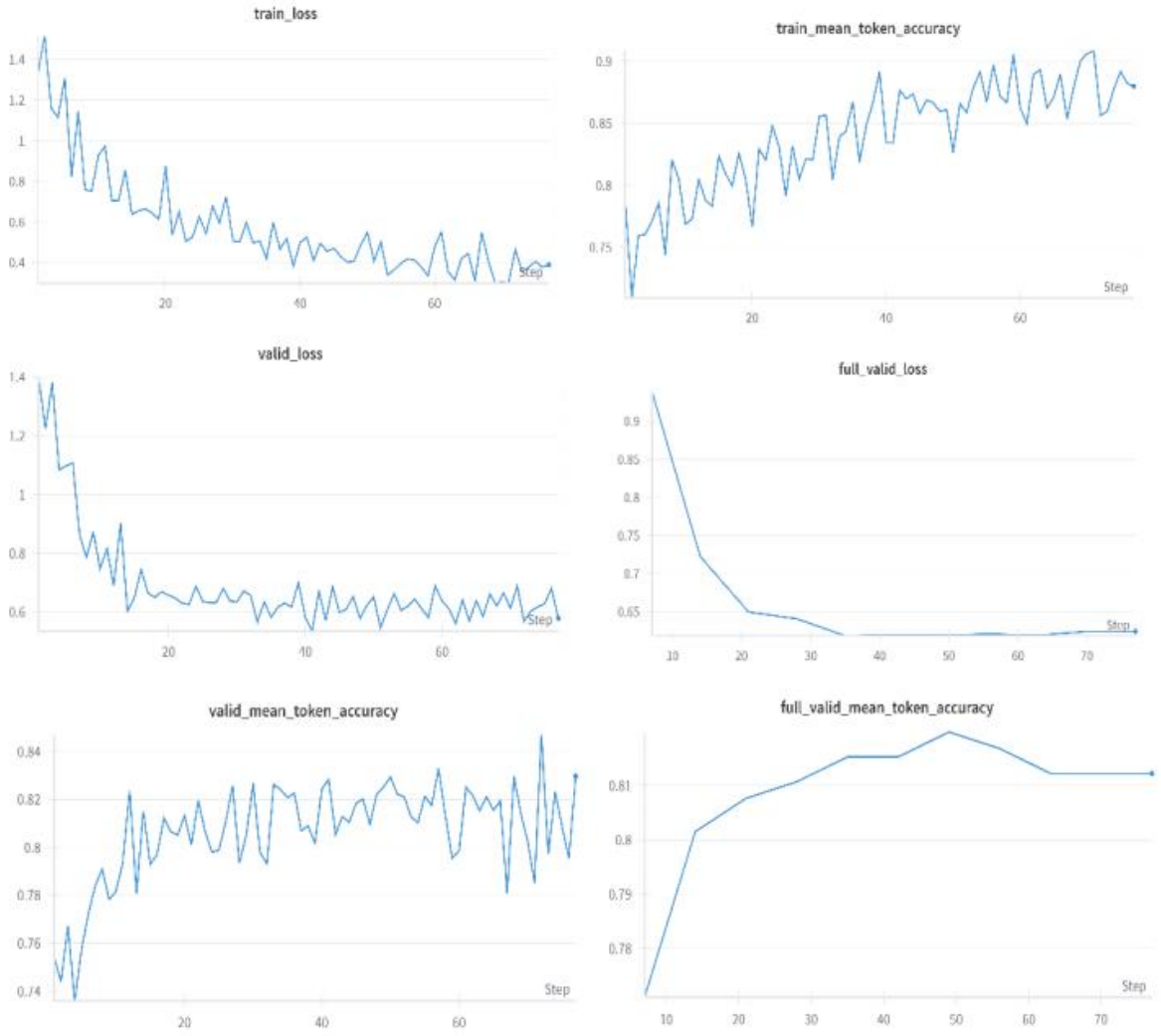
3.2. Fine-Tuning Results of ChatGPT

Table 2 presents the results of fifteen fine-tuning experiments conducted on the ChatGPT model. Each experiment was performed using different parameter configurations to evaluate model performance. The results include Training Loss, Training Mean Token Accuracy, Validation Loss, Full Validation Loss, Validation Mean Token Accuracy, and Full Validation Mean Token Accuracy.

