Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Self- Healing Problems Systems to Support IT Helpdesk

Nadir K. Salih [1]

[1] *Computer Enigeening, College of Engineering, University of Burami. Al Buraimi, Oman Email: nadir@uob.edu.om*

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

Introduction: focuses on the development and implementation of a self-healing problem system aimed at revolutionizing IT helpdesk operations. The system leverages Case-Based Reasoning (CBR) combined with machine learning to autonomously diagnose and resolve recurring IT issues.

Objectives: The research highlights the inefficiencies of traditional helpdesk frameworks, emphasizing the need for automation to handle complex and high-volume IT queries effectively. By incorporating predictive algorithms, rule-based solutions, and real-time problem-solving mechanisms, the system aims to reduce downtime, improve user satisfaction, and enhance operational efficiency.

Methods: Experimental results demonstrate the efficacy of five similarity functions—Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared through the case base reasoing.

Results: in identifying and resolving IT issues across three categories: managers, employees, and students. The Manhattan function consistently achieved the highest accuracy, with 89.9% for manager cases, 65.6% for employees, and 54.6% for students. Error rates calculated using the Root Mean Square Error (RMSE) revealed similar trends, with the Manhattan function demonstrating strong reliability across all categories. For instance, the error rates for Manhattan were 27.76 for managers, 21.01 for employees, and 16.34 for students. Conversely, other functions like Canberra and Squared Chord exhibited limited effectiveness, particularly for complex or diverse cases

Conclusions: These results affirm the system's ability to adapt to varying data complexities, making it a robust solution for modern IT challenges. Future research should focus on enhancing these systems' scalability and exploring advanced analytics for broader applications in dynamic IT environments.

Keywords: Self-Healing Systems, Case-Based Reasoning (CBR), Machine Learning in IT, Helpdesk, Similarity Functions, Predictive Algorithms.

INTRODUCTION

The increasing complexity of IT systems and the rising expectations for uninterrupted service delivery have paved the way for innovative solutions in IT management [1][2]. One such advancement is the adoption of self-healing systems. These systems are designed to autonomously detect, diagnose, and resolve issues within IT infrastructure without requiring human intervention [3][4]. A self-healing system leverages automation, machine learning, and artificial intelligence (AI) to identify potential problems, predict failures, and execute corrective actions in real time. This proactive approach minimizes downtime, enhances service reliability, and reduces the workload on IT helpdesk teams. In the context of an IT helpdesk, self-healing systems play a crucial role by addressing routine issues like network connectivity problems, software crashes, or misconfigurations before they escalate [5]. This not only improves efficiency but also allows IT personnel to focus on more complex tasks that require human expertise. The implementation of self-healing systems involves integrating advanced monitoring tools, establishing automated workflows, and employing AI-driven algorithms [6]. These components work together to ensure seamless operations and continuous availability of IT services, ultimately enhancing user satisfaction and operational resilience [7].

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

This research explores the architecture, benefits, challenges, and real-world applications of self-healing systems, highlighting their transformative impact on IT helpdesk operations [8]. As part of this study, we implemented a model for the system at the University of Buraimi, dividing it into three main sections: the first for managers, the second for staff, and the third for students. Each section addresses a set of issues related to the university's services. The Eclipse platform was used to input data, specifically focusing on the problems and their corresponding solutions. Furthermore, we created detailed tables for each section, outlining the multiple attributes associated with all identified issues.

PROJECT BLOCK DIAGRAM

Figure 1 illustrates the categorization of IT helpdesk problems within Al Buraimi University into three main groups: managers, employees, and students. These issues are addressed using a Case-Based Reasoning (CBR) approach to self- healing. The CBR process consists of four key phases: Retrieve, Reuse, Revise, and Retain. Each problem identified is processed through these phases to ensure efficient resolution and continuous learning within the system.

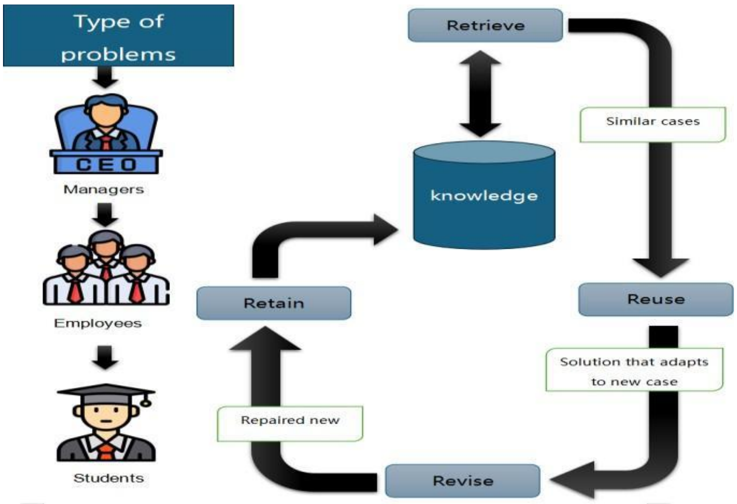

Figure 1 Project Block Diagram

1. Retrieve: In this phase, the system searches the CBR database to identify cases that are like the current problem. In this process we will mention one of the problems, which is the System Access Issue (SAI), the system retrieves past cases of similar access problems to analyze the solutions that worked previously. This step ensures efficiency by avoiding the need to resolve problems from scratch [9].
2. Reuse: Once the relevant cases are retrieved, their solutions are adapted to the current problem. For instance, if employees encounter recurring System Access Issues (SAI), the system applies solutions from past cases with adjustments as needed to fit the specific context. This step highlights the system's flexibility in handling variations of similar problems.
3. Revise: During this phase, the proposed solution is tested and refined to ensure it resolves the problem effectively. if a student reports an issue with the System Access Issue (SAI), the system tests the solution and collects feedback to make improvements. This ensures that the resolution is both accurate and applicable in real-world scenarios.
4. Retain: After successfully solving the problem, the refined solution is stored back in the CBR database for future reference. This phase builds the system's knowledge base, enabling it to improve over time. For example, a resolved issue related to the System Access Issue (SAI) to help address similar problems in the future more effectively.

