Bournemouth University

FACULTY OF SCIENCE & TECHNOLOGY

MSc Data Science and Artificial Intelligence
August 2024

Generative AI in the Enterprise as a co-worker for tech support

by

Yasemin Karaca

Faculty of Science & Technology
Department of Computing and Informatics
Individual Masters Project

Abstract

The application of Generative AI (GenAI) into enterprise tech support systems represents a significant change towards automating and enhancing technical support processes. This dissertation examines the potential of GenAI to act as a co-worker in tech support roles by integrating the OODA (Observe, Orient, Decide and Act) loop framework. It explores how this technology can augment support engineers, automate routine tasks, and assist in complex problem-solving scenarios, aiming to improve and optimise efficiency and customer satisfaction.

Through a systematic literature review and a methodologically robust research design, this study identifies specific tasks within tech support that can be automated or augmented by GenAI. The challenges and limitations of applying AI systems are researched, especially in performing tasks that demand emotional intelligence or involve complex decision-making. To address these challenges, the research addresses a hybrid AI-human approach, applying explainable AI (XAI) methods to ensure transparency and build trust among users.

Machine learning algorithms such as Logistic Regression, Random Forest, and Support Vector Machines are employed to classify support tickets by technical level, category, and emotional state. Findings show that while GenAI significantly improves efficiency for routine requests, human oversight is essential for managing complex interactions. This research highlights the need for a balanced AI-human approach to optimise tech support systems, suggesting strategies for future work to enhance AI integration and effectiveness.

Dissertation Declaration

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. Any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history, or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed: Yasemin Karaca .

Name: Yasemin Karaca

Date: 20/08/2024

Programme: MSc. Data Science and Artificial Intelligence

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

Signed: Yasemin Karaca .

Name: Yasemin Karaca

Date: 20/08/2024

Acknowledgements

I would like to express my heartfelt appreciation to my family, whose constant support and encouragement have been the key to my achievements. Their belief in me and constant push to follow my dreams have been crucial in bringing me to this point.

A special thanks to my supervisor, Huseyin Dogan, for his invaluable guidance and assistance throughout this term. His insights and support were crucial in the completion of my thesis, and I am deeply appreciative of his mentorship.

I also extend my thanks to Stephen Giff from Google, for providing us with the incredible opportunity to work alongside their team. This experience has been highly enjoyable and enriching, contributing significantly to my professional and personal growth.

Thank you all for your support, guidance, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Tech support in enterprises used to rely heavily on human expertise to troubleshoot issues, provide solutions, and maintain customer satisfaction. However, with the increasing complexity of technological systems and the rising volume of support requests, human-only support models face challenges such as longer response times, higher costs, and inconsistent service quality.

The study by Begicheva and Zhukovskaya (2022) states that customer satisfaction in tech support is greatly affected by elements such as the complexity of tasks, the speed at which problems are resolved, the wait time for responses, and the working conditions of the support staff. These challenges drive the need for integrating advanced technologies like Generative AI (GenAI) to augment and optimise support processes.

1.1 Problem Definition

In the rapidly changing and evolving technology environment, enterprises are increasingly using GenAI to improve different aspects of their operations, including customer tech support activities. Despite the potential benefits, such as improved efficiency, cost savings, and enhanced customer experience, implementing GenAI in enterprise tech support presents several challenges. These include a lack of comprehensive understanding of the specific needs and use cases of both customers and support agents, uncertainties in applying decision-making frameworks like the OODA (Observe, Orient, Decide, Act) Loop, and identifying the most relevant tasks for AI training. Additionally, there is a need to balance automation with human interaction and develop robust methodologies to evaluate AI effectiveness. According to Brynjolfsson et al. (2023), GenAI tools improve productivity among customer support agents significantly by increasing the number of issues resolved per hour by 14% on average. This productivity boost shows the potential of GenAI to transform tech support functions, making them more efficient and responsive.

1.2 Aims and Objectives

The project's primary goal is to explore how the OODA Loop framework, combined with GenAI technologies, can be utilised to augment, assist, or automate Support Engineer tasks. This study seeks to address the challenges associated with the implementation of GenAI in tech support functions and to provide actionable insights and recommendations for enterprises.
It will assess the relevance of the OODA loop framework in describing the workflows of support engineers in enterprise settings and explore the seamless integration of GenAI models into enterprise workflows to either automate or enhance tasks undertaken by these professionals. Objectives include:

1. Reviewing the state of the art in literature involves conducting a thorough analysis of current literature to identify advancements, trends, and gaps in the application of GenAI in tech support, all aimed at highlighting areas for further research and innovation.
2. Systematically map the stages of the OODA loop framework to specific activities and tasks performed by support engineers, identifying tasks best suited for training GenAI models within support workflows.

3. Develop models or scenarios presenting full automation or augmentation through AI in support tasks through a concept display.
4. Evaluate potential improvements in efficiency, scalability, and user experience that could come from using GenAI models.
5. Provide feasible options for integrating GenAI models into support processes for businesses.

1.3 Research Questions

1) In what ways can we leverage the OODA Loop framework to augment, assist, or automate the goals and activities of Support Engineers?

    a) How well do the stages of OODA Loop Framework <Observe, Orient, Decide, Act>, map to the activities and tasks of a Support Engineer?

2) What are the types of tasks and activities that will be most relevant in training an AI model to assist, augment, or automate tasks/activities of Support Engineers?

    a) What tasks and activities would not be relevant/suitable?

3) Can we leverage the OODA Loop (or similar) framework to model Support Engineer scenarios, with a view to:

    a) Replacing the need for support engineers (Self-Serve)
    b) Augmenting Support Engineers (AI-Supported Support Engineers)

2 LITERATURE REVIEW

2.1 Introduction to OODA Loop Framework

The OODA (Observe, Orient, Decide, Act) loop, created by military strategist John R. Boyd in 1995, is a well-recognised decision-making framework that can be used to model organisational processes. Enck (2012) states that the concept of the OODA loop emphasises the importance of being agile and adapting quickly to changing situations. It highlights the significance of being able to make efficient decisions by quickly adjusting to evolving circumstances. According to Endsley (1995), effective decision-making relies on the ability to assess the current state of the environment, evaluate the available options, and anticipate the consequences of different actions. Situation awareness provides the foundation for this process. A diagram of the OODA loop is given in Figure 1.

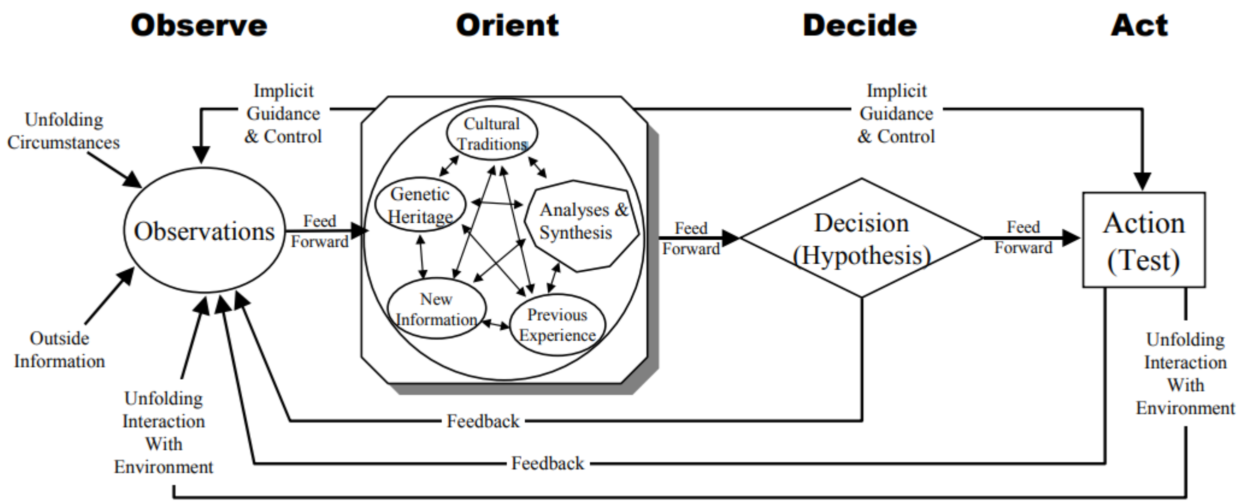


Figure 1. Boyd's OODA Loop Diagram (Brehmer 2005)

2.2 The Role and Responsibilities of Support Engineers

Being a support engineer is crucial for maintaining an enterprise's technology framework. Support engineers are responsible for technical assistance, customer support, system maintenance, documentation, and collaboration with development teams. They encounter challenges such as increasing system complexity, a high volume of support requests, rapid technological advancements, and rising user expectations. To tackle these issues, various solutions like automation, AI, remote support tools, knowledge management systems, predictive maintenance, collaboration platforms, and continuous training programs have been developed. These tools boost the efficiency of support engineers, allowing them to provide higher quality support and better manage the growing demands of their roles.

2.3 Overview of GenAI

A subset of artificial intelligence technology known as "GenAI" dedicates itself to content creation, this includes everything from text and graphics to code and music. This technology analyses large datasets and finds patterns or features that can be applied to produce new, creative outputs by utilising machine learning techniques, especially deep learning networks. One of the most significant advancements in GenAI has been the development of models like GPT (Generative Pre-trained Transformer) and DALL-E, which show remarkable capabilities in natural language processing (NLP) and image generation, respectively. These models utilise techniques such as transformers and neural networks, which have been pre-trained on extensive and diverse datasets. This pre-training allows them to produce content that is often hard to tell apart from that created by humans. Cevallos et al. (2023) state that GenAI can produce creative outputs such as images, music, text, and more by learning from vast datasets and generating content with similar characteristics to the original data.

2.4 The Role of Large Language Models (LLMs) in Tech Support

Enhancing the efficiency of technical support in customer service operations, improving accuracy and automating responses are provided by applying LLMs. In tech support, LLMs are employed primarily to automate responses to common queries, significantly reducing resolution times and enhancing the efficiency of support teams. They can parse through complex customer issues, extract relevant information, and provide or suggest solutions based on learned patterns. According to Debbah (2023), LLMs assist in diagnosing and resolving technical issues by accessing large datasets of troubleshooting steps and solutions. This capability allows for quicker resolution of customer problems and reduces the need for human intervention. However, Ma et al. (2023) indicated that there are challenges in implementing LLMs in tech support, including the need for large computational resources and the risk of generating incorrect or irrelevant responses.

2.5 GenAI in Enterprise Environment

GenAI is increasingly being recognised as an innovative technology in enterprises, offering a wide range of applications that can promote creativity, effectiveness and scalability. In enterprise environments, GenAI is used to improve processes across various domains such as customer service, marketing and human resources. One of the key advantages of GenAI in these settings is its ability to generate high-quality, customised content quickly and at scale.

Although GenAI has numerous business opportunities, it also carries the risk of being misused leading to the creation of deep fakes and other harmful uses. Investigating these potential misuse incidents may assist in creating safety measures (Houde et al. 2020).

2.5.1 AI-Powered in Enterprise Environment-Customer Support

Customer service operations can be improved by using AI-powered customer support. This application of AI can convert traditional customer support systems by automating interactions, providing instant responses to customer requests, and managing large volumes of requests without compromising service quality, which can enhance the quality and efficiency of the support departments.

Chatbots and Virtual Assistants: Interaction with customers using NLP, offering solutions on predefined scripts, and learning from ongoing interactions to improve responses. According to Sousa et al. (2019), chatbots and virtual assistants enhance customer service by automating responses and providing immediate assistance, thereby improving enterprise efficiency.
Automated Ticketing Systems: By automatically categorising, prioritising, and assigning support tickets to the appropriate human agents or, if feasible, resolving them directly. Prabhakar (2018) states that AI-driven ticketing systems improve customer experience by providing faster and more reliable responses, reducing the workload on human agents, and improving overall service quality.
Predictive Customer Service: AI models can predict customer issues before they become apparent by analysing patterns in customer behaviour and usage data. Prakash et al. (2023) state that these models help businesses gain insight into customer preferences and purchasing patterns, which are critical to enhancing customer satisfaction and loyalty.

The core benefits of a Self-Serve GenAI-powered support agent are:

24/7 Availability: When a user needs help, Large Language Models (LLM) are available around the clock to provide it. As a result, wait times are shorter and the general customer experience is enhanced. According to Al-Mekhlal et al. (2023), AI in customer service allows for continuous, 24/7 support, increasing customer satisfaction and engagement.
Scalability: LLMs can manage numerous requests concurrently, which is essential during peak customer assistance needs or unplanned spikes.
Multilingual Support: Skilled LLMs can understand and produce content in multiple languages, enhancing communication and assisting international customers.
Cost-Efficiency: LLMs can automate many repetitive tasks, freeing up human agents for more complex problems and potentially reducing overall operating expenses, despite the initial setup cost.
Improved User Empowerment: Users gain confidence in their ability to resolve problems independently, advancing a sense of control and competence.

According to De Andrade & Tumelero (2022), AI chatbots greatly increase customer service efficiency by addressing basic requests, easing the load for human agents, and setting them up to handle more complicated problems.

2.5.2 Existing Implemented AI Tool to Enhance Support Engineering Functions

Case Study: IBM Watson in Customer Support

According to the article by Omar (2024), IBM's Watson is a notable example of AI integration into support systems, particularly in customer service. Deployed in various customer support scenarios, Watson assists in handling requests and troubleshooting, significantly enhancing operational efficiency. For instance, when integrated into a major telecom company’s support operation, Watson led to a considerable reduction in response times and improved customer satisfaction scores. This AI-driven solution utilises NLP to learn from interactions, providing accurate and personalised assistance across multiple channels. In Figure 2, the system architecture of the IBM Watson study can be seen.

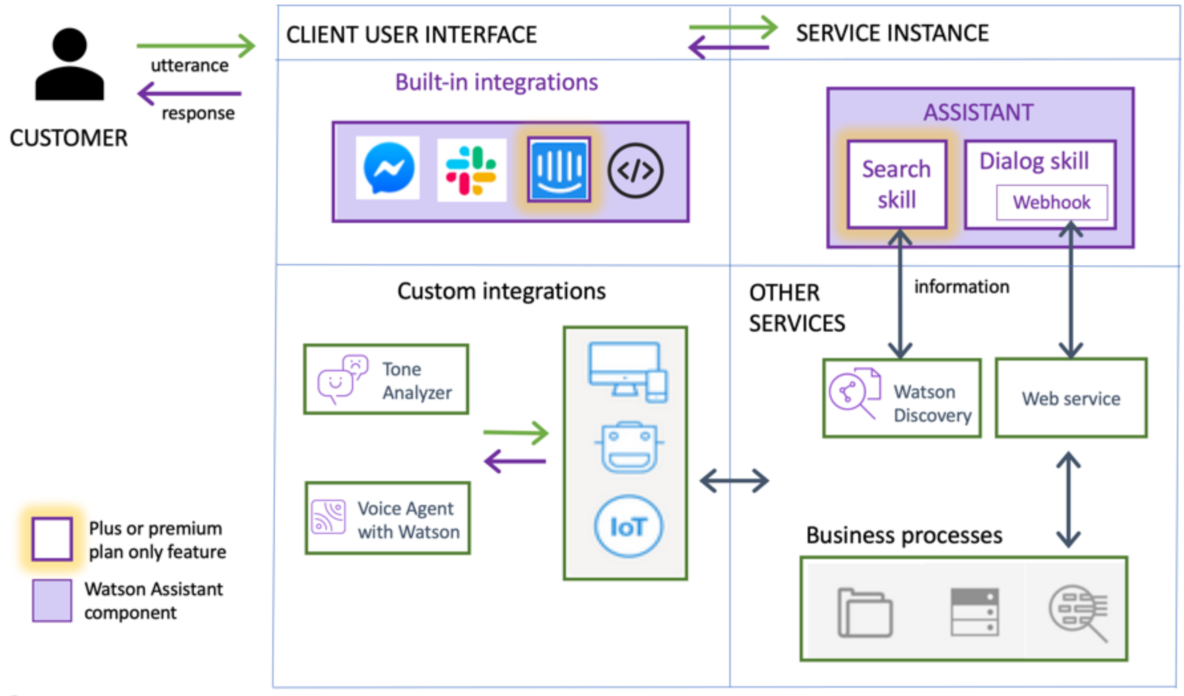


Figure 2. IBM Watson Study (Omar 2024)

By implementing this study, Watson reached the success of having high efficiency by reducing the response times, scalability by managing a high volume of customer interactions and enhancing customer satisfaction by providing quick and accurate responses. However, while Watson performed well with routine queries, they struggled with more complex issues requiring human expertise.

2.6 Explainable AI in Tech Support

Building trust and understanding in AI systems, especially in tech support, requires explainability in AI. Explainable AI (XAI) improves the overall support experience by enabling users and agents to understand, trust, and engage with AI-driven solutions in productive ways. According to Kim et al. (2023), XAI develops trust and enhances collaboration between users and AI systems, making the support process smoother and more effective. In Figure 3, the concept of XAI is illustrated.

