International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

Automated Self-Healing Maintenance System for Windows Server

[1]Ruchiranga G.K.N., [2]Bandara G.K.A.H., [3]Rathnayake L.A.N.M., [4]Ganepola G.A.T.S., [5]Anjalie Gamage, [6]Amali Gunasinghe

[1,2,3,4]Department of Computer Systems Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[5]Senior Lecturer, Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[6]Assistant Lecturer, Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
Authors E-mail: [1]it20131210@my.sliit.lk, [2]it20004118@my.sliit.lk, [3]it20130534@my.sliit.lk, [4]it20126988@my.sliit.lk, [5]angalie.g@sliit.lk, [6]amali.g@sliit.lk

Abstract **- Ensuring the best performance of a Windows Server system is critical for businesses since any operational failures may result in significant financial losses. Considering this, the use of automated mistake resolution solutions appears as a proactive technique for dealing with prospective concerns. This study provides a ground-breaking technique specialized for Windows Server settings, consisting of three essential pieces deliberately intended to identify and control faults. A thorough examination of the first factor, Predictive Errors, necessitates a thorough examination of server performance data. By studying this data, the system may proactively detect locations where problems may occur. This foresight allows the system to take preventive steps, eventually reducing downtime and improving overall server performance. The second component, the Automated Error Resolution module, uses scripts and automated procedures to correct mistakes that occur. Notably, this module allows for the inclusion of human operators in the resolution process when necessary, guaranteeing a balanced and successful approach. Configuration Management is critical in preserving the server environment's accuracy. It manages the tracking of configuration changes using a combination of automated and human-driven mechanisms. This comprehensive approach results in an integrated solution that adeptly prioritizes the correction of misconfigured services based on their possible influence on the overall performance of the system. Acknowledgment The research activities are divided into three stages. The first phase is a thorough Literature Review to obtain insights and knowledge from current sources. This informs the succeeding System Design and Implementation phase, in which a prototype of the desired system is created. This prototype is subjected to a thorough review that considers issues such as accuracy and scalability. The prototype is exposed to real-world testing situations in the last phase, Evaluation and Validation, to fine-tune its functionality and improve its performance. The consequences of this research are considerable, with the potential to greatly improve the reliability of Windows Server settings. This research offers itself as a disruptive method with far-reaching advantages for organizations by reducing downtime, improving performance, and implementing cost-effective strategies.**

Keywords: Maintenance system, Automation, Self-healing, Predictive maintenance, Windows servers, Configuration Management, Fault Management, Error resolution, Machine learning, reliability, uptime, accuracy, efficiency, Web-based dashboard.

I. INTRODUCTION

In today's Business landscape, the smooth and continuous operation of IT systems is a vital aspect of preserving operational efficiency. Among these systems, Windows Server environments play a critical role in providing a wide range of organizational duties, from basic file sharing to more complicated services such as web hosting and application administration. However, relying on such complex ecosystems exposes firms to the inherent vulnerabilities of mistakes and system failures, which can result in severe downtime and financial consequences.

To solve these issues, automatic self-healing and maintenance systems appear to be a potential approach. These systems identify, diagnose, and correct mistakes and problems before they cause substantial operational disruptions by harnessing the capabilities of machine learning, artificial intelligence, and automation. This study presents a complete and customized automated self-healing and maintenance architecture intended exclusively for Windows Server settings.

This structure is built on three primary functions, each of which serves a distinct purpose within the greater system. The first function is Predictive Error Detection, which is accomplished by analyzing system data with tools such as Task Manager and Windows Event Viewer. Potential mistakes

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

can be recognized before they cause significant failures by recognizing patterns and anomalies.

The second critical function is Automated Issue Resolution. When faults are found, this function uses automated diagnostic processes to identify the underlying causes of the problems and then implements the appropriate remedies. This quick and self-sufficient reaction reduces the time necessary to handle errors, decreasing possible downtime and assuring smoother operations.

Configuration Management is the third and equally important job. This entails creating configuration scripts in PowerShell and developing a Windows service in C#. This feature automates the complex operations required in establishing, maintaining, and continuously monitoring server systems, greatly reducing the manual strain put on IT professionals. This, in turn, improves the server infrastructure's security and resilience.

Furthermore, the suggested system makes use of cloud-based services to improve its capabilities. The use of services such as AWS Lambda, AWS S3, AWS EC2, AWS RDS, and AWS API Gateway helps the framework's resilience and scalability. These services allow for seamless integration, efficient data storage, and dynamic computing resources, guaranteeing that the framework remains successful even in complex and demanding contexts.