SOFTWARE ASPECT AND PROJECT CONFIGURATION

In the context of the Self-Healing Problem Systems to Support IT Helpdesk project, Eclipse serves as the primary integrated development environment (IDE) for the software development process. Eclipse offers powerful tools and

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

frameworks for building, debugging, and deploying Java-based applications, making it an ideal choice for implementing the system's backend, AI/ML components, and database interactions. Software Aspect of the Project The software aspect encompasses the technologies, tools, frameworks, and methodologies used to develop the system. It also outlines how these components interact to create a self-healing system that can automatically resolve IT helpdesk issues using Case-Based Reasoning (CBR) and AI-driven algorithms [10].

Core Technologies for the System The system leverages advanced core technologies to implement a dynamic, self-healing IT problem-solving mechanism, enhancing the efficiency of helpdesk operations. At its core is Case-Based Reasoning (CBR), which forms the foundation of the system's problem-solving capabilities by storing past IT issues and their resolutions [11]. This allows the system to retrieve and adapt previous cases to address new challenges effectively. The CBR engine is developed in Java using Eclipse, with a case database (e.g., MySQL or SQLite) connected via JDBC (Java Database Connectivity). Additionally, the system integrates Machine Learning (ML) and Artificial Intelligence (AI) through Eclipse-compatible libraries such as TensorFlow, Apache Mahout, or Weka. These components enable advanced functionalities, including problem classification, predictive analytics, and proactive issue identification. For instance, predictive models can analyze system usage data to forecast hardware failures, allowing for preemptive maintenance. Together, these technologies create a robust and intelligent solution for IT helpdesk operations [12].

The self-healing system is equipped with functionalities designed to detect, diagnose, and resolve IT issues autonomously, ensuring seamless operations. Developed using the Eclipse IDE and Java, the system leverages automated monitoring tools to identify problems such as network outages, hardware failures, or software crashes. Once detected, issues are classified based on severity and routed through the Case-Based Reasoning (CBR) engine for resolution [13]. The CBR engine plays a pivotal role by retrieving similar past problems from its database and suggesting appropriate solutions. To enhance accuracy, the system incorporates an AI-based classifier to assess the severity and nature of the issues [14] . By integrating machine learning libraries into Eclipse, the classification and resolution process continuously improves, adapting to feedback and user interactions. This intelligent workflow ensures timely and effective problem-solving for IT environments.

Design of Experiments

The application is structured into three hierarchical sections: Provider, Tenant, and User [15]. The Provider represents the highest category, followed by the Tenant, which occupies the middle tier, and finally, the User at the base level. To contextualize this structure within Al Buraimi University, we have linked these categories to specific roles: the Manager corresponds to the Provider, the Employee represents the Tenant, and the student aligns with the User.

Figure 2 Three levels of university

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

The application's hierarchical structure, consisting of Provider, Tenant, and User levels, allows for efficient management and resolution of IT helpdesk issues across different user tiers [16]. From the IT helpdesk perspective, this structure enables the system to tailor solutions based on the specific roles and permissions of each level. For instance, the IT support team can provide high-level administrative access and control over system configurations and troubleshooting tools for the Provider (Manager) role, allowing them to oversee and manage large-scale infrastructure problems. The Tenant (Employee) level can receive more specific support related to department-wide issues or user- specific problems, ensuring targeted assistance is provided. Finally, the User (Student) level is designed for individual support, where students can access helpdesk services for personal or localized issues, such as account access or software troubleshooting [17][18]. This tiered approach ensures that each level of the application is equipped with appropriate resources and troubleshooting capabilities, streamlining the IT support process and improving the efficiency of problem resolution across Al Buraimi University [19].

Specification and Requirements

The program's requirements were organized in the form of a table that outlines key information for each user category: managers, employees, and students. Each table includes at least ten cases, along with their attributes and the resulting values for each scenario. Additionally, the tables refer to the number of tables identify common problems and their corresponding solutions. This structure enables new users to easily diagnose issues based on the displayed values and identify the most appropriate solutions. At the end of each table, a summary of shortcuts for the identified problems and solutions is provided for quick reference.

Table 1 Cases information of Manager

Case ID	Type	Level	Feasibility	Scalability	Complexity	Training Needs	Financial Impact	Solution
C1	MSP	High	Feasible	Moderate	Simple	Yes	Moderate	Automated Reporting Tools (ART)
C2	SII	Critical	Challenging	High	Simple	No	High	Automated Compatibility Testing (ACT)
C3	STH	Urgent	Moderate	Scalable	Complex	No	Low	AI-Driven Security (ADS)
C4	DMG	Medium	Easy	Scalable	Moderate	Yes	Low	Self-Healing Databases (SHD)
C5	IAV	Moderat e	Feasible	Moderate	Complex	No	Low	Automated Asset Management (AAM)
C6	**UST**	Low	**Easy**	**High**	**Complex**	**Yes**	**Low**	Automated Usage Monitoring (AUM)
C7	**RST**	Medium	**Moderate**	**Scalable**	**Simple**	**No**	**Low**	Ticket Management Consolidation (TMC)
C8	**CDR**	High	**Feasible**	**Moderate**	**Moderate**	**No**	**Low**	Automated Compliance Checks (ACC)

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

C9	**SDA**	Critical	**Challenging**	**High**	**Complex**	**Yes**	**High**	Proactive Alert System (PAS)

Abbreviation References

MSP: Problem monitoring staff productivity and response times. **ART**: Automated reporting tools for tracking IT staff performance. **SII**: System integration problems between legacy and new systems. **ACT**: Automated testing to ensure system compatibility.