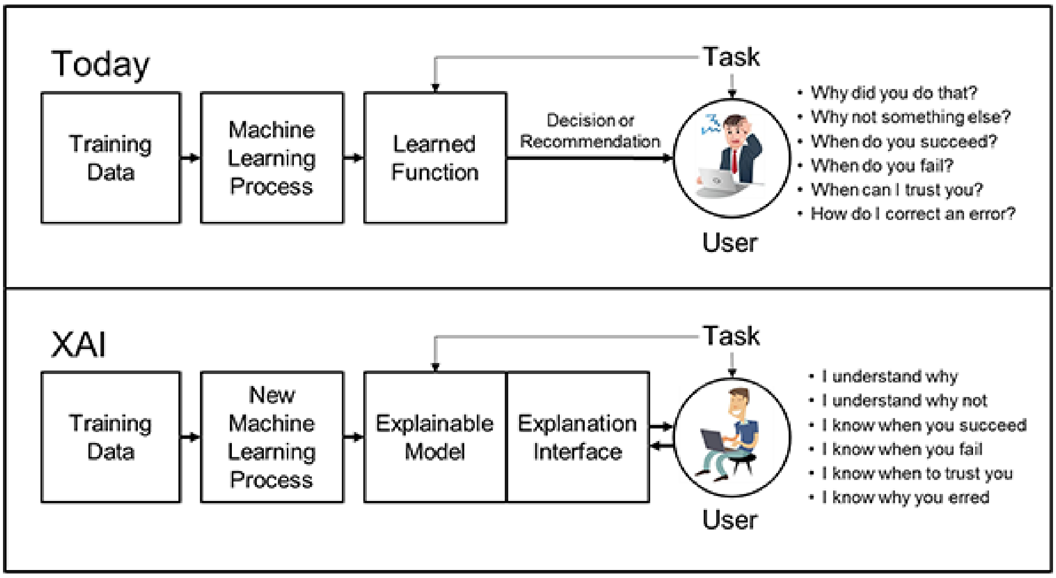


Figure 3. XAI Concept (Defense Advanced Research Projects Agency 2024)

Best practices of XAI in tech support should have these concepts:

Clear documentation and User guides: Detailed and user-friendly manuals can be provided to explain the AI's workings, capabilities, and limitations.
XAI Techniques: Models like decision trees can be used to enhance the understanding. Tools like LIME or SHAP can help to explain the complex models.
Transparency in Data and Algorithms: The collection methods, processing and source of the data can be outlined clearly. The processes of training and validation can be described.
Human–AI Collaboration: Having human intervention and oversight will enhance trust in AI. According to Lai et al. (2023), to improve the effectiveness of the explanations, this may involve presenting model reasoning selectively in a way that corresponds with what users feel is relevant.
Continuous Monitoring and Feedback Loops: Implementing ongoing monitoring of AI performance and having feedback mechanisms to ensure the AI remains accurate and reliable.

2.7 Hybrid AI-Human Tech Support Model

According to the study by Zheng et al. (2017) hybrid-augmented intelligence merges human intellect with artificial intelligence to address complex problems and make better decisions. This approach combines the strengths of both humans and computers for improved outcomes. In Figure 5, an integrated workflow for a machine learning system is enhanced by a knowledge base. New inputs are processed into structured data, which a machine learning algorithm uses to generate decisions. These decisions are evaluated for confidence: high-confidence results are outputted directly, while low-confidence results are reviewed by human experts, whose feedback helps tune the model. At the same time, the system updates its knowledge base, by composing new knowledge from each decision. This continuous feedback loop between machine learning, human expertise, and a dynamically evolving knowledge base ensures improved decision-making accuracy and reliability over time. Figure 4 illustrates the framework of the hybrid AI-human tech support model.

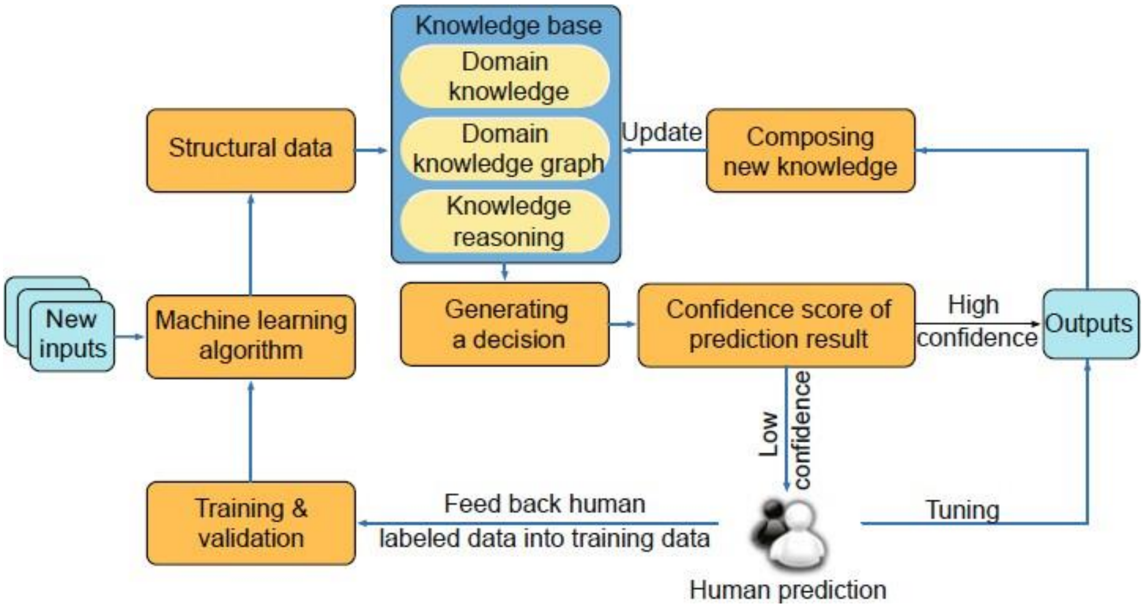


Figure 4. Basic framework of human-in-the-loop hybrid-augmented intelligence (Zheng et al. 2017)

A study by Jiang et al. (2022) claims that situational awareness is key to improving interactions between humans and AI. In the study, it is indicated that adopting a situational awareness approach can mitigate the negative effects of AI on user experiences, enhance human control, and facilitate effective decision-making in real-world contexts.

3 METHODOLOGY

3.1 Synthetic Dataset

For the purposes of this dissertation, two synthetic datasets were generated. The first dataset, ticket_database.csv includes a comprehensive collection of ticket_id, customer_id, technical level, FaQ, category, answer, emotional state and status. The second dataset, ticket_base_proposed.csv, includes ticket_id, support engineer, proposed solution, error code and contact. These data were created to simulate FAQs and corresponding answers, while also labelling them with appropriate technical levels, categories, and emotional states. In Figures 5, 6 and 7 details about the dataset can be seen.

Ticket_ID	Customer_ID	Technical Level	FaQ	Category	Answer	Emotional State	Status
1000	C1	Beginner	How can I create an account?	Account	To create an account, click on th	Satisfied	Closed
1001	C2	Intermediate	How can I recover deleted files?	Software	Use Microsoft OneDrive File Rec	Satisfied	Closed
1002	C3	Beginner	How can I reset my password?	Account	To reset your password, click on	Satisfied	Closed
1003	C4	Intermediate	Why can't I connect to the Wi-Fi?	Network	Check your router settings and e	Frustrated	Open
1004	C5	Advanced	How do I backup my data?	Software	Set up automated, incremental I	Confused	Closed
1005	C6	Intermediate	What antivirus should I use?	Software	Norton 360 Provides comprehen	Satisfied	Closed
1006	C7	Beginner	How to set up my email?	Account	Open Your Email Client, follow th	Confused	Closed
1007	C8	Beginner	What is the wifi password?	Network	The WiFi password is bournemo	Satisfied	Closed
1008	C9	Intermediate	How to configure VPN?	Network	Open Settings, Go to Network &	Confused	Closed
1009	C10	Advanced	Can I upgrade my computer hardware?	Hardware	Yes, by upgrading RAM can impr	Confused	Open
1010	C11	Advanced	Why is the network so slow?	Network	Simply restarting your router, mo	Frustrated	Open
1011	C12	Intermediate	How do I update my software?	Software	Download the latest version fron	Satisfied	Closed
1012	C13	Advanced	Why is my screen flickering?	Hardware	Diagnose hardware issues using	Confused	Open
1013	C14	Advanced	How do I fix a Blue Screen of Death (BSOD)?	Hardware	Analyze crash dump files to iden	Frustrated	Closed
1014	C15	Beginner	How can I improve my internet speed?	Network	Restart your router and modem.	Frustrated	Closed
1015	C16	Intermediate	How do I configure my email on my phone?	Software	Open Settings, scroll to mail, tap	Confused	Closed

Figure 5. ticket_database.csv dataset

Ticket_ID	Support En	Proposed Solution	Error Code	Contact
1035	Yasemin K	If your account does	A35	yaseminkaraca@example.com
1036	Gözde Sars	To verify the RDP po	N36	gozdesarsar@example.com
1037	John Doe	If you can't use the t	A37	johndoe@example.com
1039	Emily Moor	Ensure that your Wi-	S39	emilymoore@example.com
1040	Thomas Jo	IP is a set of commu	N40	thomasjones@example.com
1041	Gözde Sars	In a working comput	N41	gozdesarsar@example.com
1043	John Doe	Check "All Mail", "T	A43	johndoe@example.com
1046	Gözde Sars	Close Windows, Op	A46	gozdesarsar@example.com
1047	John Doe	Check if the applica	N47	johndoe@example.com
1030	Yasemin K	If the program was r	H30	yaseminkaraca@example.com
1031	Emily Moor	Close unnecessary	H31	emilymoore@example.com

Figure 6. ticket_base_proposed.csv dataset

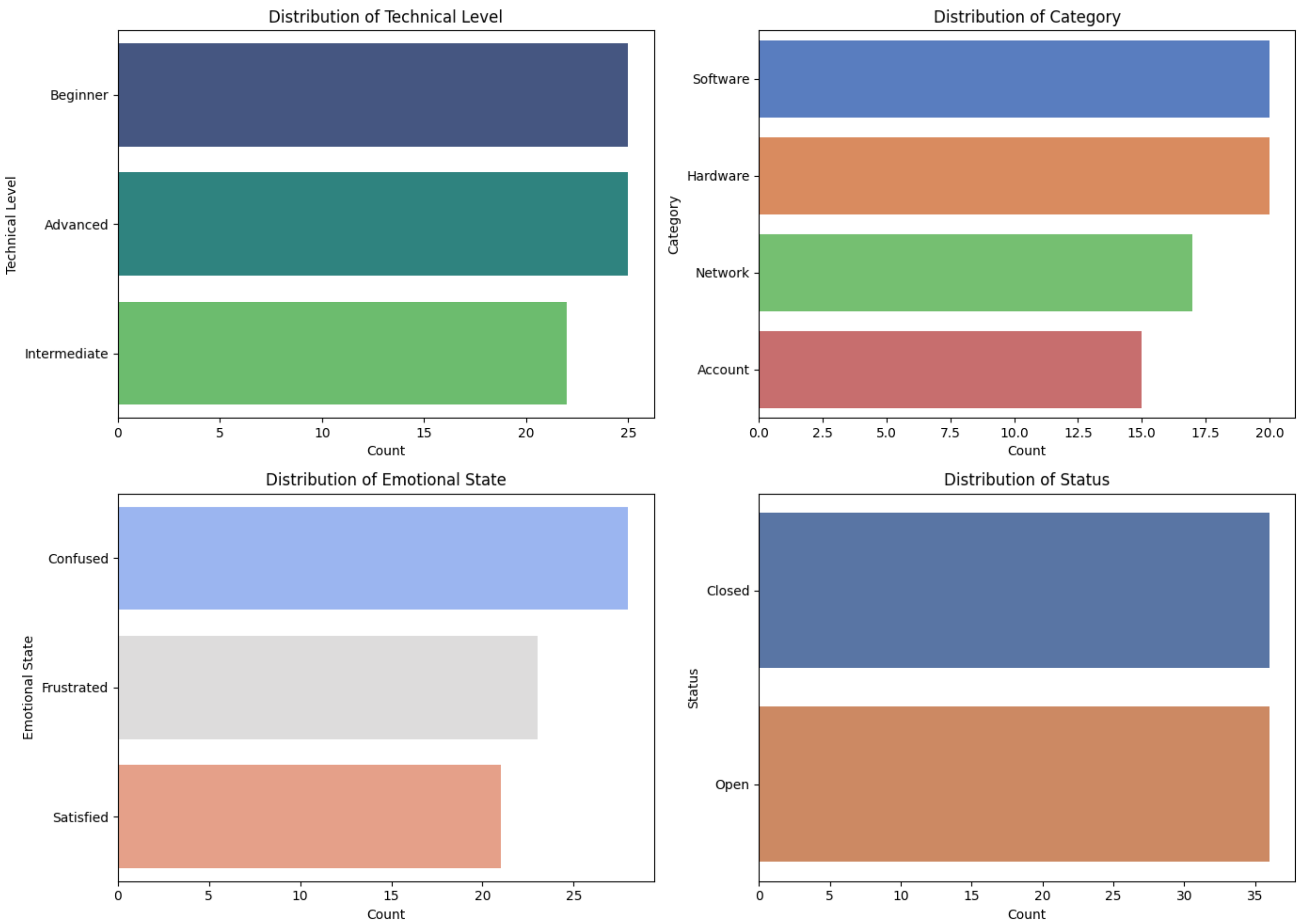


Figure 7. Distribution of dataset details

3.2 Model Design, Training and Evaluation

The dataset was split into training and testing sets using an 80/20 split for each classification task: technical level, category, and emotional state. Three machine-learning models were evaluated for this system. To predict technical level, logistic regression was used due to its simplicity for binary classification. A random forest classifier was applied to predict categories due to its ability to handle complex interactions between features. And support vector machine (SVM) was used to predict the emotional state due to its effectiveness in high-dimensional spaces and versatility with different kernel functions. Grid Search Cross-Validation was utilised to find the optimal hyperparameters for each model using predefined parameter grids. In Figure 8 and Table 1, hypermeter tuning, and its search range are shown.

```
[7]  # Define parameter grids for hyperparameter tuning
     param_grid_logistic = {'clf__C': [0.01, 0.1, 1, 10, 100], 'clf__solver': ['liblinear', 'lbfgs'], 'clf__max_iter': [100, 500, 1000]}
     param_grid_random_forest = {'clf__n_estimators': [100, 200, 300], 'clf__max_depth': [None, 10, 20, 30]}
     param_grid_svc = {'clf__C': [0.1, 1, 10], 'clf__kernel': ['linear', 'rbf']}
```

Figure 8. Hyperparameter tuning

Table 1. Model Hyperparameters and Search Range

Model	Hyperparameters	Search Range
Logistic Regression	['C', 'solver', 'max_iter']	{'C': [0.01, 0.1, 1, 10, 100], 'solver': ['liblinear', 'lbfgs'], 'max_iter': [100, 500, 1000]}
Random Forest	['n_estimators', 'max_depth']	{'n_estimators': [100, 200, 300], 'max_depth': [None, 10, 20, 30]}
SVC	['C', 'kernel']	{'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}

3.3 Recommendation System

A recommendation system was developed to provide solutions based on the user problem requests. Sentence-BERT library was integrated to generate embeddings for FAQs in the ticket database. SBERT (Sentence-BERT) can generate high-quality embeddings efficiently, making it suitable for real-time recommendation systems. FAISS (Facebook AI Similarity Search) was used to build an index for the embeddings to facilitate fast nearest-neighbour searches. The semantic search mechanism retrieves the most similar FAQ from the historical ticket data to recommend a solution. If the recommended solution is not satisfactory, the system provides an advanced solution based on error codes or connects the user to a support engineer. It also writes this information into a new dataset, fetching the customer ID and marking the ticket status as pending.

3.3.1 Finding the most relevant FAQ

When a user submits a request, it is encoded into an embedding using the same SBERT model. This ensures that both the FAQs and the request are represented in the same semantic space. The request embedding is then passed to the FAISS index, which has been pre-populated with the FAQ embeddings. FAISS performs a nearest-neighbour search to find the most similar FAQ embedding to the request embedding. The index returns the identifier of the closest FAQ, which corresponds to the most relevant FAQ in the dataset. The system retrieves the FAQ and its solution based on the identifiers returned by FAISS.

3.4 Implementing XAI

To ensure transparency and interpretability of the model predictions, XAI techniques are used. The LimeTextExplainer from the LIME (Local Interpretable Model-agnostic Explanations) library is used to provide insights into why certain predictions were made by the models. This helps users understand the model's decision-making process, increasing trust and adoption. In Figure 9, the implementation of XAI is given.

```
def explain_prediction(model, problem_statement):
    explainer = LimeTextExplainer(class_names=model.named_steps['clf'].classes_)
    exp = explainer.explain_instance(problem_statement, model.predict_proba, num_features=10)
    return exp
```

```
tech_exp = explain_prediction(tech_level_model, problem_statement)
cat_exp = explain_prediction(category_model, problem_statement)
emo_exp = explain_prediction(emotional_model, problem_statement)

print("\nExplanation of Technical Level Prediction:")
print(tech_exp.as_list())

print("\nExplanation of Category Prediction:")
print(cat_exp.as_list())

print("\nExplanation of Emotional State Prediction:")
print(emo_exp.as_list())
```

Figure 9. Implementation of XAI

3.5 System Architecture and Scenario

When a user submits a request via the command-line interface, the AI-based IT Ticketing Support System first classifies the problem by predicting its technical level, category, and the user's emotional state using pre-trained models. Based on these classifications, the system recommends an initial solution retrieved from a semantic search of historical ticket data using Sentence-BERT and FAISS. If the user indicates that the initial solution is ineffective, they are prompted to provide an error code, which is used to fetch an advanced solution from a database of proposed solutions. If further assistance is required, the system assigns a support engineer and provides the engineer's contact information to the user, ensuring that unresolved tickets are managed and followed up efficiently. This integrated approach combines machine learning and semantic search to optimise the resolution of IT support tickets. In Figure 10, the workflow integration of this scenario is given.