Table 2. Fine-Tuning Experiment Results of ChatGPT

Experiment	Training Loss	Training Mean Token Accuracy	Validation Loss	Full Validation Loss	Validation Mean Token Accuracy	Full Validation Mean Token Accuracy
Experiment 1	0.3234	0.8838	0.6286	0.6582	0.7885	0.8122
Experiment 2	0.8033	0.7921	0.8500	0.9836	0.7915	0.7680
Experiment 3	1.1923	0.7893	0.9436	1.0445	0.7710	0.7633
Experiment 4	1.3278	0.7330	1.2764	1.1899	0.7900	0.7557
Experiment 5	1.0050	0.7722	1.3073	1.1253	0.7528	0.7588
Experiment 6	0.8224	0.7967	1.2834	0.9958	0.8000	0.7664
Experiment 7	0.9585	0.7897	0.9573	1.0230	0.7738	0.7648
Experiment 8	0.7234	0.8092	1.0273	0.9493	0.7601	0.7694
Experiment 9	1.1577	0.7830	1.2312	1.2638	0.7540	0.7526
Experiment 10	0.9485	0.7840	0.9550	0.9586	0.7630	0.7679
Experiment 11	0.9569	0.7568	0.8857	0.8680	0.7872	0.7801
Experiment 12	0.3435	0.9158	0.8148	0.7531	0.7857	0.8045
Experiment 13	0.3894	0.8799	0.5787	0.6242	0.7971	0.8122
Experiment 14	0.2087	0.9238	0.7393	0.6615	0.7990	0.8137
Experiment 15	0.2583	0.9173	0.6464	0.6370	0.8191	0.8107

Based on the fine-tuning experiments conducted, several metrics were used to evaluate the model's performance in each trial, including Training Loss, Training Mean Token Accuracy, Validation Loss, Full Validation Loss, Validation Mean Token Accuracy, and Full Validation Mean Token Accuracy. The following figure illustrates the results of Experiment 13, providing a comparative view of the metrics used in this evaluation.



From all experiments conducted, Experiment 13 demonstrated the most optimal results. The Training Loss for this experiment was 0.3894, which is considered relatively low, along with a low Validation Loss of 0.5787. This indicates that the model was able to effectively learn from the training data without showing signs of overfitting and was capable of generalizing well to the validation data.

Furthermore, the Training Mean Token Accuracy in Experiment 13 reached 0.8799, one of the highest among all experiments, indicating that the model achieved a very high level of accuracy in predicting tokens on the training data. Similarly, the Validation Mean Token Accuracy reached 0.7971, suggesting that the model was able to maintain strong accuracy on the validation set.

When compared with Experiment 14 which exhibited an extremely low Training Loss (0.2087) and high Training Mean Token Accuracy (0.9238) there was a notable difference in the Validation Loss, which was relatively high (0.7393). This suggests that although the model learned very well on the training data, it struggled to generalize on the validation data, a clear indication of overfitting. In contrast,

Experiment 13 successfully maintained a balance between Training Loss and Validation Loss, making it a more stable and efficient model.

In addition, the Full Validation Loss for Experiment 13 was recorded at 0.6242, which was lower than that of other experiments, indicating greater efficiency in reducing errors across the entire validation dataset. The Full Validation Mean Token Accuracy of 0.8122 further demonstrates the model's ability to accurately predict tokens across the full validation data.

Based on these results, it can be concluded that Experiment 13 achieved the best performance overall. This model demonstrates an ideal balance between low Training Loss and high Validation Accuracy, as well as optimal generalization capability, making it a more stable and reliable experiment compared to the others.