STH: Security threats such as phishing and unauthorized access.

ADS: AI-based security systems that detect suspicious activities.

DMG: Challenges with managing large volumes of data efficiently.

SHD: Self-healing databases that optimize and organize data.

IAV: Difficulty tracking IT assets across the organization.

AAM: Automated tools for managing and tracking IT assets.

UST: Underutilized software that still incurs licensing costs.

AUM: Automated monitoring to track and report software usage.

RST: Redundant support tickets that create inefficiencies.

TMC: Tools to consolidate and prioritize support tickets.

CDR: Ensuring compliance with regulations like GDPR and FERPA.

ACC: Automated systems that check compliance with data regulations.

SDA: Missed or delayed alerts for critical system downtimes.

PAS: Proactive alert systems that notify managers of downtimes.

Table 2 Cases Information of Employee

Cases	Type of problem	Severity	Knowledge Required	Availability	History Issue	Level	Solution
C1	SAI	minor	none	available	yes	Personal	Auto Password Reset (APR)
C2	SND	critical	advanced	not available	no	Collective	Network Monitoring Fallback (NMF)
C3	SCI	major	basic	not available	yes	Collective	Automated Patch Management (APM)
C4	FDL	critical	advanced	not available	no	Collective	Automated Backup Recovery (ABR)
C5	PPI	major	basic	available	yes	Collective	Auto Printer Diagnostics (APD)
C6	EOI	critical	basic	not available	yes	Collective	Auto Email Prioritization (AEP)
C7	ESL	major	advanced	available	yes	Personal	Auto Performance Resource (APR)
C8	DEE	minor	advanced	available	yes	Personal	Automated Validation System (AVS)

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

C9	RAP	major	basic	available	yes	Collective	Auto VPN Setup (AVP)
C10	ITI	major	basic	available	yes	Personal	Real-Time Guidance (RTG)
C11	PMI	major	basic	available	yes	Collective	Auto Projector Fix (APF)
C12	SBC	major	basic	available	yes	Collective	Self-Healing Connectivity (SHC)

Abbreviation References

SAI: Refers to difficulties with logging into IT systems due to password or account issues.

APR: An automated password reset tool to quickly resolve login problems.

SND: Describes slow network or system downtime issues affecting staff productivity.

NMF: Tools that monitor networks and switch to fallback systems during downtimes.

SCI: Software incompatibilities or outdated programs causing disruptions.

APM: Automatically updates and manages patches for all software.

FDL: Describes accidental loss of files or important data.

ABR: Backup and recovery tools that restore lost or deleted files.

PPI: Issues with printers or peripherals like scanners.

APD: Automated diagnostics tools that fix printer or connection problems.

EOI: Overwhelming amounts of email or missing critical communications.

AEP: Automatically prioritizes important emails and filters less relevant ones.

ESL: Slow or unresponsive ERP systems during peak times.

APR: Monitors system performance and adjusts resource usage for ERP systems.

DEE: Human errors during data entry causing inaccurate information.

AVS: Automated systems that validate and flag incorrect data entries.

RAP: Problems with accessing systems or files while working remotely.

AVP: VPN tools that automate secure remote access setup.

ITI: Lack of sufficient training for staff on new software or systems.

RTG: Real-time guidance systems providing in-app help and tutorials.

PMI: Projector issues like overheating or poor image quality.

APF: Automatic tools that diagnose and fix projector problems.

SBC: Connectivity issues with smart boards or touch displays.

SHC: Self-healing tools that automatically resolve smart board connection problems.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 3 Cases Information of Student

Cases	Type	Category	Level	Frequency	Time Resolution	Affected System	Solution
C1	WFI	Connectivity	High	Occasional	Moderate	Network	Auto Wi-Fi Diagnostics (AWD)
C2	LMS	Access	High	Rare	Moderate	Software	Self-Healing Login (SHL)
C3	DCI	Access	High	Rare	Quick	Hardware	Device Compatibility System (DCS)
C4	ECI	Connectivity, Access	High	Occasional	Quick	Software, Network	Email Monitoring Alerts (EMA)
C5	OEI	Connectivity	High	Frequent	Quick	Network	Auto-Save Recovery (ASR)
C6	SLI	Access, Security	Low	Rare	Lengthy	Software, Hardware	Automated License Management (ALM)
C7	UED	Access	Medium	Rare	Moderate	Software, Hardware	Responsive Design System (RDS)
C8	CSS	Connectivity, Access	High	Rare	Quick	Software	Auto-Sync Troubleshoot (AST)
C9	DAS	Access	Medium	Rare	Quick	Software, Network	AI Chatbot (AIC)
C10	UAS	Connectivity, Access	High	Rare	Lengthy	Software, Network	Multi-Factor Authentication (MFA)

Abbreviation References

WFI: Refers to Wi-Fi connectivity issues that students commonly face on campus.

AWD: An automated solution that identifies and fixes Wi-Fi problems.

LMS: Refers to problems accessing learning management systems (e.g., Moodle, Blackboard).

SHL: A system that automatically fixes LMS login problems.

DCI: Problems with devices not being compatible for tasks like exams or registrations.

DCS: A compatibility-check system that resolves device issues.

ECI: Refers to issues with university-provided email services.

EMA: Automatically monitors and fixes email configuration problems.

OEI: Problems students face during online exams, like connectivity drops.

ASR: Automatically saves progress and allows recovery during online exams.

SLI: Software access issues due to expired or limited licenses.

ALM: Automated software license management and distribution.

UED: User experience inconsistencies across devices, such as desktops and mobiles.

RDS: Ensures consistent user interfaces across devices through responsive design.