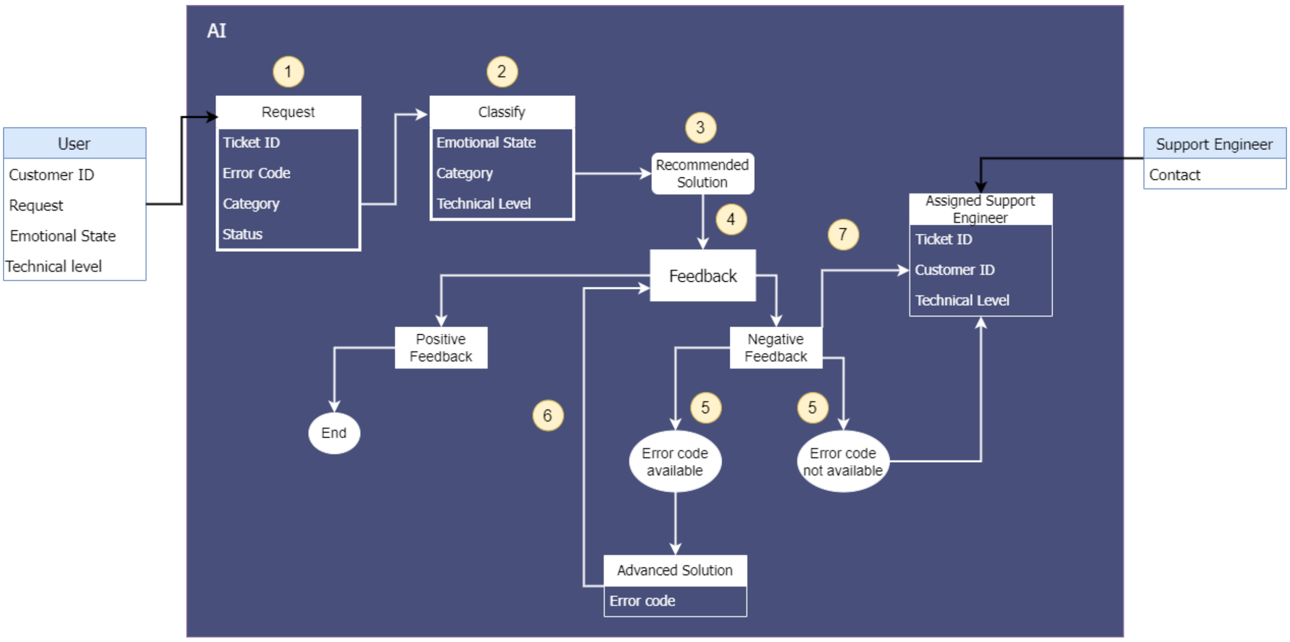


Figure 10. Workflow Integration

3.6 Project Management Tools and Techniques

During this project, regular meetings with the client Google were held every Monday. In these meetings, I received critical updates and feedback that significantly influenced the project's direction and success. One key area of discussion was the integration of the OODA Loop framework into the problem-solving approach. This integration aimed to enhance the system's agility and responsiveness, ensuring that it could quickly adapt to new information and evolving user needs. Additionally, Google provided valuable insights into optimising the dataset attributes. They suggested refining the data schema to include more detailed information about technical issues, emotional states and user interactions, which improved the accuracy of the model. Guidance on research topics was another significant area of support from Google, particularly regarding XAI. They emphasised the importance of transparency in AI-driven decisions and recommended integrating XAI techniques to make the model's predictions and recommendations more understandable to users and support engineers.

4 ARTEFACT – (REFERRED BY APPENDIX B)

The artefact developed is an AI recommendation system that processes an input request from the user, predicts the user's technical level, emotional state and request's category. From the historical data, the model is trained and evaluated to recommend a solution depending on the request. More details about the implementation can be found in Appendix B.

4.1 Data Preprocessing

The first step is to check for the missing values in the ticket_database.csv by replacing them with empty strings to ensure that the model training process is not affected by null values.

4.1.1 Encoding Categorical Variables

Categorical variables that were predicted like "Technical Level", "Emotional State" and "Category" were encoded using a label encoder to convert them into the numerical format, which is required for machine learning models. Figure 11 shows the encoding of categorical variables.

```
# Encoding categorical variables
le_technical = LabelEncoder()
le_category = LabelEncoder()
le_emotional = LabelEncoder()

ticket_data['Technical Level Encoded'] = le_technical.fit_transform(ticket_data['Technical Level'])
ticket_data['Category Encoded'] = le_category.fit_transform(ticket_data['Category'])
ticket_data['Emotional State Encoded'] = le_emotional.fit_transform(ticket_data['Emotional State'])
```

Figure 11. Encoding Categorical Variables

4.1.2 Data Splitting

The dataset was split into training and testing sets with an 80/20 ratio. This split was performed separately for each classification task to ensure that the models could be evaluated on unseen data. Figure 12 shows how the data was split.

```
[7] # Splitting the data for classification tasks
X_train, X_test, y_train_tech, y_test_tech = train_test_split(ticket_data['FaQ'], ticket_data['Technical Level Encoded'], test_size=0.2, random_state=42)
_, _, y_train_cat, y_test_cat = train_test_split(ticket_data['FaQ'], ticket_data['Category Encoded'], test_size=0.2, random_state=42)
_, _, y_train_emo, y_test_emo = train_test_split(ticket_data['FaQ'], ticket_data['Emotional State Encoded'], test_size=0.2, random_state=42)
```

Figure 12. Splitting the Data

4.2 Model Training and Evaluation

For each classification task, a diverse set of models including Logistic Regression, Random Forest, and SVC were used to predict different aspects such as technical level, category, and emotional state of the tickets. Hyperparameter tuning was performed using GridSearchCV to optimise the models' performance. The training process was integrated within a pipeline that included a TF-IDF vectorizer followed by the classifier, ensuring an efficient workflow from text vectorization to model training. Evaluation metrics such as accuracy, MCC, which measures the quality of binary classifications, and log-loss, which quantifies prediction uncertainty, were computed to assess model performance. The confusion matrix was also visualised to identify misclassifications. Figure 13 illustrates the implementation of model training and evaluation.

```
# Define parameter grids for hyperparameter tuning
param_grid_logistic = {'clf__C': [0.01, 0.1, 1, 10, 100], 'clf__solver': ['liblinear', 'lbfgs'], 'clf__max_iter': [100, 500, 1000]}
param_grid_random_forest = {'clf__n_estimators': [100, 200, 300], 'clf__max_depth': [None, 10, 20, 30]}
param_grid_svc = {'clf__C': [0.1, 1, 10], 'clf__kernel': ['linear', 'rbf']}
```

```
[39] def train_evaluate_model_with_pipeline(X_train, y_train, X_test, y_test, target_names, model, param_grid, plot_confusion=False):
    pipeline = Pipeline([
        ('tfidf', TfidfVectorizer(max_features=5000, ngram_range=(1, 2))),
        ('clf', model)
    ])

    grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
    grid_search.fit(X_train, y_train)
    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    y_pred_proba = best_model.predict_proba(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=target_names)
    mcc = matthews_corrcoef(y_test, y_pred)
    logloss = log_loss(y_test, y_pred_proba)

    if plot_confusion:
        cm = confusion_matrix(y_test, y_pred)
        plt.figure(figsize=(10, 7))
        sns.heatmap(cm, annot=True, fmt="d", xticklabels=target_names, yticklabels=target_names)
        plt.xlabel('Predicted')
        plt.ylabel('True')
        plt.title(f'Confusion Matrix for {model.__class__.__name__}')
        plt.show()

    return best_model, accuracy, report, mcc, logloss
```

Figure 13. Implementation of Model Training and Evaluation

4.3 AI-Based Recommendation System

The recommendation system uses trained models to classify requests, searches for solutions using semantic search, and explains predictions using LIME. The following Figures 14, and 15 shows the model setup, how the system integrates classification, explanation of the predictions, and finding the solution.

```
# Semantic search setup using Sentence-BERT and FAISS
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

# Generate embeddings for all FAQs in the ticket database
ticket_embeddings = model.encode(ticket_data['FaQ'].tolist())

# Build the FAISS index
dimension = ticket_embeddings.shape[1]
index = faiss.IndexFlatL2(dimension)
index.add(ticket_embeddings)
```

Figure 14. Semantic Search Set Up

```
[12] def classify_problem(problem_statement):
         problem_tfidf = tech_level_model.named_steps['tfidf'].transform([problem_statement])
         tech_pred = tech_level_model.named_steps['clf'].predict(problem_tfidf)[0]
         cat_pred = category_model.named_steps['clf'].predict(problem_tfidf)[0]
         emo_pred = emotional_model.named_steps['clf'].predict(problem_tfidf)[0]

         return {
             'Technical Level': le_technical.inverse_transform([tech_pred])[0],
             'Category': le_category.inverse_transform([cat_pred])[0],
             'Emotional State': le_emotional.inverse_transform([emo_pred])[0]
         }
```

```
[13] def explain_prediction(model, problem_statement):
         explainer = LimeTextExplainer(class_names=model.named_steps['clf'].classes_)
         exp = explainer.explain_instance(problem_statement, model.predict_proba, num_features=10)
         return exp
```

```
[14] def find_solution(problem_statement):
         problem_embedding = model.encode([problem_statement])
         distances, indices = index.search(problem_embedding, 1)
         closest_match_idx = indices[0][0]

         return ticket_data.iloc[closest_match_idx]['Answer'], ticket_data.iloc[closest_match_idx]['Ticket_ID']

     def recommend_solution(problem_statement):
         classification = classify_problem(problem_statement)
         solution, ticket_id = find_solution(problem_statement)

         return {
             'Classification': classification,
             'Recommended Solution': solution,
             'Ticket_ID': ticket_id
         }
```

Figure 15. Implementation of Problem Resolution Functions

5 PROBLEM SPACE CONTEXTUALISATION AND ANALYSIS

5.1 Relevant and Irrelevant Tasks and Activities for GenAI Training

It is crucial to identify the boundaries of tasks and activities that are relevant for GenAI. Understanding these differences between relevant and irrelevant tasks helps in optimising the integration of AI systems within various workflows, ensuring both efficiency and effectiveness. By systematically categorising tasks that GenAI can perform effectively versus those that require human intelligence, we can better couple the strengths of both AI and human capabilities. Table 2 provides a detailed comparison of tasks and activities, focusing on where GenAI performs effectively and where human intervention remains essential.

Table 2. Relevant and Irrelevant Tasks and Activities for GenAI

	Relevant Tasks	Irrelevant Tasks	
	Routine issue identification and categorisation	**Tasks requiring high emotional intelligence**	
Task	Can automatically identify and categorise common technical issues based on the input.	Handling emotionally charged interactions with frustrated or confused users.	*Example*
Activity	Uses NLP to parse user statements and classifies.	AI lacks the subtle understanding and empathy required to navigate complex emotional situations effectively.	*Reason*
	Data Collection and Analysis	**Complex problem-solving needing human intuition**	
Task	Gathers and analyses data from various sources to diagnose the problem	Diagnosing and resolving highly complex technical issues that demand deep domain expertise and innovative problem-solving skills.	*Example*
Activity	Implements protocols to collect user logs, error reports, and performance metrics. Machine learning analyses this data for faster, more accurate problem diagnosis.	Such tasks often require unconventional solutions and insight that AI cannot match. Human engineers excel due to their critical thinking and ability to adapt to unexpected challenges.	*Reason*
	Solution recommendation and automation	**Client-Specific Customisation**	
Task	Providing automated suggestions for resolving technical issues.	Developing customised solutions designed to meet the unique needs and configurations of individual clients.	*Example*
Activity	Training AI models on historical ticket data to learn common solutions for specific problems.	These tasks require a detailed understanding of the client's specific environment, preferences, and requirements, which can vary significantly.	*Reason*

5.2 XAI in Tech Support: Example Scenario sequence diagram based on Literature

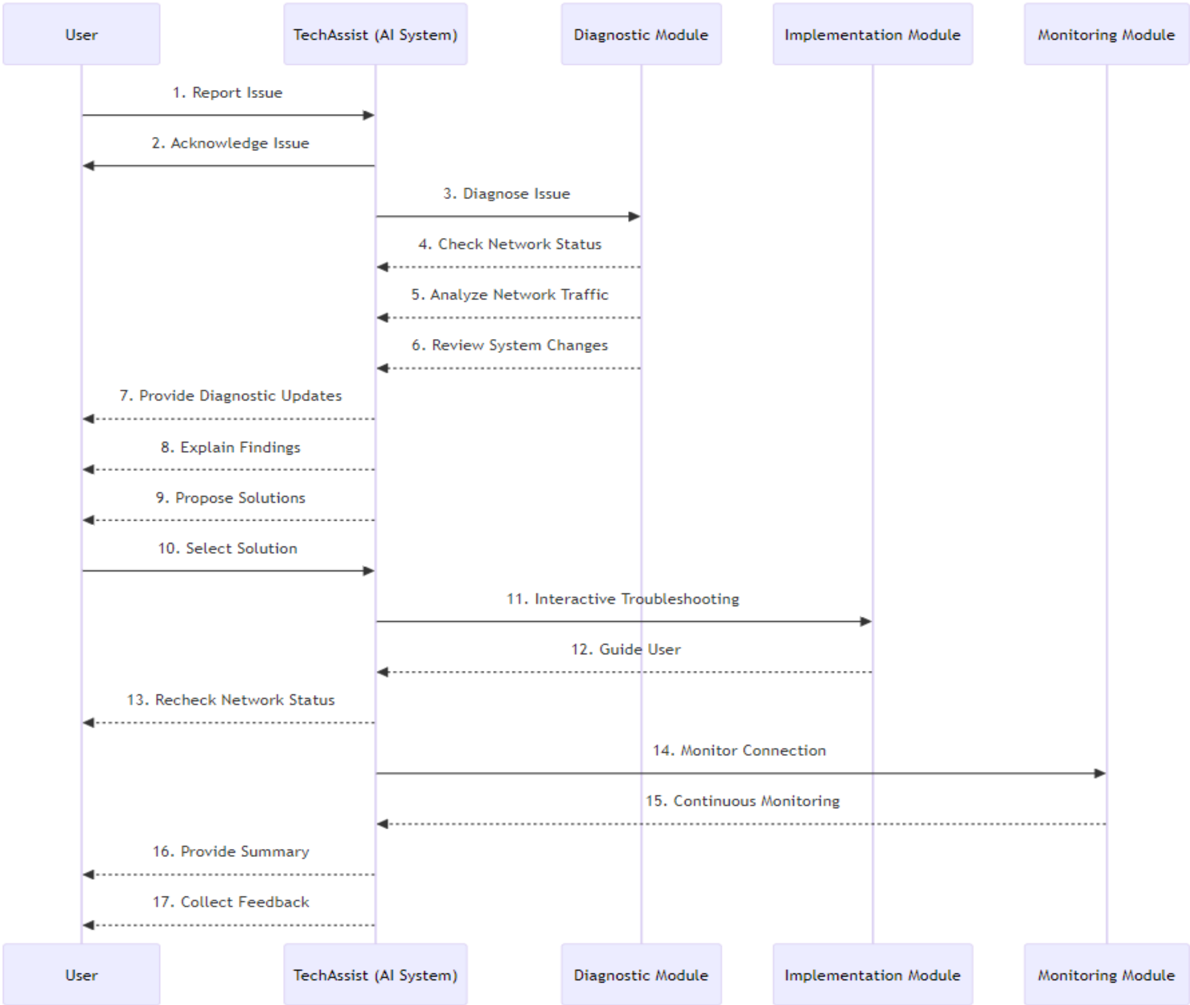


Figure 16. Scenario: Sequence Diagram of a Network Issue and XAI

The sequence diagram in Figure 16 shows the user's interactions with TechAssist, a powered by AI tech support system, as it diagnoses and fixes problems with network connectivity. XAI, which offers clarity and transparency at every stage, is essential to this process. TechAssist makes sure the user is informed and involved at every step, from identifying the problem and carrying out diagnostics to outlining conclusions and suggesting fixes. XAI improves user satisfaction and trust by providing real-time updates, understandable explanations, and reasons for suggested actions. This makes the AI system's functions easier for humans to understand and more transparent.