3.3 Transaction Speed Comparison

A transaction speed comparison was conducted by measuring the response time between the old system and the newly developed system. Data were collected from 15 respondents to determine the performance differences between the two systems.

Table 3. Transaction Time Data for Old and New Systems

Respondent	Transaction Time – Old System (Minutes)	Transaction Time – New System (Minutes)
1	280	8
2	200	8
3	72	10
4	261	7
5	6	7
6	248	10
7	46	11
8	271	9
9	112	8
10	250	14
11	107	9
12	118	13
13	63	11
14	94	7
15	32	13

3.3.1 Average Transaction Time for Each Group

Based on the data presented above, several statistical analyses were performed to compare the performance of the two systems. These analyses included the calculation of mean, variance, standard deviation, normality testing, and independent t-test. The average transaction times obtained from the data are as follows:

1. The average transaction time for the old system was 144 minutes.
2. The average transaction time for the new system was 9.67 minutes

3.3.2 Sample Variance Calculation

The sample variance and standard deviation were calculated as follows:

1) Sample Variance of the Old System

s_1^2= 9429,14 Menit

2) Sample Variance of the New System
   s_2^2= 5,38 Menit

3.3.3 Normality Test

Before conducting the statistical comparison test, a normality test was performed to ensure that the transaction time data from both systems followed a normal distribution. The Shapiro-Wilk method was employed for this purpose. The results indicated that the p-value for the old system was 0.0642, while the p-value for the new system was 0.1295. Since both p-values were greater than 0.05, it can be concluded that the data from both systems were normally distributed. Consequently, parametric statistical analysis, such as the independent t-test, could be applied.

3.3.4 Outlier Test

In addition to the normality test, the next step was to check for potential outliers that might influence the results. The outlier detection was performed using the z-score method, which compares each data point to the group mean and standard deviation. Data points with z-scores greater than 3 or less than -3 were considered outliers. The analysis confirmed that no significant outliers were present in the dataset. This finding, combined with the normality test results, allowed for the use of parametric statistical methods such as the independent t-test.

3.3.5 F Table Test

Following the normality and outlier tests, an F-test was conducted to determine whether the variances of the two groups could be assumed equal (homoscedasticity) or unequal (heteroscedasticity). The F-test compares the variances of the two groups to check for statistically significant differences. If the calculated F-value is greater than the F-table value, the null hypothesis (H_0) is rejected, indicating significantly different variances (unequal variance). Conversely, if the calculated F-value is less than or equal to the F-table value, H_0 is accepted, meaning the variances can be assumed equal.

3.3.6 *Independent t-Test*

In the program output, the F-test result indicated that the calculated F-value was greater than the F-table value, suggesting that the variances of the two groups were significantly different. Therefore, Welch's t-test was employed, which is a modification of the independent t-test that accounts for unequal variances between groups.

After confirming normal distribution, absence of outliers, and unequal variances between groups, Welch's t-test was performed to determine whether there was a statistically significant difference between the mean transaction times of the old and new systems.

The t-value was calculated as follows:

$$t = \frac{144 - 9,67}{\sqrt{\frac{9429,14}{15} + \frac{5,38}{15}}}$$

$$t = 5,35622$$

3.3.7 Determining the Degrees of Freedom (DF)

After obtaining the t-value, the degrees of freedom (DF) were calculated to find the critical value from the t-distribution table, which serves as a reference for determining statistical significance. Given: s_1^2= 9429,14, s_2^2=5,38, n_1 = 15, and n_2 = 15, the resulting degree of freedom: df = 14.0160.

3.3.8 Determining the p-Value

The next step was to calculate the p-value to test the significance of the difference in mean transaction times between the two systems. Based on the calculations, the p-value obtained was 0.0001, which is lower than 0.05. This result indicates that the difference in response times between the old and new systems is statistically significant. Therefore, the null hypothesis (H_0), which states that there is no significant difference between the two systems, can be rejected. This finding confirms that the new system demonstrates a significantly faster response time compared to the old system, indicating an improvement in system performance.