CSS: Problems syncing files between cloud storage and personal devices.

AST: Automatically resolves syncing issues in cloud storage systems.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

DAS: Difficulty for students to get quick IT support when needed.

AIC: AI-driven chatbot that provides instant help and escalates complex issues.

UAS: Unauthorized access to student accounts, often due to hacking.

MFA: Multi-factor authentication for extra security, plus suspicious login alerts.

Practical Implementation and Coding

In practice, the system will consist of three key elements: attributes, value cases, and mappers. These elements will be defined for each of the three hierarchical levels: the Provider level, representing the highest level and corresponding to managers; the Tenant level, representing the intermediate level and corresponding to employees; and the User level, representing the base level and corresponding to students. Additionally, the input for this system will be implemented using the Eclipse program.

In the practical implementation of the self-healing problem system, the key elements—attributes, value cases, and mappers—play a vital role in structuring and managing IT helpdesk queries across the hierarchical levels. Here's an explanation of each element:

1. Attributes: Attributes represent the defining characteristics of the problems encountered at each hierarchical level. For each level—Provider, Tenant, and User—attributes are tailored to capture the relevant features of issues: Provider Level (Manager): Attributes could include system-wide performance metrics, network health, resource allocation, and security configurations. These attributes help in identifying large-scale IT issues that affect the overall infrastructure.

   Tenant Level (Employee): Attributes might include departmental software configurations, internal network connectivity, or employee access permissions. These focus on issues affecting groups or teams within the organization. User Level (Student): Attributes would be more focused on individual concerns, such as login problems, application crashes, or specific software errors that are unique to the student's device or account.

2. Value Cases: Value cases represent the potential scenarios or specific instances that each attribute can take on. These are the possible values for each attribute that guide the troubleshooting or resolution process:

   Provider Level: Examples of value cases might include high CPU usage, network outages, or system resource limitations. These cases could trigger broad corrective actions or alerts at the system administration level.

   Tenant Level: Common value cases could include issues like software compatibility errors, departmental access issues, or network misconfigurations, which are typically resolved by the IT team within specific departments.

   User Level: Value cases could consist of issues like incorrect login credentials, software installation failures, or personal device malfunctions, which are resolved through user-focused support or automated troubleshooting.

3. Mappers: Mappers are responsible for linking the attributes and value cases to the relevant solutions or actions. They serve as the decision-making mechanism in the system, determining the appropriate response based on the problem's characteristics at each level:

   Provider Level: Mappers here might associate system-wide issues like server crashes with escalated support or automated self-healing actions (e.g., server reboot, resource reallocation).

   Tenant Level: At this level, mappers can match problems like employee access errors with administrative solutions, such as resetting credentials, restoring access permissions, or reconfiguring internal systems.

   User Level: For individual issues, mappers will associate problems like login failures or software bugs with automated responses such as password reset procedures, application updates, or system diagnostics.

   Together, these elements allow the system to process IT helpdesk queries at each hierarchical level, ensuring that problems are diagnosed and resolved efficiently based on their complexity and scope. The input for this system, implemented using Eclipse, will allow seamless handling of these elements and trigger appropriate actions based on the identified attributes and value cases.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Provider Level (Manager):

1. Attributes:

Type of problem, Urgency Level, Implementation Feasibility, Scalability, System Complexity, Staff Training Needs, Financial Impact, Solution

2. Value Cases:

MSP,High,Feasible,Moderate,Simple,Yes,Moderate,Automated Reporting Tools
SII,Critical,Challenging,High,Simple,No,High,Automated Compatibility Testing
STH,Urgent,Moderate,Scalable,Complex,No,Low,AI-Driven Security
DMG,Medium,Easy,Scalable,Moderate,Yes,Low,Self-Healing Databases
IAV,Moderate,Feasible,Moderate,Complex,No,Low,Automated Asset Management
UST,Low,Easy,High,Complex,Yes,Low,Automated Usage Monitoring
RST,Medium,Moderate,Scalable,Simple,No,Low,Ticket Management Consolidation
CDR,High,Feasible,Moderate,Moderate,No,Low,Automated Compliance Checks
SDA,Critical,Challenging,High,Complex,Yes,High,Proactive Alert System

3. Mapper:

categorical,MSP:1,High:2,Feasible:3,Moderate:4,Simple:5,Yes:6,Moderate:4,default:4
categorical,SII:8,Critical:9,Challenging:10,High:2,Simple:5,No:11,High:2,default:4
categorical,STH:12,Urgent:13,Moderate:4,Scalable:14,Complex:15,No:11,Low:16,default:4
categorical,DMG:17,Medium:18,Easy:19,Scalable:14,Moderate:4,Yes:6,Low:16,default:4
categorical,IAV:20,Moderate:4,Feasible:3,Moderate:4,Complex:15,No:11,Low:16,default:4
categorical,UST:21,Low:16,Easy:19,High:2,Complex:15,Yes:6,Low:16,default:4
categorical,RST:22,Medium:18,Moderate:4,Scalable:14,Simple:5,No:11,Low:16,default:4

Tenant Level (Employee):

1. Attributes:

Type of problem,Severity,Technical Knowledge Required,Documentation Availability,History of Issue,Escalation Level,Solution

2. Value of Cases:

SAI,minor,none,available,yes,personal,Auto Password Reset SND,critical,advanced,not available,no,collective,Network Monitoring Fallback SCI,major,basic,not available,yes,collective,Automated Patch Management FDL,critical,advanced,not available,no,collective,Automated Backup Recovery PPI,major,basic,available,yes,collective,Auto Printer Diagnostics EOI,critical,basic,not available,yes,collective,Auto Email Prioritization ESL,major,advanced,available,yes,personal,Auto Performance Resource DEE,minor,advanced,available,yes,personal,Automated Validation System RAP,major,basic,available,yes,collective,Auto VPN Setup ITI,major,basic,available,yes,personal,Real-Time Guidance PMI,major,basic,available,yes,collective,Auto Projector Fix SBC,major,basic,available,yes,collective,Self-Healing Connectivity