5.3 Risk Assessment of Integrating AI Agent into Enterprises

As enterprises increasingly integrate AI technologies, it is crucial to assess the associated risks and develop strong mitigation strategies. The integration of AI agents into enterprise environments presents challenges that must be carefully managed to ensure successful deployment and operation. By identifying these risks and implementing effective mitigation strategies, enterprises can better prepare for the complexities of AI integration, enhancing operational efficiency and maintaining high standards of data security and customer service. Table 3 provides a comprehensive risk assessment framework, detailing the descriptions, impacts, and mitigation strategies for various risks associated with the integration of AI agent into enterprises.

Table 3. Risk Assessment

Risk	Description	Impact	Mitigation
Data Privacy	According to Arthur et al. (2023), the vast amount of personal and sensitive data handled by genAI systems raises privacy concerns.	**High**	Implementing strong data encryption, access controls and audit trails. Ensuring compliance with data protection regulations.
Technology Dependency	According to Dencik et al. (2023), relying too much on AI might lead to reduced human oversight and the potential for AI to make decisions without proper human validation, increasing the risk of errors and unintended consequences.	**High**	Develop possible plans and manual override procedures. Maintain a balanced human-AI operation.
Bias and Fairness	Weidinger et al. (2023) state that AI systems can inadvertently perpetuate biases present in the training data, leading to unfair treatment.	**Medium**	Using diverse datasets for training, conduct regular audits for bias, and deploy algorithms that improve fairness.
Integration and Compatibility	Khlaisamniang et al. (2023), indicate that issues in integrating AI systems with existing IT infrastructure and tech support software can be complex.	**Medium**	Conduct compatibility tests and phased rollouts. Work closely with IT for smooth integration.
Skill Gap and Training	Lack of sufficient skills among staff to manage and troubleshoot AI systems.	**Medium**	Invest in ongoing training programs for tech support staff. Hire specialists where necessary.
Customer Satisfaction	According to Brynjolfsson et al. (2023), AI lacks the ability to genuinely understand and respond to the emotional needs of customers.	**Medium**	Implement hybrid AI-human support models. Monitor customer feedback and adjust AI responses accordingly.

5.4 Emotional State of Customer’s Human Understanding (Support Engineers) vs AI Understanding in Tech Support

Table 4. Comparison of the handling of Human Support Engineers and AI

Aspect	Human Support Engineers	AI
Emotional perception	High empathy and intuition	Relies on NLP and sentiment analysis
Response appropriateness	Personalised, context-aware responses	Standardised, may lack context sensitivity
Speed of response	Slower, dependent on agent availability	Immediate, 24/7 availability
Consistency of response	Variable, dependent on individual agents	High consistency
Handling complex queries	Better at managing complex, multifaceted issues	Limited by predefined algorithms
Adaptability	Highly adaptable, can think outside of the box	Limited to programmed ability
Customer satisfaction	Generally higher when empathy is critical	Lower in emotionally charged situations
Training and Scalability	Requires ongoing training and development	Scalable with initial setup and maintenance
Trust and reliability	Builds trust through personal interaction	Trust varies based on AI performance
Error Rate	Higher potential for human error, especially under stress	Lower error rate for routine tasks, but can make mistakes in unexpected scenarios

As shown in Table 4, both AI and human support engineers bring unique strengths and weaknesses when it comes to handling the emotional state of a customer. Shukla (2022) states that AI uses NLP to detect frustration, satisfaction, or confusion, offering standardised responses and efficient troubleshooting steps. However, AI often lacks the detailed understanding and personalised touch required to fully address emotional concerns. According to Prentice and Nguyen (2020), human support engineers excel in emotional intelligence, adapting their responses based on real-time feedback and providing personalised solutions and empathy. They can effectively defuse frustration, reinforce satisfaction, and clarify confusion through active listening and strong communication.

Therefore, according to Banerjee et al. (2023), while AI offers speed and consistency, the human touch remains crucial for handling complex emotional interactions. This can indicate that a hybrid approach can optimise customer support by combining AI's efficiency with human empathy.

5.5 Mapping the OODA Loop Framework to Support Engineering

Table 5. Summary of OODA loop mapping to Support Engineers

OODA Loop Stage	Support engineer activity	Tasks
Observe	Monitoring systems, gathering data, listening to users	Review system health, record incident details, collect user information
Orient	Analysing data, diagnosing problems, setting priorities	Identify patterns, use diagnostic tools, assess the impact
Decide	Formulating solutions, planning, consulting	List possible solutions, develop an action plan, review with peers
Act	Implementing solutions, communicating, verifying resolution	Execute steps, update users, test the system

Table 5 summarises the OODA loop mapping to support engineers. The provided flowchart in Figure 17 illustrates a model of an AI-driven, explainable customer support system designed to enhance the resolution process by using the OODA loop.

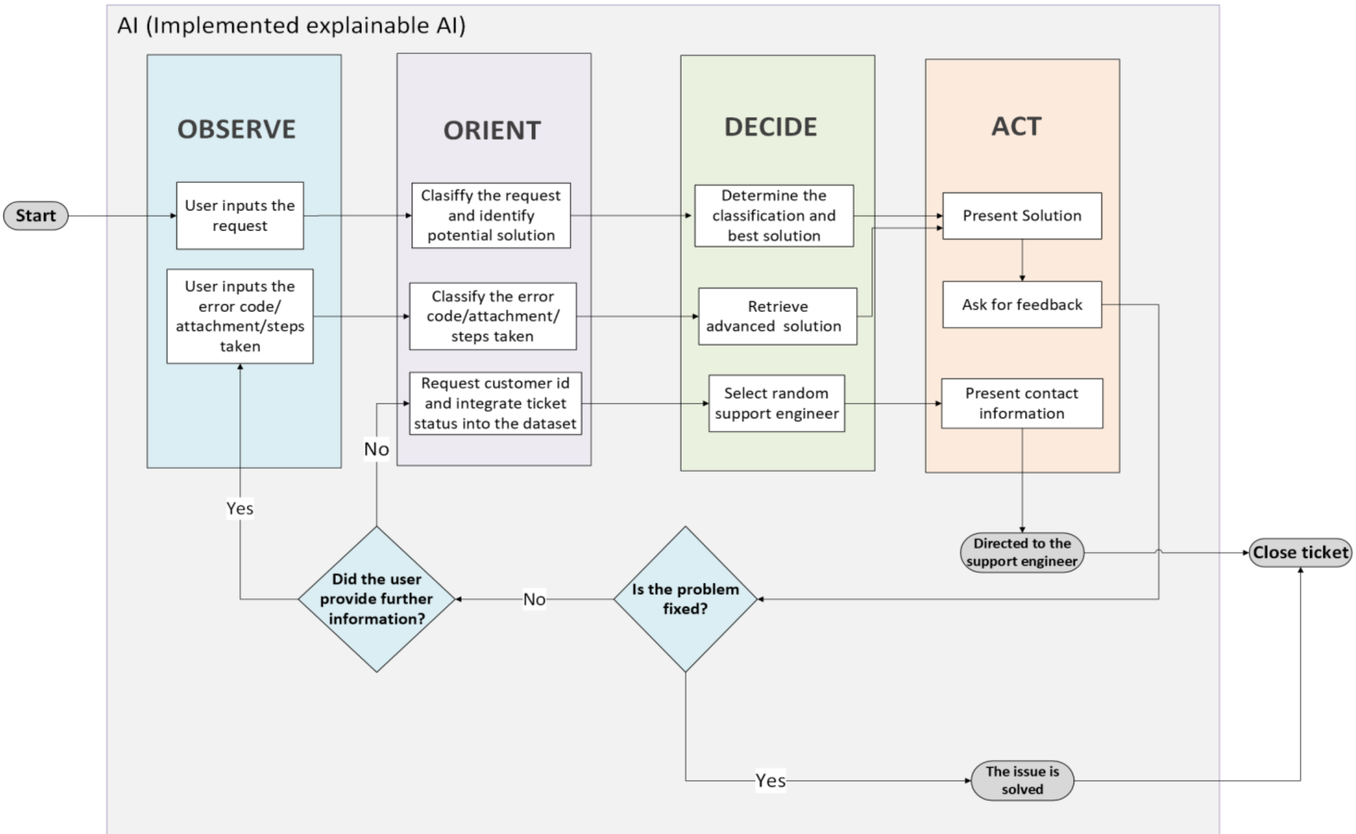


Figure 17. OODA Loop Integration

Observe: In the context of tech support, observation involves the initial intake of a customer's problem. Support engineers use AI to gather data, including direct customer input and historical interaction logs. The key here is to collect sufficient and relevant information that accurately defines the problem.

Orient: Orientation can vary depending on the steps taken. Initially, it requires the AI to analyse the information collected during the observation phase and place it within the broader context of known issues, technical documentation, and classification of technical level, emotional state and category. Depending on the step, if necessary, details are missing, the system prompts the user to provide further information.

Decide: This step involves selecting the best course of action based on the current understanding of the issue. This might mean choosing between escalating the issue to a support engineer or guiding the customer through an advanced solution.

Act: The action phase involves implementing the chosen solution and monitoring its effectiveness. The AI presents the solution to the user, and requests feedback. If the problem is resolved, the issue is marked as solved and the ticket is closed. If not, it directs the issue to a support engineer for further assistance.

Feedback Loop: Crucially, the OODA Loop in tech support is cyclical. After acting, the system decides where to return according to the feedback.

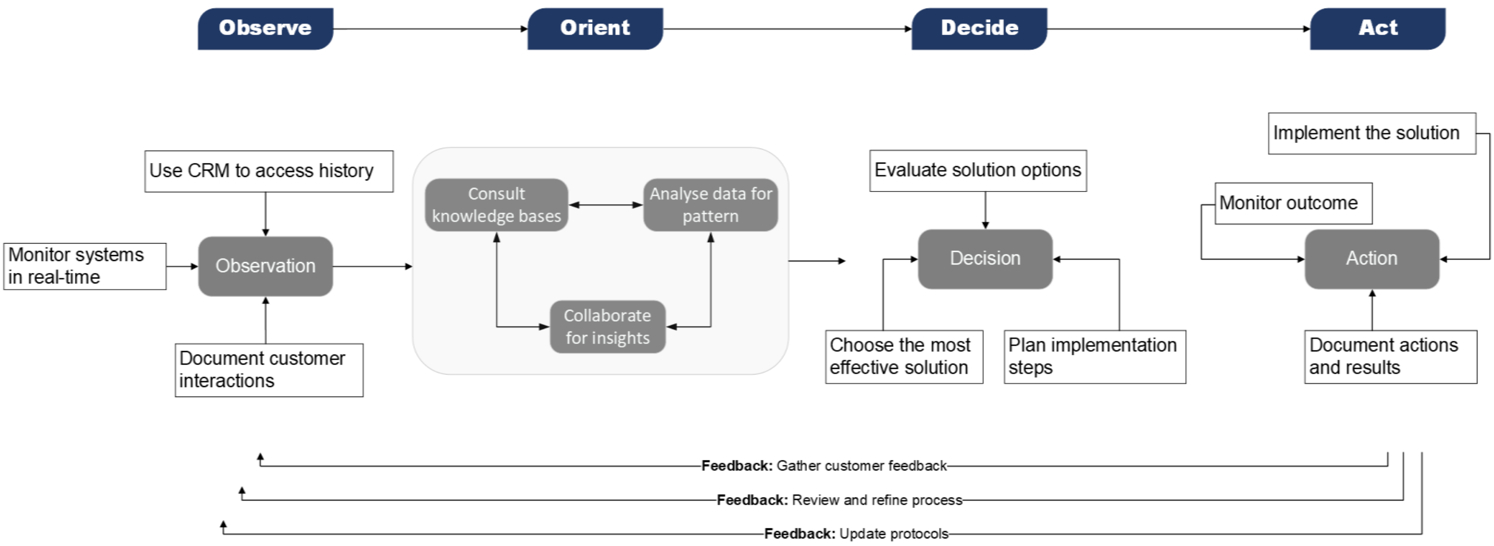


Figure 18. Back-end Process of OODA Loop

5.6 XAI-OODA Hybrid Description based on the case scenario for Support Engineers

The XAI-OODA Hybrid Description in Figure 19 is designed to integrate XAI, the OODA Loop, and a hybrid human-AI collaboration model to enhance the efficiency of support engineering.

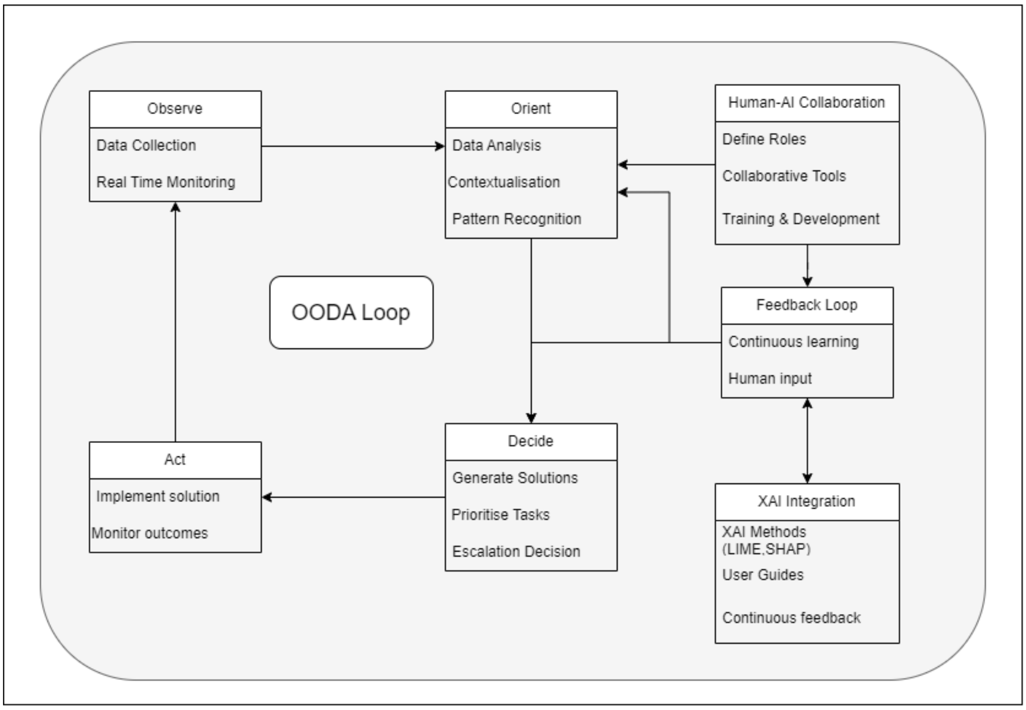


Figure 19. XAI-OODA Hybrid Description

Benefits of using XAI-OODA Hybrid Description:

1. Enhanced decision-making by using the OODA loop to ensure a structured and agile decision-making process
2. Transparency and trust by using XAI and facilitating human-AI collaboration
3. Efficiency and scalability
4. Continuous improvement to ensure the system evolves and improves.

5.6.1 Integration of this description into the case study

The XAI-OODA Hybrid description can be integrated into the AI-based IT Ticketing Support System outlined in section 3.5. by customising each component to improve the ticketing process.

• During the **Observe** phase, the system should automatically collect and monitor real-time data on each requested ticket. The enhanced observation can help in accurately defining the problem and providing context-aware solutions.
• In the **Orient** phase, the system utilises XAI to analyse and contextualise the issue, utilising semantic search and pattern recognition to align historical data with current problems.
• During the **Decide** phase, solutions are created using trained models, predicting the problem's technical level and category, and prioritising tasks based on their criticality. Solutions are explained to users and support staff to ensure transparency. Use the insights from XAI to make informed decisions on whether to provide an advanced AI-driven solution or involve a human support engineer.
• In the **Act** phase, the system either implements solutions directly or guides users on manual interventions, monitoring the outcomes to ensure effectiveness. Provide real-time feedback to the user, explaining the steps taken and why they were chosen. If the solution is unsuccessful, the

system seamlessly transitions to human support with a comprehensive handover of all collected data and decisions made.

This approach emphasises human-AI collaboration by defining clear roles, using collaborative tools, and prioritising continuous training. This integrated approach not only enhances efficiency and decision-making but also fosters trust and scalability, ensuring agility and user-centricity in the IT support process.

5.6.2 Limitations of XAI-OODA Hybrid Description

While there are many advantages of using this approach, the system's effectiveness is heavily reliant on the quality and diversity of training data, making it vulnerable to the system's ability to observe accurately and orient appropriately, leading to less effective decision-making. AI's ability to interpret complex human emotions is limited, potentially leading to less empathetic responses. Scaling human support for complex issues is challenging, and ensuring enough qualified personnel can be difficult. Moreover, there is a risk of overreliance on AI solutions, which might reduce human involvement and control, potentially leading to decreased situational awareness and critical thinking among support staff. Despite XAI efforts, achieving complete user trust and understanding remains a challenge. Recognising these limitations is crucial for setting realistic expectations and guiding future improvements.

6 RESULTS AND DISCUSSION

6.1 Decisions Made and Advantages of the Approach

The project integrated the OODA loop framework in order to enhance decision-making in support engineering tasks, chosen for its agility and relevance to dynamic tech support environments. Synthetic datasets were created due to the lack of suitable public data, which, while practical, was acknowledged as a limitation affecting model accuracy. A hybrid AI-human model was adapted to balance AI's efficiency in handling routine tasks with human intervention for complex and emotionally charged interactions, addressing AI's limitations in emotional intelligence. XAI techniques, particularly LIME, were integrated to ensure transparency and build trust in AI-driven solutions.