4. Conclusions

The use of fine-tuned ChatGPT Turbo 3.5 successfully provides faster, more relevant, and contextually appropriate responses, thereby enhancing the overall user experience. Based on the statistical analysis, the obtained p-value was 0.0001, which is lower than 0.05. This result confirms that the new system, which integrates the AI-powered chatbot, significantly reduces transaction time compared to the old system, indicating a substantial improvement in overall system performance.

The best-performing chatbot model achieved a Training Loss of 0.3894 and a Validation Loss of 0.5787, indicating effective learning without signs of overfitting. The Training Mean Token Accuracy of 0.8799 and Validation Mean Token Accuracy of 0.7971 demonstrate high accuracy in token prediction. Furthermore, the Full Validation Loss of 0.6242 and Full Validation Mean Token Accuracy of 0.8122 confirm the model's efficiency in reducing errors and improving prediction accuracy across the entire validation dataset.

5. Acknowledgment

The authors would like to express their gratitude to Alni Accessories Store for providing the data used in the preparation of this article.

6. Authors Note

The authors declare that there are no conflicts of interest related to the publication of this article. The manuscript has undergone thorough verification and is confirmed to be free from any form of plagiarism.

7. References

Hairi, M. H., Purnawansyah, P., & Sugiarti, S. (2020). Pencarian Produk Mall Online Menggunakan Metode Multiple Keywords searching. *Buletin Sistem Informasi Dan Teknologi Islam*, *1*(1), 29–35. https://doi.org/10.33096/busiti.v1i1.659

Herfian, M. R., & Adriansyah, A. R. (2021). Analisis dan Perancangan Aplikasi Chatbot dalam Pelayanan Penerimaan Mahasiswa Baru pada Perguruan Tinggi. *Jurnal Informatika Terpadu*, *7*(2), 87–93. https://doi.org/10.54914/jit.v7i2.370

Kappi, C. M. K., & Marlina, L. (2023). The Effect of Chatbot Services on Online Shop Customer Satisfaction. *Brilliance: Research of Artificial Intelligence*, *3*(2), 252–261. https://doi.org/10.47709/brilliance.v3i2.3133

Kishore, Y. (2024). Optimizing Enterprise Conversational AI: Accelerating Response Accuracy with Custom Dataset Fine-Tuning. *Intelligent Information Management*, *16*(02), 65–76. https://doi.org/10.4236/iim.2024.162005

Kole, V., & Sugeng, S. (2021). Implementasi Penjualan Makanan Secara Online dengan Metode DevOps pada Restaurant Zenbu House Jakarta Barat. *Jurnal Sosial Teknologi*, *1*(8), 867–874. https://doi.org/10.59188/jurnalsostech.v1i8.174

Misnawati. (2023). ChatGPT: Keuntungan, Risiko, Dan Penggunaan Bijak Dalam Era Kecerdasan Buatan. *PROSIDING SEMINAR NASIONAL PENDIDIKAN BAHASA SASTRA SENI DAN BUDAYA*, 54–67.

OpenAI. (2024). *Fine-tuning. OpenAI*. https://platform.openai.com/docs/guides/fine-tuning?utm_source=chatgpt.com

Septiadi, B, Isnandar, A. (2024). Transformasi Bisnis di Era Digital: Analisis Sistematis Terhadap E-Bisnis di Indonesia pada Konteks UMKM. *Journal of Digital Literacy and Volunteering*, *2*, 38–43. http://jurnal.relawantik.or.id/index.php/ict

Wibowo, S. M., Susanti, E., & Fatkhiyah, E. (2023). Perancangan Aplikasi Mobile Sistem Informasi Akademik Mahasiswa Sebagai Salah Satu Tahapan Metode DevOps. *INSOLOGI: Jurnal Sains Dan Teknologi*, *2*(6), 1191–1202. https://doi.org/10.55123/insologi.v2i6.2876

Xu, N., & Hu, C. (2023). *Enhancing E-Commerce Recommendation using Pre-Trained Language Model and Fine-Tuning*. http://arxiv.org/abs/2302.04443