3. Mapper:

categorical,SAI:1,minor:2,none:3,available:4,yes:5,personal:6,default:4 categorical,SND:7,critical:8,advanced:9,not available:10,no:11,collective:12,default:4 categorical,SCI:13,major:14,basic:15,not available:10,yes:5,collective:12,default:4 categorical,FDL:16,critical:8,advanced:9,not available:10,no:11,collective:12,default:4
categorical,PPI:17,major:14,basic:15,available:4,yes:5,collective:12,default:4
categorical,EOI:18,critical:8,basic:15,not available:10,yes:5,collective:12,default:4

User Level (Student):

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1. Attributes:

Type of problem,Category,Impact Level,Frequency,Resolution Time,Affected System,solution

2. Value of Cases:

WFI,connectivity,high,occasional,moderate,network,Auto Wi-Fi Diagnostics
lms,access,high,rare,moderate,software,Self-Healing Login dci,access,high,rare,quick,hardware,Device Compatibility System
dci,connectivity and access,high,occasional,quick,software and network,Email Monitoring Alerts
oei,connctivity,high,frequent,quick,network,Auto-Save Recovery
sli,access and security,low,rare,lengthy,software and hardware,Automated License Management
ued,access,medium,rare,moderate,software and hardware,Responsive Design System css,connectivity and access,high,rare,quick,software,Auto-Sync Troubleshoot das,access,medium,rare,quick,software and network,AI Chatbot
uas,connectivity and access,high,rare,lengthy,software and network,Multi-Factor Authentication

3. Mapper:

categorical,WFI:1,connectivity:2,high:3,occasional:4,moderate:5,network:6,default:4
categorical,lms:7,access:8,high:3,rare:10,moderate:11,software:12,default:4
categorical,dci:13,access:8,high:3,rare:10,quick:14,hardware:15,default:4
categorical,dci:13,connectivity and access:16,high:3,occasional:4,quick:14,software and network:17,default:4
categorical,oei:18,connctivity:2,high:3,frequent:19,quick:14,network:6,default:4
categorical,sli:20,access and security:21,low:22,rare:10,lengthy:23,software and hardware:24,default:4
In terms of coding, we will incorporate the essential scripts required to execute the program and process the input data, including attributes, value cases, and the mapper. Additionally, we will provide a detailed explanation of the components being implemented and their functionality.

RESULTS

After implementing the system, we analyzed the results and presented them in a series of tables. In the first table 4, we focused on case similarities to calculate the error rate. This table was organized based on five functions—Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared—applied across three levels: manager, employee, and student. By leveraging these functions, we computed case similarities and subsequently derived the error rate for each level.

Table 4 Cases similarities to calculate error rate

Function	Chosen	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	
MANHATTAN	10	-	-	3.0	58.0	53.0	128.0	120.0	53.0	53.0	61.0	11.0	6.0	
EUCLIDEAN	10	-	-	1.73	28.02	24.10	54.64	48.99	24.10	24.10	29.75	7.39	4.74	
CANBERRA	10	-	-	0.15	0.89	0.70	1.05	0.72	0.70	0.70	1.04	0.48	0.75	
SQUARED CHORD	10	-	-	0.08	3.72	3.11	9.10	8.41	3.11	3.11	3.80	0.76	1.08	Student
SQUARED CHI-SQUARED	10	-	-	0.15	6.89	5.70	15.62	14.29	5.70	5.70	7.04	1.48	1.95	
MANHATTAN	8	86.0	86.0	38.0	86.0	5.0	5.0	118.0	86.0	80.0	110.0	80.0	8.0	
EUCLIDEAN	8	40.89	40.89	21.29	40.89	2.24	2.24	58.55	40.89	37.26	52.89	37.26	5.24	
CANBERRA	8	1.20	1.20	0.70	1.20	0.12	0.12	1.96	1.20	1.27	1.61	1.27	1.72	Employee
SQUARED CHORD	8	5.71	5.71	3.57	5.71	0.06	0.06	7.75	5.71	4.19	7.06	4.19	1.06	

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

SQUARED CHI-SQUARED	8	10.48	10.48	6.48	10.48	0.12	0.12	14.39	10.48	7.91	13.06	7.91	1.92	
MANHATTAN	2	-	-	-	77.0	88.0	124.0	151.0	139.0	136.0	179.0	10.0	19.0	Manager
EUCLIDEAN	2	-	-	-	36.57	34.58	51.75	61.25	56.35	56.65	76.35	7.08	10.73	
CANBERRA	2	-	-	-	1.17	0.94	1.16	1.38	1.18	1.36	1.74	0.39	1.14	
SQUARED CHORD	2	-	-	-	4.72	4.35	7.04	7.72	7.51	7.24	10.55	1.00	1.55	
SQUARED CHI-SQUARED	2	-	-	-	8.70	7.87	12.76	13.97	13.57	13.16	19.13	1.93	2.87	

In the subsequent three tables, we present the accuracy rates for the five similarity functions—Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared—across the three levels: manager, employee, and student. These tables follow the earlier results that focused on case similarities for calculating the error rate. The table 5 presented and the figure 3 illustrated, the results demonstrate the accuracy rate of the similarity function specifically for the role of a manager.