By utilising SBERT, the system can understand and match the semantic content of user queries and FAQs, providing highly relevant recommendations. The combination of SBERT for embedding generation and FAISS for similarity search ensures that the recommendation process is both accurate and fast, suitable for real-time applications. FAISS allows the system to handle large datasets efficiently, making it scalable to accommodate growing numbers of FAQs and user requests.

6.2 Output of the Code

```
Welcome to AI-Based IT Ticketing Support System!
Please submit your request (or type 'exit' to quit): Can I upgrade my computer?

Classification:
Technical Level: Advanced
Category: Hardware
Emotional State: Confused

Explanation of Technical Level Prediction:
[('upgrade', -0.28282653675589803), ('computer', -0.10792279241478224), ('my', 0.06370729751090867), ('I', -0.004792438137723698), ('Can', 0.0014583146717593073)]

Explanation of Category Prediction:
[('computer', 0.35800337817556177), ('my', 0.29085278907626416), ('Can', 0.061296695637374755), ('upgrade', 0.0433708469581204), ('I', -0.018619033165507597)]

Explanation of Emotional State Prediction:
[('upgrade', -0.2514631801412457), ('Can', 0.02861670617189312), ('my', -0.014105492872980801), ('computer', 0.007547574748662402), ('I', 0.0025598079912906636)]

I can see that you're confused. I'll do my best to clarify things.
Ticket ID: 1009
Recommended Solution:
Yes, by upgrading RAM can improve multitasking and overall system performance, especially if you currently have insufficient RAM for your tasks.

Did the solution work? (yes/no): no
Please provide the error code if available (or type 'skip' to continue): H09

Advanced Solution:
Solution: You can upgrade your storage drive to increase capacity or improve read/write speeds.

Did the proposed advanced solution work? (yes/no): no
Please enter your Customer ID: C48
Please contact the support engineer for further assistance:
Support Engineer: Emily Moore
Contact: emilymoore@example.com
```

Figure 20. Example output of the code

Figure 20 is the example output of the code. The system offers an output that handles IT requests by combining accurate classification, relevant recommendations, and explainability. By correctly identifying the technical level, category, and emotional state, the system personalises its responses to meet the specific needs of the user. The provision of practical solutions based on the request context ensures users receive meaningful assistance, while the use of LIME to explain predictions builds trust through transparency in the decision-making process. Moreover, the system's empathetic and personalised responses address users' emotional states, enhancing satisfaction and engagement. When initial solutions are insufficient, the system escalates the issue by assigning support engineers, ensuring unresolved problems receive appropriate human intervention. This combination of intelligent, transparent, and user-friendly support makes the system a valuable system for both users seeking help and support staff managing requests.

6.3 Performance Metrics

The performance metrics for the three classification tasks (technical level, category and emotional state) indicate varying levels of success. The accuracy of these tasks ranges from 40% to 60%. The technical level classification model shows moderate performance with an accuracy of 46.7%. The MCC of 0.257 indicates a weak positive correlation between the predicted and actual classifications. The log-loss of 1.121 suggests that the model's probability predictions are not very confident. The category classification model achieves an accuracy of 60%, which is relatively better compared to the technical-level classification. The MCC of 0.522 shows a moderate positive correlation, indicating a better performance than the technical-level classification model. The log-loss of 1.186 still suggests room for improvement in probability predictions. The emotional state classification model exhibits the lowest accuracy at 40%. The MCC of 0.143 indicates a weak positive correlation and the log-loss of 1.204 points to low confidence in predictions. The models show the category classification model is performing the best among the three. These metrics can be improved by increasing the dataset size.

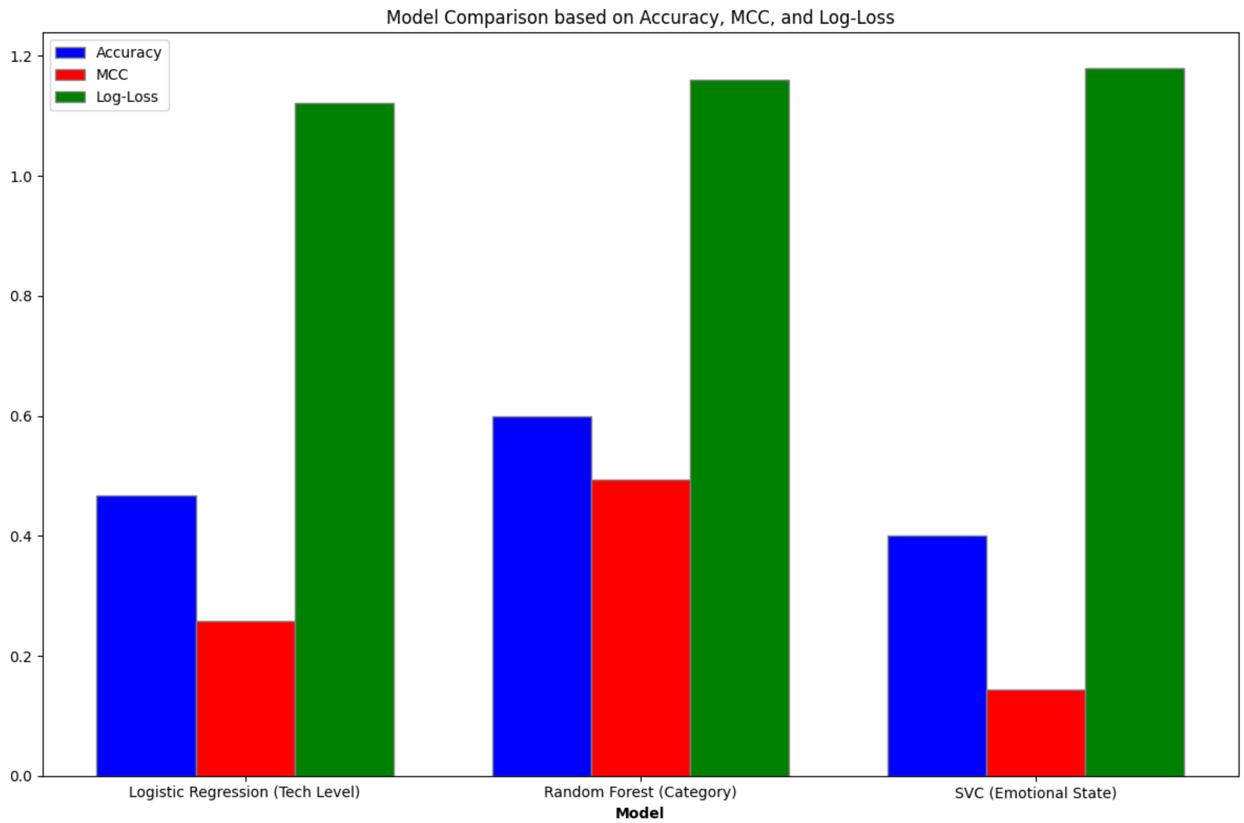


Figure 21. Model Comparison

In Figure 21, the varying performance levels of different models across classification tasks are illustrated. These results suggest that the Random Forest model is the most reliable.

6.4 Potential of Using OODA to Develop Self-Serve AI Systems as Replacements for Human Support Engineers

The potential of self-serve AI systems to replace support engineers depends on factors that can be found on Table 6.

Table 6. Factors of Self-Serve AI systems

Factor	*Description*
Complexity of tasks	While routine tasks can be automated using AI, complex tasks that require deep domain knowledge, creativity or emotional intelligence are more challenging for AI to manage independently.
AI Training and Adaptation	To replace human support engineers, AI models must be trained on extensive datasets that capture a wide range of scenarios. They also need to be capable of learning and adapting to new incidents, which is where reinforcement learning, and continuous feedback loops come into play.
Transparency and Trust	For AI to replace human support, it must be trusted by users. Explainable AI (XAI) techniques are crucial in this regard, as they allow users to understand the AI's decision-making process, thereby increasing their confidence in the system.
User Empowerment	Self-serve systems can empower users by providing them with tools to solve their own problems. This not only reduces the need for human intervention but also enhances user satisfaction by offering immediate solutions.

While the OODA loop framework is robust for modelling support scenarios, there are limitations such as where AI systems can struggle to address uncertain situations. Humans excel in these scenarios due to their ability to apply experience and insight. Human engineers are better equipped to handle emotionally charged interactions. AI, while efficient, may lack the empathy required to manage such situations effectively. The effectiveness of AI in tech support depends on continuous learning from new data. However, this requires ongoing updates to the AI models and systems, which can be demanding on resources.

6.5 Limitations and Strengths

This research explored the integration of GenAI into enterprise tech support operations. The focus is on mapping decision-making frameworks like the OODA Loop to support activities, identifying relevant tasks for AI training, and evaluating the role of AI in augmenting and automating support functions. The study aimed to enhance the understanding of the specific needs of those seeking and providing support and to develop methodologies for assessing the effectiveness of AI systems in this context. However, the rapidly evolving nature of AI technology may render the findings quickly outdated. The research was limited to examining GenAI within tech support scenarios and may not fully account for variations in other enterprise functions or industry-specific challenges. The results' generalisation is further limited by practical problems including the lack of real-world data and the variations in AI deployment across enterprises.

Since there was no dataset available on the web, the datasets were generated. Which lowered the accuracy of the AI model and training. However, with larger datasets, the accuracy and the evaluation metrics of the system will be higher, since the AI will have more input to train itself.

Table 7. Strengths and Limitations

Strengths	1. Innovative approach with the integration of OODA loop using AI in tech support
	2. Comprehensive analysis through literature review and detailed methodology
	3. Exhibits the potential of real-world application
	4. Uses advanced models like XAI and machine learning algorithms
Limitations	1. Focuses mainly on tech support scenarios, which limits the scope
	2. Utilises synthetic dataset, which may not fully capture real-world complexities
	3. Findings may quickly become outdated due to rapid AI developments
	4. The study initially aimed to implement GenAI but had to rely on conventional AI models due to practical constraints.

6.6 Differentiation and Contribution of this Thesis

This thesis differs from existing case studies by providing a focused exploration of how the OODA Loop can be utilised to systematically map and enhance the activities and tasks of Support Engineers using AI technologies. While the case studies primarily discuss the implementation and outcomes of AI systems in support environments, this research explores the theoretical and practical integration of the OODA Loop with XAI to enhance tech support operations. It provides a structured approach to decision-making and task automation that is not addressed in the case studies.

It systematically maps the stages of the OODA Loop to the specific tasks and activities of support engineers, offering a clear framework for enhancing and automating support processes. Moreover, it provides a detailed analysis of the types of tasks and activities that are relevant and irrelevant for training AI models, which is crucial for developing effective AI systems.
This research addresses the emotional intelligence aspect of AI vs human support engineers, indicating the limitations of AI in handling emotionally charged interactions and proposing a hybrid model to mitigate this gap. It emphasises the importance of XAI to build trust and transparency in support systems, which is essential for user acceptance and effective collaboration between human agents and AI.

6.7 Future Work

The findings of this study provide a foundation for further exploration in key areas. Enhancing the dataset with real-world data from various enterprises will improve AI model accuracy and robustness, as the current reliance on synthetic data may not fully capture real-world complexities. Collaborating with companies to test AI models in actual tech support scenarios will assess their effectiveness and scalability.

Future research can focus on integrating advanced AI techniques and GenAI. Using reinforcement learning and transfer learning can improve model adaptability and efficiency. Implementing GenAI models will create more personalised and dynamic responses, enhancing user experience.

It's important to focus on creating a strong hybrid support system that combines AI and human input effectively. By continuously improving and testing this model, we can strike the right balance between AI automation and human interaction to boost productivity without compromising customer satisfaction. Evaluating how this hybrid approach affects response times, problem-solving accuracy, and overall customer satisfaction will help us gain valuable perspectives.

Expanding the scope to other enterprise functions beyond tech support is another important direction. Applying the OODA Loop framework and GenAI models to functions such as marketing, sales, and human resources will show the broader benefits of the proposed approach. Evaluating the specific needs of these functions will allow for personalised AI models.

By pursuing these areas, future research can build on the current study's findings to create more effective, reliable, and transparent AI systems for enterprise tech support and beyond. These efforts will contribute to the evolution of AI applications in the enterprise sector, promoting greater efficiency and enhanced user experiences. These recommendations aim to advance the integration of GenAI and the OODA Loop framework in various enterprise applications, addressing current limitations and exploring new opportunities for innovation and improvement.

7 CONCLUSION

This research explored the integration of AI within enterprise tech support, applying the OODA loop framework to enhance support engineer activities. Findings indicate that AI can significantly augment the efficiency of tech support operations by automating routine tasks and supporting engineers with complex problem-solving processes. Specifically, the application of the OODA loop has shown the potential to optimise decision-making processes and improve response times.

Throughout the study, several challenges were identified, including the complexity of training GenAI systems with high emotional intelligence and ensuring the accuracy of AI-generated responses. Additionally, while GenAI can handle a significant volume of routine queries, its application in emotionally charged or highly complex technical situations remains limited. These challenges underscore the importance of a hybrid AI-human tech support model, where human empathy and AI efficiency coexist to optimise customer service. The importance of XAI was proved by enhancing transparency and trust between users and AI systems.

In conclusion, while this dissertation establishes a foundational understanding of GenAI's role in enterprise tech support, it also highlights the vast potential and existing limitations of AI technologies in complex enterprise settings. Continuous advancements in AI and ongoing research will be crucial to fully realise the capabilities and address the challenges of using GenAI as a co-worker in tech support.

7.1 Appraisal of the research questions

The thesis addressed the research questions through comprehensive analysis and theoretical frameworks. Table 8 provides the outcome of the research questions.

Table 8. The outcome of the research questions

Research Questions	Outcome
1. How can the OODA Loop framework augment, assist, or automate the activities and tasks of Support Engineers?	This question was answered in sections 2.1, 5.5 and 5.6 where the systematic mapping of the OODA stages to support engineer tasks provided a structured approach to automating and augmenting support tasks, showing how AI can optimise decision-making processes.
2. What types of tasks and activities are most relevant for training an AI model to assist, augment, or automate tasks/activities of Support Engineers? a. What tasks and activities would not be relevant/suitable?	This question was answered in section 5.1. and 5.4., where a detailed analysis of relevant and irrelevant tasks for AI in tech support was presented. This distinction helped in focusing AI training on tasks that maximise efficiency without compromising the quality of human interactions.

3. Can we leverage the OODA Loop (or similar) framework to model Support Engineer scenarios, with a view to: a. Replacing the need for support engineers (Self-Serve) b. Augmenting Support Engineers (AI-Supported Support Engineers)	These questions were answered in sections 2.7, 5.1, 5.6 and 6.4 where the application of the OODA loop was explored to create both AI-driven and hybrid human-AI support scenarios. In these sections, it was discussed how such models can either replace routine support tasks or augment the capabilities of humans.

Main body word count: 5,833

REFERENCES

Al-Mekhlal, M., Al-Buraik, M. and Al-Lubli, M., 2023. Digital Transformation: AI-Powered Bot Solutions and Automation for Customer Services [online]. Available from: https://www.semanticscholar.org/paper/Digital-Transformation%3A-AI-Powered-Bot-Solutions-Al-Mekhlal-Al-Buraik/6a17a3240f3f5caeb440445186cbf2c4bb1585d9.

Arthur, L., Costello, J., Hardy, J., O'Brien, W., Rea, J., Rees, G. and Ganev, G., 2023. On the Challenges of Deploying Privacy-Preserving Synthetic Data in the Enterprise [online]. arXiv.org. Available from: https://arxiv.org/abs/2307.04208.

Banerjee, D., Poser, M., Wiethof, C., Subramanian, V. S., Paucar, R., Bittner, E. a. C. and Biemann, C., 2023. A System for Human-AI collaboration for Online Customer Support [online]. arXiv.org. Available from: https://arxiv.org/abs/2301.12158.

Begicheva, S. and Zhukovskaya, I., 2022. Technical support customer satisfaction research. *Cifrovye Modeli I Rešeniâ* [online], 1 (2). Available from: https://doi.org/10.29141/2782-4934-2022-1-2-3.

Boyd, J., 2012. The Essence of Winning and Losing [online]. Available from: https://fasttransients.files.wordpress.com/2010/03/essence_of_winning_losing.pdf.

Brehmer, B., 2005. *The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control* [online]. 10th International Command and Control Research and Technology Symposium. Available from: http://www.dodccrp.org/events/10th_ICCRTS/CD/papers/365.pdf.

Brynjolfsson, E., Li, D. and Raymond, L., 2023a. Generative AI at Work [online]. Available from: https://www.semanticscholar.org/paper/Generative-AI-at-Work-Brynjolfsson-Li/5a23a489bb9e742edacc8b8e778b06e1594365d3.

Cevallos, A., Latorre, L., Alicandro, G., Wanner, Z., Cerrato, I., Zarate, J. D., Alvarez, J., Villacreses, K., Pfeifer, M., Gutierrez, M., Villanueva, V., Rivera-Fournier, A., Riobó, A., Pombo, C., Puerto, F. and Breuning, J. R., 2023. Tech Report Generative AI [online]. Available from: https://publications.iadb.org/en/tech-report-generative-ai.