Table 5 Accuracy rate of similarity function for (manager)

Function	Accuracy
Manhattan	89.9 %
Euclidian	43.4 %
Canberra	1.16 %
Squared chord	5.74 %
Squared chi-squared	10.4 %

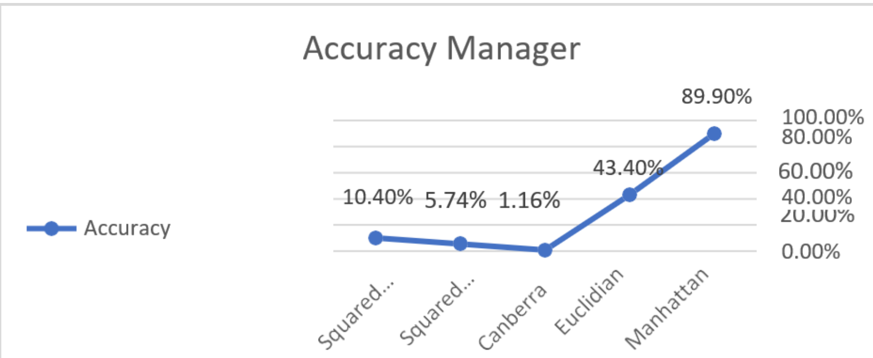

Figure 3 Accuracy for Manager level

In the table 6 presented and the figure 4 illustrated, the results demonstrate the accuracy rate of the similarity function specifically for the role of an employee.

Table 6 Accuracy rate of similarity function for (employee)

Function	Accuracy
Manhattan	65.6 %
Euclidian	31.7 %

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Canberra	1.04 %
Squared chord	3.75 %
Squared chi-squared	7.81 %

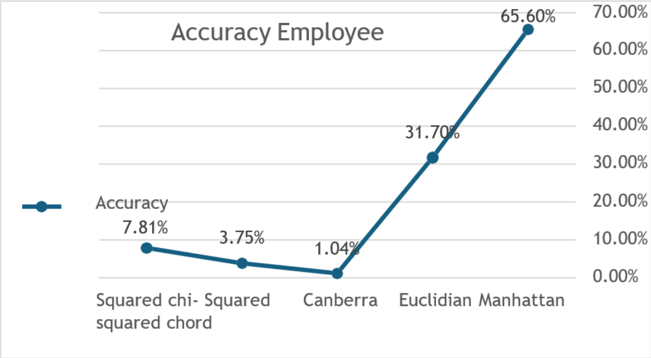

Figure 4 Accuracy for Employee level

In the table 7 presented and the figure 5 illustrated, the results demonstrate the accuracy rate of the similarity function specifically for the role of a student.

Table 7 Accuracy rate of similarity function for (student)

Function	Accuracy
Manhattan	54.6 %
Euclidian	22.3 %
Canberra	0.71 %
Squared chord	3.62 %
Squared chi-squared	6.45 %

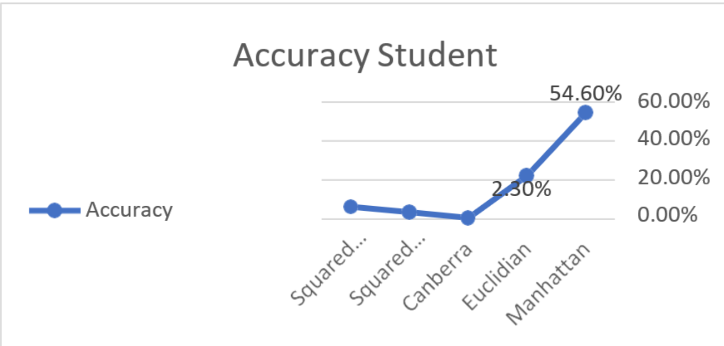

Figure 5 Accuracy for Student level

Additionally, we included three more tables to calculate the error rate for each category—manager cases, employee cases, and student cases—using the RMSE (Root Mean Square Error) formula. These error rates were computed for all five similarity functions: Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared.

The table 8 displayed and the figure 6 illustrated present the results of the error rate calculated using the RMSE formula specifically for the manager cases.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 8 Error rate calculated using RMSE formula

Function	Error rate
Manhattan	27.76
Euclidian	10.91
Canberra	2.31
Squared chord	14.22
Squared chi-squared	25.54

Figure 6 Error rate of manager cases

Table 9 displayed and the figure 7 illustrated present the results of the error rate calculated using the RMSE formula specifically for the employee cases.

Table 9 using RMSE formula

Function	Error rate
Manhattan	21.01
Euclidian	10.26
Canberra	3.47
Squared chord	14.45
Squared chi-squared	26.67

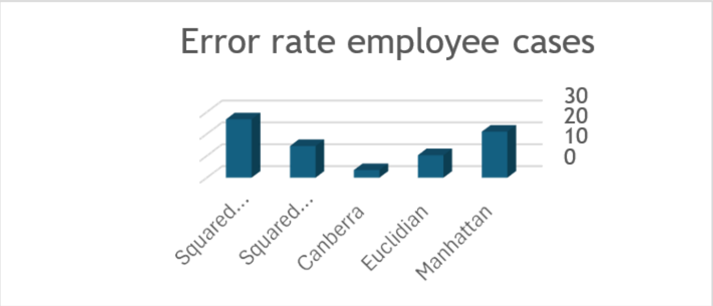

Figure 7 Error rate of employee cases

The table 10 displayed and the figure 8 illustrated present the results of the error rate calculated using the RMSE formula specifically for the student cases.

Table 10 Error rate using RMSE formula

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Function	Error rate
Manhattan	16.34
Euclidian	72.8
Canberra	1.79
Squared chord	11.2
Squared chi-squared	19.92

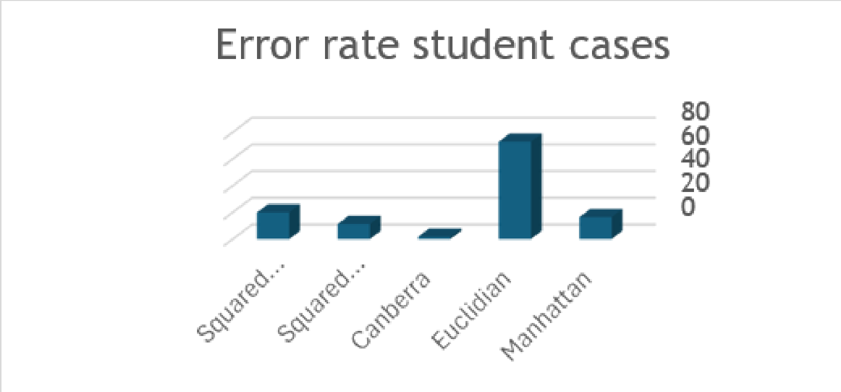

Figure 8 Error rate of student cases

DISCUSSION

Case Similarities for Error Rate Calculation: The results for case similarities were calculated using five prominent similarity functions: Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared. These functions were applied across three specific levels—students, employees, and managers—to measure the effectiveness of case comparisons. The Manhattan and Euclidean functions exhibited notably larger values across most cases, reflecting their sensitivity to variations in the dataset. In contrast, the Canberra and Squared Chord functions produced smaller values, highlighting their more constrained range of measurements. For student cases, the calculations revealed greater fluctuations in similarities, suggesting a higher level of complexity or inconsistency in case characteristics. Employee cases demonstrated more moderate values, indicating relatively stable patterns. Manager cases, on the other hand, showed a narrower range of variability, implying simpler and more structured problem instances. This variability across levels suggests that similar measurements are influenced not only by the functions themselves but also by the nature and complexity of the cases being analyzed.

Accuracy Rate of Similarity Functions: The accuracy of the five similarity functions—Manhattan, Euclidean, Canberra, Squared Chord, and Squared Chi-Squared—was evaluated for managers, employees, and students to determine their reliability in identifying correct similarities. Across all levels, the Manhattan function emerged as the most accurate, achieving an impressive 89.9% accuracy for manager cases, followed by 65.6% for employees and 54.6% for students. This strong performance highlights Manhattan's ability to handle a wide range of case structures effectively. Conversely, the Canberra and Squared Chord functions consistently recorded low accuracy rates, with Canberra showing as little as 1.16% for managers and 0.71% for students, suggesting limited utility in scenarios with significant data variations. Euclidean accuracy demonstrated mixed results, performing reasonably well for manager cases but declining sharply for student cases, where its accuracy was only 22.3%. These findings indicate that the Manhattan function provides the most consistent and reliable results across all categories, while other functions may require adjustments or specific conditions to perform effectively. Notably, the lower accuracy for student cases reflects the inherent complexity of these cases, which may contain less structured or more diverse patterns compared to manager cases.

Error Rate Calculation using RMSE: To further evaluate the performance of the five similarity functions, error rates were calculated using the Root Mean Square Error (RMSE) formula for manager, employee, and student cases.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

For manager cases, the Manhattan function demonstrated strong reliability with an error rate of 27.76, while the Euclidean function achieved the lowest error rate at 10.91. This suggests that for more structured cases, such as those associated with managers, Euclidean performs optimally due to its ability to minimize deviations. However, Squared Chi-Squared produced a higher error rate of 25.54, indicating its reduced effectiveness in capturing case similarities in this category. In employee cases, Manhattan again performed well with an error rate of 21.01, while Euclidean achieved a competitive rate of 10.26, reinforcing its suitability for moderately structured cases. The Canberra function, despite its low accuracy, demonstrated a surprisingly low error rate of 3.47, suggesting its sensitivity to minor deviations in data values. For student cases, the Euclidean function exhibited a significant error rate of 72.8, far exceeding those of the other functions, which suggests its inability to handle the inherent complexity and variability of student data effectively. In contrast, Manhattan maintained a much lower error rate of 16.34, further underscoring its robustness across all case categories. These results emphasize the importance of selecting the right similarity function based on the complexity of the data, with Manhattan consistently proving to be the most reliable choice.

Comparison and Key Insights: The comparative analysis of the five similarity functions across all three levels—managers, employees, and students—reveals several key insights. The Manhattan function consistently outperformed the other methods in terms of accuracy and error rates, demonstrating its effectiveness in handling both structured and unstructured cases. Its reliability stems from its ability to quantify absolute differences in case attributes, making it adaptable across diverse datasets. Euclidean, while effective for manager and employee cases, struggled significantly with student cases, as evidenced by its 72.8 error rate. This discrepancy suggests that Euclidean may be more sensitive to variations or noise in the student dataset. The poor performance of the Canberra and Squared Chord functions, both in terms of accuracy and error rates, indicates their limited applicability for complex or diverse cases. While they performed adequately under specific conditions, their narrow range and high sensitivity to minor data changes reduced their overall effectiveness. Additionally, the results show that student cases introduce the greatest challenges due to their complexity and variability, likely stemming from the diverse nature of student-related problems. These findings highlight the need for adaptive similarity measures and further preprocessing techniques, particularly when dealing with unstructured or highly variable data. Overall, the Manhattan function emerges as the most robust and versatile similarity measure, capable of addressing the varying complexities observed across all three levels.