De Andrade, I. M. and Tumelero, C., 2022. Increasing customer service efficiency through artificial intelligence chatbot [online]. Available from: https://www.semanticscholar.org/paper/Increasing-customer-service-efficiency-through-Andrade-Tumelero/6f2461a9ed5943867db49002fe8e4976b1375adf.

Debbah, M., 2023. Large Language Models for Telecom [online]. IEEE Conference Publication | IEEE Xplore. Available from: https://ieeexplore.ieee.org/document/10305960 [Accessed 25 Jun 2024].

Defense Advanced Research Projects Agency, 2024. Explainable Artificial Intelligence (XAI) [online]. DARPA. Available from: https://www.darpa.mil/program/explainable-artificial-intelligence [Accessed 23 Jul 2024].

Dencik, J., Goehring, B. and Marshall, A., 2023. Managing the emerging role of generative AI in next-generation business. Strategy & Leadership [online], 51 (6), 30–36. Available from: https://doi.org/10.1108/sl-08-2023-0079.

Houde, S., Liao, Q., Martino, J., Muller, M. J., Piorkowski, D., Richards, J. T., Weisz, J. D. and Zhang, Y., 2020. Business (mis)Use Cases of Generative AI [online]. Available from: https://www.semanticscholar.org/paper/Business-(mis)Use-Cases-of-Generative-AI-Houde-Liao/274a7839ca00ed0291d904b9ca6ee9bca9d8ad6d.

Jiang, J., Karran, A. J., Coursaris, C. K., Léger, P.-M. and Beringer, J., 2022. A Situation Awareness Perspective on Human-AI Interaction: Tensions and Opportunities. *International Journal of Human-computer Interaction* [online], 39 (9), 1789–1806. Available from: https://doi.org/10.1080/10447318.2022.2093863.

Khlaisamniang, P., Khomduean, P., Saetan, K. and Wonglapsuwan, S., 2023. Generative AI for Self-Healing Systems [online]. IEEE Conference Publication | IEEE Xplore. Available from: https://ieeexplore.ieee.org/document/10354608 [Accessed 25 Jun 2024].

Kim, S. S. Y., Watkins, E. A., Russakovsky, O., Fong, R. and Monroy-Hernández, A., 2023. “Help Me Help the AI”: Understanding How Explainability Can Support Human-AI Interaction. [online]. Available from: https://doi.org/10.1145/3544548.3581001.

Lai, V., Zhang, Y., Chen, C., Liao, Q. V. and Tan, C., 2023. Selective Explanations: Leveraging Human Input to Align Explainable AI. Proceedings of the ACM on Human-computer Interaction [online], 7 (CSCW2), 1–35. Available from: https://doi.org/10.1145/3610206.

Ma, X., Fang, G. and Wang, X., 2023. LLM-Pruner: On the Structural Pruning of Large Language Models [online]. arXiv.org. Available from: https://arxiv.org/abs/2305.11627.

Omar, F., 2024. The Future Role of Support Engineers in the World of AI [online]. Available from: https://www.linkedin.com/pulse/future-role-support-engineers-world-ai-farhad-omar-dvvmc/.

Prabhakar, S., 2018. Futuristic Automated Application for Travel & Hospitality Managed Operation [online]. Available from: https://www.semanticscholar.org/paper/Futuristic-Automated-Application-for-Travel-%26-Prabhakar/75c669ab957a87eaca72727e68cbbdd66f4e4edc.

Prakash, Babu, S. M., Kumar, P., Devi, S., Reddy, K. P. and Satish, M., 2023. Predicting Consumer Behaviour with Artificial Intelligence [online]. IEEE Conference Publication | IEEE Xplore. Available from: https://ieeexplore.ieee.org/document/10346916 [Accessed 25 Jun 2024].

Prentice, C. and Nguyen, M., 2020. Engaging and retaining customers with AI and employee service. Journal of Retailing and Consumer Services [online], 56, 102186. Available from: https://www.sciencedirect.com/science/article/pii/S0969698920306184?via%3Dihub.

Shukla, A., 2022. Utilizing AI and Machine Learning for Human Emotional Analysis through Speech-to-Text Engine Data Conversion [online]. Available from: https://www.semanticscholar.org/paper/Utilizing-AI-and-Machine-Learning-for-Human-through-Shukla/133158d48b7f5ba68a8a4ab1ff40d7f23831f679.

Sousa, D. N., Brito, M. A. and Argainha, C., 2019. Virtual customer service. [online]. Available from: https://doi.org/10.1145/3361785.3361805.

Weidinger, L., Rauh, M., Marchal, N., Manzini, A., Hendricks, L. A., Mateos-Garcia, J., Bergman, S., Kay, J., Griffin, C., Bariach, B., Gabriel, I., Rieser, V. and Isaac, W., 2023. Sociotechnical Safety Evaluation of Generative AI Systems [online]. arXiv.org. Available from: https://arxiv.org/abs/2310.11986.

Zheng, N.-N., Liu, Z.-Y., Ren, P.-J., Ma, Y.-Q., Chen, S.-T., Yu, S.-Y., Xue, J.-R., Chen, B.-D. and Wang, F.-Y., 2017. Hybrid-augmented intelligence: collaboration and cognition. Frontiers of Information Technology & Electronic Engineering [online], 18 (2), 153–179. Available from: https://doi.org/10.1631/fitee.1700053.

APPENDIX A: LIST OF CONTENT OF LARGE FILES

Large File Zip Name: IT_Support_Dissertation(s5619032)

Contents:

- AI_Ticket_Support.ipynb
- ticket_database.csv
- ticket_base_proposed.csv

How to run:

This project uses Google Colab to analyse and enhance support ticket handling using two datasets: ticket_base_proposed.csv and ticket_database.csv. To run the project, ensure you have Python 3., a Google Colab account, and installed the required libraries (pandas, numpy, scikit-learn, matplotlib and more). Open the notebook, upload the datasets, and follow the workflow within the notebook. For detailed steps, refer to the comments inside the notebook.

APPENDIX B: ARTEFACT

AI-Based IT Ticketing Support System

1. Install the required libraries

```
[1] pip install lime
```

Show hidden output

```
pip install sentence-transformers
```

Show hidden output

```
pip install faiss-cpu
```

Show hidden output

```
[4] import pandas as pd
    import numpy as np
    from sklearn.linear_model import LogisticRegression
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.preprocessing import LabelEncoder
    from sklearn.metrics import accuracy_score, classification_report
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.pipeline import Pipeline
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.svm import SVC
    from sentence_transformers import SentenceTransformer
    import faiss
    import random
    import os
    import matplotlib.pyplot as plt
    from sklearn.metrics import confusion_matrix
    import seaborn as sns
    from lime.lime_text import LimeTextExplainer
    from sklearn.metrics import log_loss
    from sklearn.metrics import matthews_corrcoef
```

2. Data Loading and Preprocessing

```
[5] # Load datasets
    ticket_data = pd.read_csv('ticket_database.csv')
    proposed_data = pd.read_csv('ticket_base_proposed.csv')

    # Handling missing values
    ticket_data.fillna('', inplace=True)
```

```
# Plot the distribution of categorical variables in ticket_database with horizontal bars
fig, axes = plt.subplots(2, 2, figsize=(14, 10))

sns.barplot(y=ticket_data['Technical Level'].value_counts().index,
            x=ticket_data['Technical Level'].value_counts().values, ax=axes[0, 0], palette='viridis')
axes[0, 0].set_title('Distribution of Technical Level')
axes[0, 0].set_xlabel('Count')
axes[0, 0].set_ylabel('Technical Level')

sns.barplot(y=ticket_data['Category'].value_counts().index,
            x=ticket_data['Category'].value_counts().values, ax=axes[0, 1], palette='muted')
axes[0, 1].set_title('Distribution of Category')
axes[0, 1].set_xlabel('Count')
axes[0, 1].set_ylabel('Category')

sns.barplot(y=ticket_data['Emotional State'].value_counts().index,
            x=ticket_data['Emotional State'].value_counts().values, ax=axes[1, 0], palette='coolwarm')
axes[1, 0].set_title('Distribution of Emotional State')
axes[1, 0].set_xlabel('Count')
axes[1, 0].set_ylabel('Emotional State')

sns.barplot(y=ticket_data['Status'].value_counts().index,
            x=ticket_data['Status'].value_counts().values, ax=axes[1, 1], palette='deep')
axes[1, 1].set_title('Distribution of Status')
axes[1, 1].set_xlabel('Count')
axes[1, 1].set_ylabel('Status')

plt.tight_layout()
plt.show()
```

3. Classification

```
[7] # Encoding categorical variables
    le_technical = LabelEncoder()
    le_category = LabelEncoder()
    le_emotional = LabelEncoder()

    ticket_data['Technical Level Encoded'] = le_technical.fit_transform(ticket_data['Technical Level'])
    ticket_data['Category Encoded'] = le_category.fit_transform(ticket_data['Category'])
    ticket_data['Emotional State Encoded'] = le_emotional.fit_transform(ticket_data['Emotional State'])
```

```
[8] # Splitting the data for classification tasks
    X_train, X_test, y_train_tech, y_test_tech = train_test_split(ticket_data['FaQ'], ticket_data['Technical Level Encoded'], test_size=0.2, random_state=42)
    _, _, y_train_cat, y_test_cat = train_test_split(ticket_data['FaQ'], ticket_data['Category Encoded'], test_size=0.2, random_state=42)
    _, _, y_train_emo, y_test_emo = train_test_split(ticket_data['FaQ'], ticket_data['Emotional State Encoded'], test_size=0.2, random_state=42)
```

4. Train and Evaluate the Model

```
[9] # Define parameter grids for hyperparameter tuning
    param_grid_logistic = {'clf__C': [0.01, 0.1, 1, 10, 100], 'clf__solver': ['liblinear', 'lbfgs'], 'clf__max_iter': [100, 500, 1000]}
    param_grid_random_forest = {'clf__n_estimators': [100, 200, 300], 'clf__max_depth': [None, 10, 20, 30]}
    param_grid_svc = {'clf__C': [0.1, 1, 10], 'clf__kernel': ['linear', 'rbf']}
```

```
def train_evaluate_model_with_pipeline(X_train, y_train, X_test, y_test, target_names, model, param_grid, plot_confusion=False):
    pipeline = Pipeline([
        ('tfidf', TfidfVectorizer(max_features=5000, ngram_range=(1, 2))),
        ('clf', model)
    ])

    grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
    grid_search.fit(X_train, y_train)
    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    y_pred_proba = best_model.predict_proba(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=target_names)
    mcc = matthews_corrcoef(y_test, y_pred)
    logloss = log_loss(y_test, y_pred_proba)

    if plot_confusion:
        cm = confusion_matrix(y_test, y_pred)
        plt.figure(figsize=(10, 7))
        sns.heatmap(cm, annot=True, fmt="d", xticklabels=target_names, yticklabels=target_names)
        plt.xlabel('Predicted')
        plt.ylabel('True')
        plt.title(f'Confusion Matrix for {model.__class__.__name__}')
        plt.show()

    return best_model, accuracy, report, mcc, logloss
```

```
# Train and evaluate models, including MCC and Log-Loss calculation

tech_level_model, tech_accuracy, tech_report, tech_mcc, tech_logloss = train_evaluate_model_with_pipeline(
    X_train, y_train_tech, X_test, y_test_tech,
    target_names=le_technical.classes_,
    model=LogisticRegression(),
    param_grid=param_grid_logistic,
    plot_confusion=True
)

category_model, cat_accuracy, cat_report, cat_mcc, cat_logloss = train_evaluate_model_with_pipeline(
    X_train, y_train_cat, X_test, y_test_cat,
    target_names=le_category.classes_,
    model=RandomForestClassifier(),
    param_grid=param_grid_random_forest,
    plot_confusion=True
)

emotional_model, emo_accuracy, emo_report, emo_mcc, emo_logloss = train_evaluate_model_with_pipeline(
    X_train, y_train_emo, X_test, y_test_emo,
    target_names=le_emotional.classes_,
    model=SVC(probability=True),
    param_grid=param_grid_svc,
    plot_confusion=True
)
```

```python
# Displaying metrics for Technical Level classification
print("Technical Level Classification:")
print(f"Accuracy: {tech_accuracy}")
print("Classification Report:")
print(tech_report)
print(f"Matthews Correlation Coefficient (MCC): {tech_mcc}")
print(f"Log-Loss: {tech_logloss}")

# Displaying metrics for Category classification
print("Category Classification:")
print(f"Accuracy: {cat_accuracy}")
print("Classification Report:")
print(cat_report)
print(f"Matthews Correlation Coefficient (MCC): {cat_mcc}")
print(f"Log-Loss: {cat_logloss}")

# Displaying metrics for Emotional State classification
print("Emotional State Classification:")
print(f"Accuracy: {emo_accuracy}")
print("Classification Report:")
print(emo_report)
print(f"Matthews Correlation Coefficient (MCC): {emo_mcc}")
print(f"Log-Loss: {emo_logloss}")
```

5. Semantic Search Setup using Sentence-BERT and FAISS

```python
# Semantic search setup using Sentence-BERT and FAISS
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

# Generate embeddings for all FAQs in the ticket database
ticket_embeddings = model.encode(ticket_data['FaQ'].tolist())

# Build the FAISS index
dimension = ticket_embeddings.shape[1]
index = faiss.IndexFlatL2(dimension)
index.add(ticket_embeddings)
```

6. Problem Resolution Functions

```
[ ] def classify_problem(problem_statement):
        problem_tfidf = tech_level_model.named_steps['tfidf'].transform([problem_statement])
        tech_pred = tech_level_model.named_steps['clf'].predict(problem_tfidf)[0]
        cat_pred = category_model.named_steps['clf'].predict(problem_tfidf)[0]
        emo_pred = emotional_model.named_steps['clf'].predict(problem_tfidf)[0]

        return {
            'Technical Level': le_technical.inverse_transform([tech_pred])[0],
            'Category': le_category.inverse_transform([cat_pred])[0],
            'Emotional State': le_emotional.inverse_transform([emo_pred])[0]
        }
```

```
[14] def explain_prediction(model, problem_statement):
         explainer = LimeTextExplainer(class_names=model.named_steps['clf'].classes_)
         exp = explainer.explain_instance(problem_statement, model.predict_proba, num_features=10)
         return exp
```

```
[15] def find_solution(problem_statement):
         problem_embedding = model.encode([problem_statement])
         distances, indices = index.search(problem_embedding, 1)
         closest_match_idx = indices[0][0]

         return ticket_data.iloc[closest_match_idx]['Answer'], ticket_data.iloc[closest_match_idx]['Ticket_ID']

     def recommend_solution(problem_statement):
         classification = classify_problem(problem_statement)
         solution, ticket_id = find_solution(problem_statement)

         return {
             'Classification': classification,
             'Recommended Solution': solution,
             'Ticket_ID': ticket_id
         }
```

```
[16] def get_proposed_solution(error_code):
         proposed_solution_row = proposed_data[proposed_data['Error Code'] == error_code]
         if not proposed_solution_row.empty:
             return proposed_solution_row.iloc[0]['Proposed Solution'], proposed_solution_row.iloc[0]['Support Engineer'], proposed_solution_row.iloc[0]['Contact']
         else:
             # Return a default message along with the random support engineer
             solution, contact = get_random_support_engineer()
             return None, solution, contact
```

```
def get_random_support_engineer():
    random_engineer = proposed_data.sample(n=1).iloc[0]
    return random_engineer['Support Engineer'], random_engineer['Contact']

def update_support_engineer_assignment(ticket_id, customer_id, assigned_engineer, predicted_category):
    # Check if the file exists
    if os.path.exists('support_engineer_assignments.csv'):
        assignment_data = pd.read_csv('support_engineer_assignments.csv')
    else:
        assignment_data = pd.DataFrame(columns=['ticket_id', 'customer_id', 'assigned_engineer', 'predicted_category', 'status'])

    # Add new assignment
    new_assignment = pd.DataFrame({
        'ticket_id': [ticket_id],
        'customer_id': [customer_id],
        'assigned_engineer': [assigned_engineer],
        'predicted_category': [predicted_category],
        'status': ['Pending']
    })

    # Concatenate the new assignment to the existing data
    assignment_data = pd.concat([assignment_data, new_assignment], ignore_index=True)

    # Save the updated assignments
    assignment_data.to_csv('support_engineer_assignments.csv', index=False)
```