CONCLUSION

The concept of Self-Healing Problem Systems represents a transformative advancement in supporting IT helpdesk operations, combining intelligent technologies to detect, diagnose, and resolve issues autonomously. By leveraging cutting-edge methodologies such as Case-Based Reasoning (CBR), Machine Learning (ML), and Artificial Intelligence (AI), these systems not only enhance the speed and accuracy of issue resolution but also significantly reduce the dependency on human intervention. Integrated with tools like the Eclipse IDE and database connectivity, self-healing systems enable a dynamic, scalable, and efficient approach to managing IT environments. This project addresses the persistent challenges in IT operations, such as system downtime, repetitive issues, and resource constraints, by offering a proactive and automated solution. The continuous learning capabilities of the system ensure that it evolves with changing IT landscapes, making it an asset for modern organizations striving for operational excellence. Here we will the contributions that the project has and that have a significant impact on Self-Healing Problems Systems to Support IT Helpdesk:

1. Automated Problem Resolution Reduces manual intervention by autonomously diagnosing and resolving IT issues.
2. Reduced Downtime Minimizes disruptions by quickly identifying and addressing problems.
3. Cost Efficiency Lowers operational costs by reducing the need for large IT support teams.
4. Improved Accuracy Utilizes AI and ML to classify problems with precision and adapt to feedback.
5. Scalability Handles increasing complexities in IT environments without compromising performance.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. Knowledge Retention Maintains a database of past cases, creating a reusable knowledge base.

The future impact of self-healing problem systems is far-reaching. As IT environments grow increasingly complex, these systems will become indispensable for maintaining operational efficiency. By integrating advanced AI models, they will predict and resolve issues with unprecedented accuracy, setting new benchmarks for automation in IT. the adoption of self-healing systems will revolutionize industries beyond IT, such as healthcare, manufacturing, and smart cities, by offering adaptive and resilient solutions. The potential to integrate these systems with IoT devices and cloud infrastructures will create a fully automated and intelligent ecosystem. Ultimately, self-healing systems will drive the evolution of IT support, shifting the focus from reactive problem-solving to proactive innovation. This paradigm shift will empower organizations to allocate resources toward strategic growth rather than routine maintenance, ensuring long-term sustainability and technological advancement.

REFRENCES

[1] Sacco, C., Radwan, A. B., Anderson, A., Harik, R., & Gregory, E. Machine learning in composites manufacturing: A case study of Automated Fiber Placement inspection. Composite Structures, 250, 112514. 2020.
[2] Nadir K.Salih, Doha Idries. Self-Healing Applications By Using Case Base Reasoning Algorithm. International Journal of Research and Innovation in Applied Science (IJRIAS). Volume IX Issue VIII. pp 32-42. 2024.
[3] Nadir K. Salih, R. Gopal. Optimization Software Fault Injection by Using Two- Dimensional Bin Packing Algorithms. Journal of Electrical Systems. Vol. 20 No. 10s. October 2024.
[4] Mousa, M. R. . Analyzing cyber-attack intention for digital forensics using case-based reasoning. 2021 Huang, K., Nie, W., & Luo, N.). Scenario-based marine oil spill emergency response using hybrid deep reinforcement learning and case-based reasoning. Applied Sciences, 10(15), 5269. 2020.
[5] Wattimena, F. Y., & Rofi'i, Y. U. E-Commerce Product Recommendation System Using Case-Based Reasoning (CBR) and K-Means Clustering. International Journal Software Engineering and Computer Science (IJSECS), 3(2), 162-173. 2023
[6] Ahn, J., Ji, S. H., Ahn, S. J., Park, M., Lee, H. S., Kwon, N., ... & Kim, Y. (). Performance evaluation of normalization-based CBR models for improving construction cost estimation. Automation in Construction, 119, 103329. 2020
[7] Jimenez, J. J. M., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. Journal of manufacturing systems, 56, 539-557. (2020)
[8] Sallu, S., Sianturi, N. M., Purwoko, B., Herliansyah, Y., & Manuhutu, M. A. Learning in Higher Education Based on Artificial Intelligence (AI) with Case Based Reasoning (CBR). Journal of Namibian Studies: History Politics Culture, 34, 1049-1064. 2023
[9] Bender, C., Cichosz, S. L., Malovini, A., Bellazzi, R., Pape-Haugaard, L., & Hejlesen, O. Using case-based reasoning in a learning system: a prototype of a pedagogical nurse tool for evidence-based diabetic foot ulcer care. Journal of Diabetes Science and Technology, 16(2), 454-459. 2022
[10] Escobar, N., & Laibach, N. Sustainability check for bio-based technologies: A review of process-based and life cycle approaches. Renewable and Sustainable Energy Reviews, 135, 110213. 2021.
[11] Graef, R., Klier, M., Kluge, K., & Zolitschka, J. F.. Human-machine collaboration in online customer service– a long-term feedback-based approach. Electronic Markets, 31, 319-341. 2021.
[12] Cattan, G. H., & Quemy, A.Case-Based and Quantum Classification for ERP-Based Brain–Computer Interfaces. Brain Sciences, 13(2), 303. 2023
[13] Schack, B. Case-base maintenance beyond case deletion (Doctoral dissertation, Indiana University). (2023).
[14] Aminoshariae, A., Kulild, J., & Nagendrababu, V.. Artificial intelligence in endodontics: current applications and future directions. Journal of endodontics, 47(9), 1352-1357. 2021
[15] Engineering, J. O. H. (2023). Retracted: Nursing Care Systematization with Case-Based Reasoning and Artificial Intelligence. Journal of healthcare engineering, , 9814073. 2023
[16] Kamran, D. Using Reported Symptoms, a Neuro-Fuzzy Case-Based Reasoning System Can Detect Lassa. 2019.
[17] Saleh, H. Using the Case-Based Reasoning-YOLO Approach for Rapid and Effective Identification of Bruises. 2023.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[18] Nadir K.Salih, Hiya Al-Risi, Shama ALjabri, Rawdah Al.Shamsi. Autonomous Delivery Robot for Pandemic Case. Research Inventy: International Journal of Engineering And Science. Vol.14, Issue , PP 86-92. 2024.
[19] Nadir K.Salih, Saria Mohieldeen. Self-protection of Mobile Applications Using Reflection Design. International Journal of Innovative Science and Research Technology. Volume 11 Issue 8. pp 117-124. 2024.