7. Main Application Flow

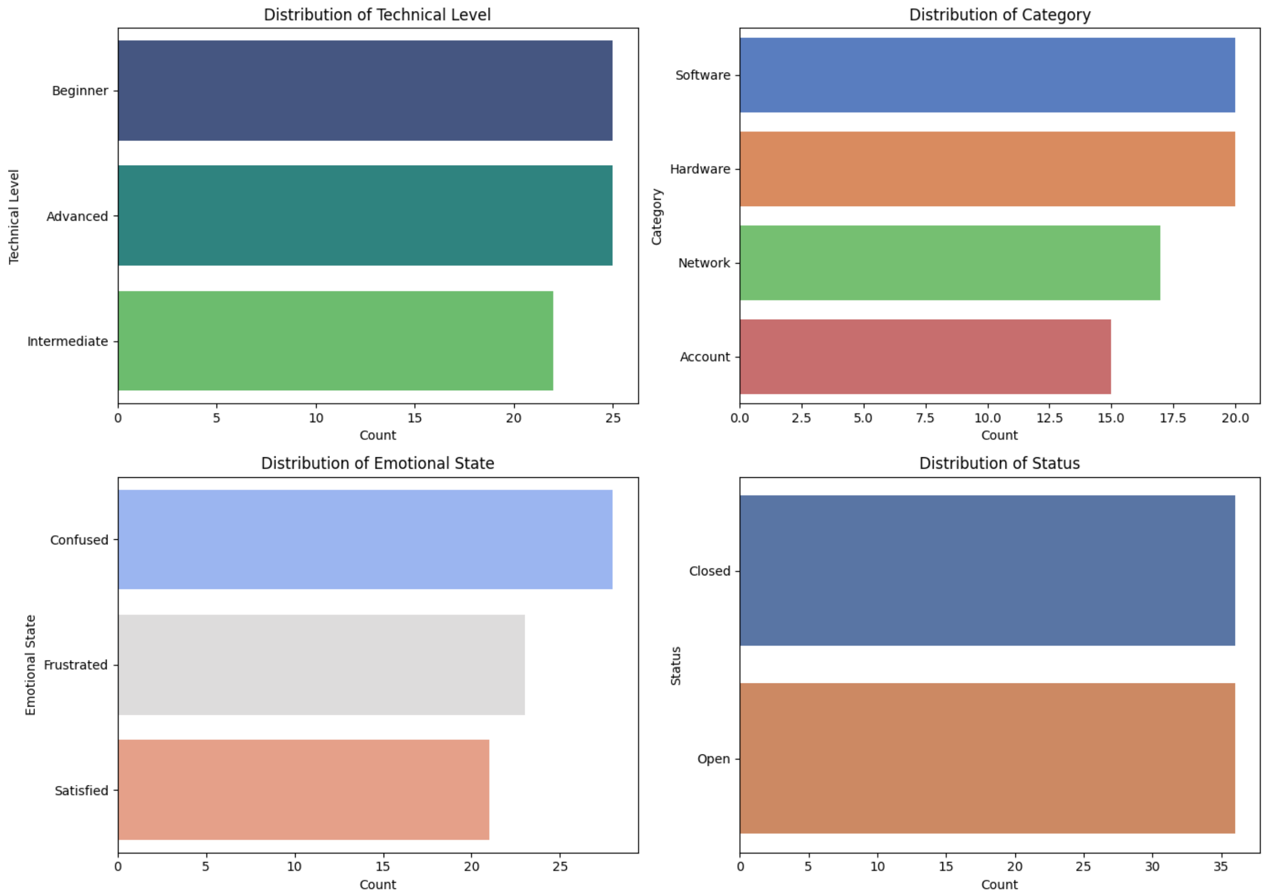
```
def main():
    print("Welcome to AI-Based IT Ticketing Support System!")

    while True:
        problem_statement = input("Please submit your request (or type 'exit' to quit): ")
        if problem_statement.lower() == 'exit':
            break
        recommendation = recommend_solution(problem_statement)

        # Determine the appropriate response based on the emotional state
        emotional_state = recommendation['Classification']['Emotional State']
        if emotional_state == 'Frustrated':
            emotional_response = "I'm sorry, I can see that you're frustrated. Let's try to resolve this quickly."
        elif emotional_state == 'Confused':
            emotional_response = "I can see that you're confused. I'll do my best to clarify things."
        elif emotional_state == "Satisfied":
            emotional_response = "Thanks for contacting us. We will help you immediately."

        print("\nClassification:")
        print(f"Technical Level: {recommendation['Classification']['Technical Level']}")
        print(f"Category: {recommendation['Classification']['Category']}")
        print(f"Emotional State: {recommendation['Classification']['Emotional State']}")

        tech_exp = explain_prediction(tech_level_model, problem_statement)
        cat_exp = explain_prediction(category_model, problem_statement)
        emo_exp = explain_prediction(emotional_model, problem_statement)

        print("\nExplanation of Technical Level Prediction:")
        print(tech_exp.as_list())

        print("\nExplanation of Category Prediction:")
        print(cat_exp.as_list())

        print("\nExplanation of Emotional State Prediction:")
        print(emo_exp.as_list())

        print("\n" + emotional_response)
        print("Ticket ID: " + str(recommendation['Ticket_ID']))
        print("Recommended Solution:")
        print(recommendation['Recommended Solution'])
```

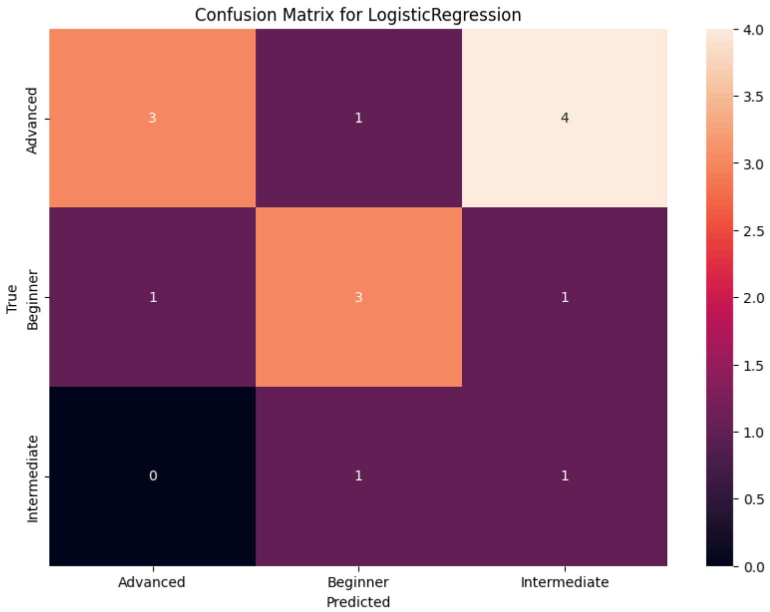
```
        feedback = input("\nDid the solution work? (yes/no): ").strip().lower()
        if feedback == 'yes':
            print("Great! Your ticket is now closed. Have a good day!")
        else:
            error_code = input("Please provide the error code if available (or type 'skip' to continue): ").strip().upper()
            if error_code != 'skip':
                proposed_solution, engineer, contact = get_proposed_solution(error_code)
                if proposed_solution:
                    print("\nAdvanced Solution:")
                    print(f"Solution: {proposed_solution}")
                    feedback = input("\nDid the proposed advanced solution work? (yes/no): ").strip().lower()
                    if feedback == 'yes':
                        print("Great! Your ticket is now closed.")
                    else:
                        customer_id = input("Please enter your Customer ID: ").strip()
                        print(f"Please contact the support engineer for further assistance:\nSupport Engineer: {engineer}\nContact: {contact}")
                        update_support_engineer_assignment(recommendation['Ticket_ID'], customer_id, engineer, recommendation['Classification']['Category'])
                else:
                    customer_id = input("Please enter your Customer ID: ").strip()
                    print(f"Please contact the support engineer for further assistance:\nSupport Engineer: {engineer}\nContact: {contact}")
                    update_support_engineer_assignment(recommendation['Ticket_ID'], customer_id
                    , engineer, recommendation['Classification']['Category'])
            else:
                customer_id = input("Please enter your Customer ID: ").strip()
                engineer, contact = get_random_support_engineer()
                print(f"Please contact the support engineer for further assistance:\nSupport Engineer: {engineer}\nContact: {contact}")
                update_support_engineer_assignment(recommendation['Ticket_ID'], customer_id, engineer, recommendation['Classification']['Category'])
        print("\n")

if __name__ == "__main__":
    main()
```

8. Plot of Performances

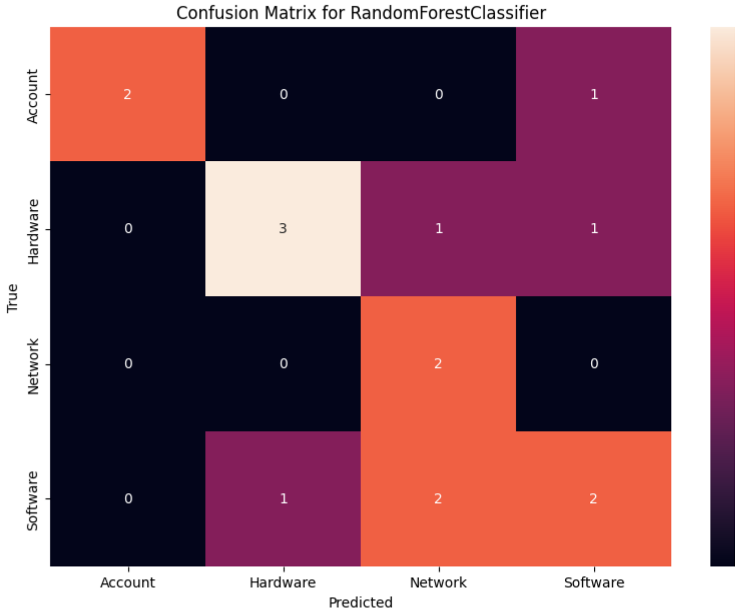
```
[19] model_names = ['Logistic Regression (Tech Level)', 'Random Forest (Category)', 'SVC (Emotional State)']

     # Metrics for each model
     accuracies = [tech_accuracy, cat_accuracy, emo_accuracy]
     mccs = [tech_mcc, cat_mcc, emo_mcc]
     loglosses = [tech_logloss, cat_logloss, emo_logloss]
     barWidth = 0.25

     # Set position of bar on X axis
     r1 = np.arange(len(accuracies))
     r2 = [x + barWidth for x in r1]
     r3 = [x + barWidth for x in r2]

     # Plot
     plt.figure(figsize=(12, 8))
     plt.bar(r1, accuracies, color='b', width=barWidth, edgecolor='grey', label='Accuracy')
     plt.bar(r2, mccs, color='r', width=barWidth, edgecolor='grey', label='MCC')
     plt.bar(r3, loglosses, color='g', width=barWidth, edgecolor='grey', label='Log-Loss')

     # Add xticks on the middle of the group bars
     plt.xlabel('Model', fontweight='bold')
     plt.xticks([r + barWidth for r in range(len(accuracies))], model_names)

     # Create graphic
     plt.legend()
     plt.title('Model Comparison based on Accuracy, MCC, and Log-Loss')
     plt.tight_layout()
     plt.show()
```

APPENDIX C: OUTPUT OF THE ARTEFACT

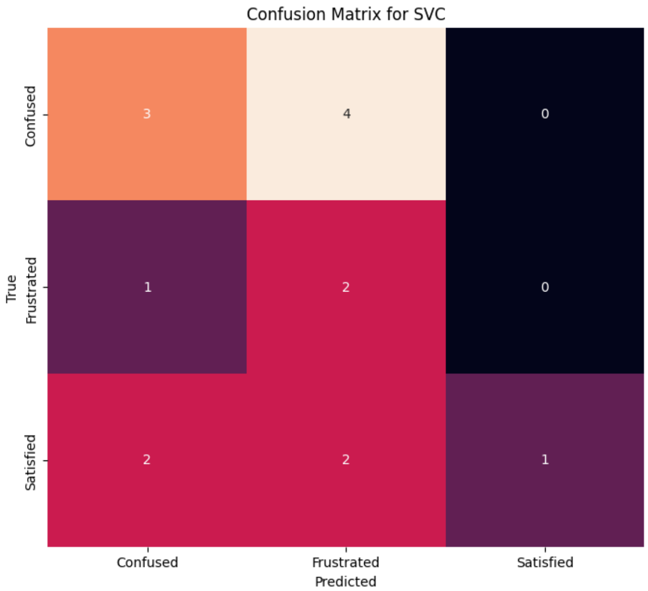





```
Technical Level Classification:

Accuracy: 0.4666666666666667

Classification Report:

              precision    recall  f1-score   support


    Advanced       0.75      0.38      0.50         8

    Beginner       0.60      0.60      0.60         5

Intermediate       0.17      0.50      0.25         2


    accuracy                           0.47        15

   macro avg       0.51      0.49      0.45        15

weighted avg       0.62      0.47      0.50        15


Matthews Correlation Coefficient (MCC): 0.25756371316446414

Log-Loss: 1.1212236376022808
```

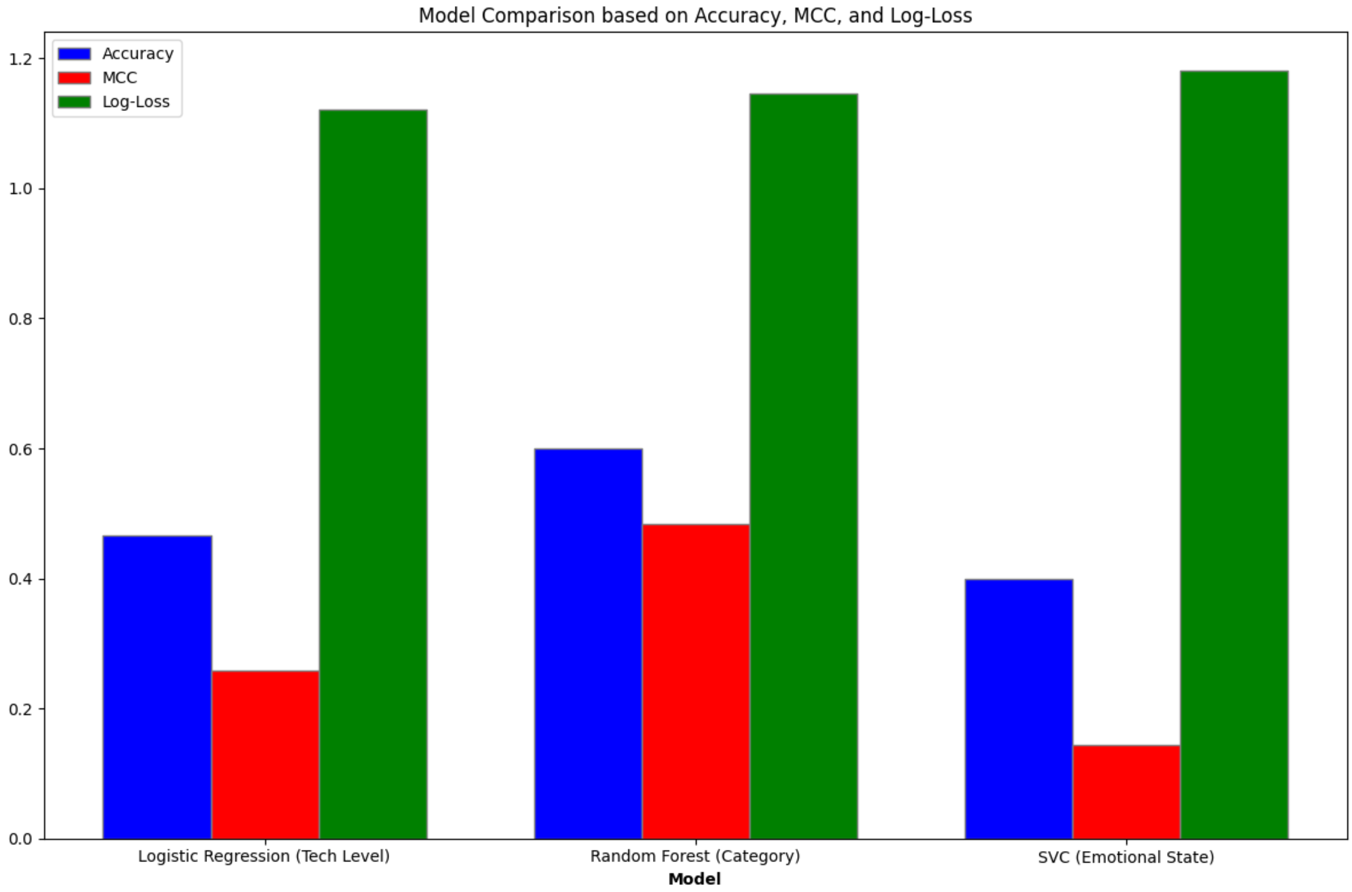



```
Category Classification:

Accuracy: 0.6

Classification Report:

              precision    recall  f1-score   support


     Account       0.50      1.00      0.67         3

    Hardware       1.00      0.60      0.75         5

     Network       0.25      0.50      0.33         2

    Software       1.00      0.40      0.57         5


    accuracy                           0.60        15

   macro avg       0.69      0.62      0.58        15

weighted avg       0.80      0.60      0.62        15


Matthews Correlation Coefficient (MCC): 0.5217491947499509

Log-Loss: 1.185800103499989
```




```
Emotional State Classification:

Accuracy: 0.4

Classification Report:

              precision    recall  f1-score   support


    Confused       0.50      0.43      0.46         7

  Frustrated       0.25      0.67      0.36         3

   Satisfied       1.00      0.20      0.33         5


    accuracy                           0.40        15

   macro avg       0.58      0.43      0.39        15

weighted avg       0.62      0.40      0.40        15


Matthews Correlation Coefficient (MCC): 0.14318535024651025

Log-Loss: 1.204353517820312
```

```
Welcome to AI-Based IT Ticketing Support System!

Please submit your request (or type 'exit' to quit):    How can I connect to a bluetooth with my device?


Classification:

Technical Level: Intermediate

Category: Network

Emotional State: Frustrated


Explanation of Technical Level Prediction:

[('connect', -0.10957845201882152), ('device', -0.06275440393790174), ('with', -0.05609720278040101), ('bluetooth', -0.0396106338181189), ('can', -0.03449879529912197), ('my', 0.011304812966158814), ('t
o', -0.007481406462383431), ('How', 0.006039386026017305), ('a', 0.00023478638475559968), ('I', 0.00022742932272879884)]


Explanation of Category Prediction:

[('How', -0.11116337072165168), ('my', 0.05745939957471959), ('to', -0.03275835226482977), ('can', -0.010448630018094372), ('connect', -0.01031350356493145), ('bluetooth', 0.007159809987404713), ('with',
-0.003971028119099355), ('I', -0.001922991616978957), ('a', -0.0016084471727810996), ('device', 0.0014878664405826266)]


Explanation of Emotional State Prediction:

[('connect', -0.04108726753175542), ('bluetooth', -0.029273833090325514), ('device', -0.025692811573368387), ('to', -0.02181920153969842), ('with', -0.018335422268148344), ('my', -0.01637514670739655),
('How', 0.007775572705087413 5), ('can', -0.003059077497518788), ('I', 0.0007228969299969398), ('a', 0.0005642680594711613)]



        I'm sorry, I can see that you're frustrated. Let's try to resolve this quickly.

        Ticket ID: 1071

        Recommended Solution:

        1. Ensure Bluetooth is enabled on both devices. 2. Restart both devices. 3. Remove any existing pairings and try pairing again.


        Did the solution work? (yes/no): no

        Please provide the error code if available (or type 'skip' to continue): N71


        Advanced Solution:

        Solution: 1. Update Bluetooth drivers or firmware. 2. Reset network settings on the device. 3. Check for interference from other wireless devices.


        Did the proposed advanced solution work? (yes/no): no

        Please enter your Customer ID: C1010

        Please contact the support engineer for further assistance:

        Support Engineer: Emily Moore

        Contact: emilymoore@example.com
```


Model Comparison based on Accuracy, MCC, and Log-Loss
Accuracy
MCC
Log-Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression (Tech Level)
Random Forest (Category)
SVC (Emotional State)
Model

APPENDIX D: PROJECT PROPOSAL

Bournemouth University

Department of Computing and Informatics

2023-24 Academic Year Individual Masters Project

Project Proposal Form

Please refer to the **Project Handbook Section 4** when completing this form. Note that your proposal should be your own original work and you must cite sources in line with university guidance on **referencing and plagiarism**[1].

Degree Title: **MSc Data Science and Artificial Intelligence**	**Student's Name:** Yasemin Karaca
	Supervisor's Name: Huseyin Dogan
	Project Title/Area: Generative AI in the Enterprise as a co-worker for tech support

Section 1: Project Overview

1.1 Problem definition - use one sentence to summarise the problem:

The problem is to explore how the OODA (Observe, Orient, Decide, Act) loop framework can be applied to leverage generative AI models to assist, augment, or automate the tasks and activities of support engineers or sales professionals in an enterprise setting.

1.2 Project description - briefly explain your project:

The project will explore how generative AI models can be enhanced to assist, augment, or automate the tasks and activities of support engineers and sales professionals in enterprise environment. The application of the OODA loop framework to model the typical workflows of these professionals will be investigated. The research will identify the types of tasks and activities that are most suitable for training AI models to either fully automate through self-serve AI-powered support or to augment human professionals through AI assistance. The project will assess whether the OODA loop can be used to model support engineer scenarios, with the goal of replacing human support engineers with self-serve AI or enabling human-AI collaboration through AI-augmented support. Finally, the study seeks to provide

[1] https://libguides.bournemouth.ac.uk/study-skills-referencing-plagiarism

Department of Computing and Informatics

2023-24 Academic Year Individual Masters Project

insights on how generative AI can be effectively integrated into enterprise workflows to improve efficiency, scalability, and customer experience in support and sales contexts.

1.3 Background - please provide brief background information, e.g., client, problem domain, and make reference to the literature (minimum 4-5 sources):

The business industry has gone through a transformation due to generative artificial intelligence (AI), which offers advantages including increased accessibility, increased efficiency, and decreased costs (Chen et al. 2023). Enterprise operations depend heavily on support engineers and sales staff to motivate revenue growth, address challenges, and assist with customer onboarding. But frequently, their tasks are too much for them, which causes delays, exhaustion, and poor customer service. These difficulties might be mitigated by integrating generative AI models to support, enhance, or automate parts of their jobs. However, effective AI adoption requires a deep understanding of existing workflows, decision-making frameworks, and identifying appropriate jobs for automation or augmentation.

ABPMS (AI-Augmented Business Process Management Systems) aim to improve how business processes are handled, focusing on adaptability, proactiveness, explainability, and context sensitivity. By carrying out procedures on its own, adjusting to environmental changes, and enhancing target achievement, ABPMS promotes efficient teamwork and decision-making, which results in simplifying the user-agent communication and provides helpful guidance (Dumas et al., 2023).

The OODA (Observe, Orient, Decide, Act) loop, created by military strategist John R. Boyd in 1995, is a well-recognised decision-making framework that can be used to model corporate processes. Enck (2012) state that the concept of OODA loop emphasises the importance of being agile and adapting quickly to changing situations. It highlights the significance of being able to make efficient decisions by quickly adjusting to the evolving circumstances.

According to Endsley (1995), effective decision making relies on the ability to assess the current state of the environment, evaluate the available options, and anticipate the consequences of different actions. Situation awareness provides the foundation for this process.

1.4 Research Questions

1. In what ways can we leverage the OODA Loop framework to augment, assist, or automate the goals and activities of Support Engineers (or Sales Professionals)?
   a) How well do the stages of OODA Loop Framework, map to the activities and tasks of a Support Engineer (or Sales Professional)?

2) What are the types of tasks and activities that will be most relevant in training an AI model to assist, augment, or automate tasks/activities of Support Engineers (or Sales Professionals)?
   a) What tasks and activities would not be relevant/suitable?

3) Can we leverage the OODA Loop (or similar) framework to model Support Engineer scenarios, with a view to:

a) Replacing the need for support engineers (Self-Serve)
b) Augmenting Support Engineers (AI-Supported Support Engineers)

1.5 Aims and objectives – what are the aims and objectives of your project? should be <u>specific</u> and <u>measurable</u>:

The project's primary goal is to assess the relevance of the OODA loop framework in describing the workflows of support engineers and sales professionals in corporate settings and to explore the seamless integration of generative AI models into enterprise workflows to either automate or enhance tasks undertaken by these professionals. Objectives include:

1. Systematically map the stages of the OODA loop framework to specific activities and tasks performed by support engineers and sales professionals.
2. Identify tasks best suited for training generative AI models within support and sales workflows.
3. Develop models or scenarios showcasing full automation or augmentation through AI in support and sales tasks.
4. Evaluate potential improvements in efficiency, scalability, and user experience that could come from using generative AI models.
5. Provide feasible options for integrating generative AI models into sales and support processes for businesses.

Section 2: Artefact

2.1 What is the artefact that you intend to produce?

The primary artefact to be produced from the project demands a set of models or scenarios demonstrating how generative AI can be integrated into enterprise workflows for support engineers and sales professionals, leveraging the OODA loop framework. This collection of artefacts includes: a mapping of the OODA loop stages to the routine activities and tasks performed by support engineers and sales professionals, which can be presented visually or as detailed documentation. The identification and documentation of tasks and activities most suitable for training generative AI models, accompanied by rationale and examples. Scenarios illustrating self-serve AI-powered support potentially replacing human support engineers or AI-augmented support where AI collaborates with human professionals, all developed using the OODA loop or similar framework. A quantitative and/or qualitative analysis assessing potential improvements enabled by the AI-automated and AI-augmented models in efficiency, scalability, and customer experience. Lastly, a detailed implementation strategy or roadmap outlining the effective integration of generative AI models into existing enterprise support and sales workflows. Together, these artefacts provide a comprehensive framework for enterprises to adopt generative AI solutions, enhancing the support and sales operations.

Bournemouth University

Department of Computing and Informatics

2023-24 Academic Year Individual Masters Project

2.2 How is your artefact actionable (i.e., routes to implementation and exploitation in the technology domain)?

The project offers enterprises a complete technological solution by improving support engineer and sales professional processes through the use of generative AI. The research provides practical use cases by mapping the steps of the OODA loop, suggesting tasks for AI training, and demonstrating AI integration in scenarios of self-serve assistance and human-AI collaboration. AI investment is supported by a review of potential efficiency and customer experience advances, and actionable integration is ensured by a comprehensive implementation plan. By combining theory and practice, this framework will help enterprises take advantage of generative AI developments to improve business processes and competitive advantages.

Section 3: Evaluation

3.1 How are you going to evaluate your project artefact?

The evaluation process will include mapping of the OODA loop stages to support engineer and sales professional activities, as well as the identification of tasks suitable for generative AI model training, will be evaluated through critical analysis based on literature reviews, case studies and input from academic resources. The developed models and scenarios for self-serve AI support and AI-augmented support will experience assessing their potential strengths, limitations and alignment with established principles and best practices in the field. User studies with target users like support staff and customers will evaluate usability aspects such as perceived utility, trust, and satisfaction. Quantitative evaluation will involve analytical simulations and modelling to estimate potential improvements in task performance metrics like completion time, accuracy and error rates when using the proposed AI-powered approaches compared to traditional methods. The evaluation will provide research-based insights into the feasibility, applicability, and predicted impact of integrating generative AI solutions into support and sales operations within enterprise.

3.2 How does this project relate to your MSc Programme and your degree title outcomes?

It involves the application of artificial intelligence techniques to solve real-world business problems. This research project requires an understanding of state-of-the art AI models, their capabilities, and their training requirements, which are related to the AI course covered in the program. Moreover, the project includes data analysis and modelling to map existing business processes to AI solutions, linking the data science aspects of extracting insights from data. This project will allow me to integrate the application of the theoretical knowledge I've gained studying my MSc program.

3.3 What are the risks in this project and how are you going to manage them?

Application of AI in support and sales and training requires good data, but this can be managed by checking data quality early, using multiple sources, and cleaning the data. AI models might not work

Bournemouth University

Department of Computing and Informatics

2023-24 Academic Year Individual Masters Project

perfectly at first, but they can be improved by using existing knowledge (transfer learning), trying different designs, optimising parameters, and gathering more data. There's a risk of adopting the usage of AI-powered automation at first from support engineers/sales staff. To address this, communicate clearly, train them on the benefits, and get everyone involved early. Finally, it's important to use AI ethically. This means making sure the AI is fair, understandable, overseen by humans, and protects people's privacy. By planning for these challenges and taking steps to solve them, enterprises can successfully bring AI into support and sales while making sure it's used responsibly.

Section 4: References

4.1 Please provide references if you have used any.

Boyd, J., 1995. The Essence of Winning and Losing [online]. Available from: https://fasttransients.files.wordpress.com/2010/03/essence_of_winning_losing.pdf.
Chen, B., Wu, Z. and Zhao, R., 2023. From fiction to fact: the growing role of generative AI in business and finance. Journal of Chinese Economic and Business Studies, 21 (4), 1–26.
Dumas, M., Fournier, F., Limonad, L., Marrella, A., Montali, M., Rehse, J.-R., Accorsi, R., Calvanese, D., De Giacomo, G., Fahland, D., Gal, A., Rosa, M. L., Völzer, H. and Weber, I., 2023. AI-Augmented Business Process Management Systems: A Research Manifesto. ACM Transactions on Management Information Systems, 14 (1).
Enck, R. E., 2012. The OODA Loop. Home Health Care Management & Practice, 24 (3), 123–124.
Endsley, M. R., 1995. Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors: the Journal of the Human Factors and Ergonomics Society [online], 37 (1), 32–64. Available from: https://www.researchgate.net/publication/210198492_Endsley_MR_Toward_a_Theory_of_Situation_Awareness_in_Dynamic_Systems_Human_Factors_Journal_371_32-64.

Section 5: Academic Practice and Ethics

Please delete as appropriate.

5.1 Have you made yourself familiar with, and understand, the University guidance on referencing and plagiarism?	**Yes**
5.2 Do you acknowledge that this project proposal is your own work and that it does not contravene any academic offence as specified in the University's regulations?	**Yes**

Bournemouth
University

Department of Computing and Informatics

2023-24 Academic Year Individual Masters Project

Note: Please complete the research ethics checklist once the proposal has been approved by your supervisor.

Section 6: Proposed Plan (please attach your Gantt chart below)

Project duration: *13/05/2024 - 24/10/2024*

weeks

ACTIVITY	START	DURATION
Literature Review & Planning	1	3
Identify Relevant Sources	1	1
Reviewing OODA Loop Literature	2	1
Reviewing Generative AI Literature	3	1
First Progress Review	4	1
OODA Loop Mapping	5	2
Analyse support/sales workflows	5	1
Map to OODA loop framework	6	1
Generative AI Tasks Identification	7	2
Identify Trainable Tasks (Support)	7	1
Identify Trainable Tasks (Sales)	8	1
Scenario Development	9	2
Design AI-Augmented support	9	1
Design self-serve AI support	10	1
Second Progress Review	10	2
Dissertation submission	14	1
Video presentation	14	1
Q&A session	15	2

APPENDIX E: RESEARCH ETHICS CHECKLIST

Bournemouth University

Research Ethics Checklist

About Your Checklist	
Ethics ID	57529
Date Created	13/04/2024 15:07:12
Status	Approved
Date Approved	16/04/2024 15:26:15
Risk	Low

Researcher Details	
Name	Yasemin Karaca
Faculty	Faculty of Science & Technology
Status	Postgraduate Taught (Masters, MA, MSc, MBA, LLM)
Course	MSc Data Science & Artificial Intelligence

Project Details	
Title	Generative AI in the Enterprise as a co-worker for tech support
Start Date of Project	13/05/2024
End Date of Project	27/08/2024
Proposed Start Date of Data Collection	03/06/2024
Supervisor	Huseyin Dogan
Approver	Huseyin Dogan

Summary - no more than 600 words (including detail on background methodology, sample, outcomes, etc.)

This project aims to explore how generative AI models can be used to assist, augment, or automate the tasks and activities of support engineers and sales professionals in an enterprise setting. The project will investigate the application of the OODA (Observe, Orient, Decide, Act) loop framework to model the typical workflows of these professionals.

The OODA loop is a known decision-making framework that can be used to model business processes, emphasising the importance of being agile and adapting quickly to changing situations. By applying the OODA loop to support engineer and sales professional scenarios, the project seeks to identify the types of tasks and activities that are most suitable for training AI models to either fully automate through self-serve AI-powered support or augment human professionals through AI assistance.

The key objectives of the project are to systematically map the stages of the OODA loop framework to the specific activities and tasks performed by support engineers and sales professionals, identify the tasks best suited for training generative AI models within support and sales workflows, develop models or scenarios highlighting full automation or augmentation through AI in support and sales tasks, evaluate the potential improvements in efficiency, scalability, and user experience that could come from using generative AI models, and provide achievable options for integrating generative AI models into sales and support processes for enterprises.

The research will involve a combination of literature review, case studies, and input from academic resources to assess the relevance of the OODA loop framework in describing the workflows of support engineers and sales professionals. User studies with target users, such as support staff and customers, will be conducted to evaluate the usability and perceived utility of the proposed AI-powered approaches.

Additionally, analysis and modelling will be used to estimate potential improvements in task performance metrics, such as completion time, accuracy, and error rates, when using the proposed AI-powered approaches compared to traditional methods. The possible effects of AI-powered automation and augmentation on support engineers and sales professionals are the project's main ethical concern. These experts run the danger of opposing the adoption of AI-powered solutions because they see them as a threat to their jobs. To address this, the project will place a strong emphasis on the value of professional involvement, training, and clear communication when using AI technologies. The possibility for AI models to be biased or to make decisions that might have a negative impact on users or customers raises further ethical concerns. The effort will make sure that the AI models are developed and trained with an emphasis on transparency, accountability, and fairness to mitigate this.

Filter Question: Is your study solely literature based?

Additional Details	
Will you have access to personal data that allows you to identify individuals which is not already in the public domain?	No
Will you have access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or separate confidentiality agreement)?	No

Storage, Access and Disposal of Research Data	
Where will your research data be stored and who will have access during and after the study has finished.	
The project won't collect or store research data from participants. Since it is a theoretical research project focused on exploring the application of the OODA loop framework and generative AI models, there won't be any data storage or access from anyone. The research is centered around literature review, case studies, input from academic resources, analysis and modelling.	
Once your project completes, will your dataset be added to an appropriate research data repository such as BORDaR, BU's Data Repository?	No
Please explain why you do not intend to deposit your research data on BORDaR? E.g. do you intend to deposit your research data in another data repository (discipline or funder specific)? If so, please provide details.	
The project won't collect or store research data that would need to be deposited in a data repository.	