The portions of this study paper that follow are painstakingly arranged to look deeper into the aspects discussed above. "Section 2: Related Work" presents a thorough examination of previous research on the topic of automated self-healing and maintenance systems, setting the current work within a larger academic context. "Section 3: Proposed System Architecture" provides a thorough overview of the framework's design, unraveling the many linkages between its operations. "Section 4: Implementation Details" delves into the framework's practical implementation, illuminating the technical features and concerns for its integration. "Section 5: Performance Evaluation" systematically evaluates the framework's efficacy, substantiating its assertions with facts and rigorous analysis.

In conclusion, this article provides a unique architecture that is positioned to improve the dependability and stability of Windows Server systems. This solution provides a complete approach to reducing mistakes, decreasing downtime, and improving the operational resilience of modern enterprises through automation, diagnostics, predictive analytics, and cloud integration.

II. LITERATURE REVIEW

A) Related Works and Critical Evaluation

Automated self-healing maintenance systems have developed as an important field of study and development in computer systems and cloud computing. A. Sahai, S. Ranjan, and A. Joshi researched the automatic self-healing of web servers [1]. This ground-breaking paper describes a method that uses machine learning to detect mistake patterns and anomalies. When a problem occurs, the system responds autonomously by writing repair scripts, a big step toward efficient system maintenance. In Automated Self-Healing of Database Systems Using Machine Learning Research A. Gupta, A. Gupta, and R. Buyya expanded the idea to database systems [2],. Their study describes a system that uses machine learning to detect mistakes and deviations in database operations and, once identified, automatically conducts repair processes, therefore increasing the dependability of database management systems. S. Zhang, H. Zhang, and J. Zhao propose a novel method for autonomous self-healing in cloud computing systems [3]. Their technology blends machine learning with rule-based systems, resulting in a comprehensive technique for detecting and correcting problems, assuring the smooth running of cloud services. S. Kaur, S. Singh, and M. Goyal investigated the important subject of virtual machines in cloud settings [4]. Their study proposes a method that uses both machine learning and rule-based systems to detect and correct problems in virtual machine configurations, therefore improving the reliability and efficiency of cloud-based infrastructure. Y. Liu, X. Zhang, and J. Sun suggested a solution for autonomous self-healing of software systems using reinforcement learning in their study [5].

This novel technique represents a significant advancement in enabling systems to independently learn how to address and rectify faults by closely monitoring their operational conditions and implementing suitable responses. The publications in question provide a comprehensive overview of the research trajectory within the domain of automated self-healing maintenance systems. They exemplify the shift from early reliance on machine learning and rule-based systems toward the more sophisticated approach of reinforcement learning. Foreseeing the continued progress in this field, we anticipate the development of increasingly intricate and effective methodologies. These will play a pivotal role in enhancing the resilience and dependability of contemporary computing environments. In sum, this research heralds a promising future for self-healing systems and their contribution to ensuring the reliability of computing operations.

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

The study on automatic self-healing maintenance systems has proved encouraging. These technologies have the potential to cut downtime dramatically, improve system availability, and boost operational efficiency. However, several problems must be overcome before these technologies can be extensively used. One problem is the requirement for precise and dependable data. Data is used by automated self-healing maintenance systems to discover and diagnose faults. If the data is erroneous or untrustworthy, the system may be unable to efficiently identify and rectify mistakes. Another problem is the requirement to create systems that are both resilient and scalable. Self-healing automated maintenance systems must be able to manage a wide range of faults and difficulties. They must also be scalable to manage a high number of systems. Finally, there is a requirement to ensure the security of automatic self-healing maintenance systems. These systems will have access to confidential information and systems. It is critical to secure sensitive information from unwanted access.

B) Research Gap and RQs

Finally, developing security methods to secure sensitive data and systems inside automated self-healing maintenance systems is a problem that necessitates creative solutions. Researchers must deal with the difficult task of safeguarding automated systems while keeping them efficient and responsive. As this sector develops, it has the potential to transform the dependability and availability of IT systems. Addressing these research gaps and challenges will be critical to fully using the possibilities of automated self-healing maintenance systems and ushering in a more robust and dependable IT infrastructure.

In the landscape of maintenance systems, The Automated Self-Healing Maintenance System aiming to bridge fundamental gaps that distinguish our Automated Self-Healing Maintenance System from traditional counterparts. These research gaps underscore the crucial need for innovation in the field of Windows Server maintenance. While existing systems are reactive, merely informing users of errors and relying on human expertise for resolution, our system pioneers a proactive approach. It can proactively detect errors and their underlying causes, applying precise resolutions automatically, this proactive approach substantially reduces system downtime and the extensive reliance on human intervention, consequently enhancing system reliability and user satisfaction.

The Automated Self-Healing Maintenance System stands out for its real-time monitoring capabilities, particularly in the realm of critical service configurations. Unlike traditional systems, which often require manual oversight, our system excels in autonomously identifying misconfiguration incidents and applying the necessary configurations for immediate resolution. This unique feature not only bolsters system dependability but also mitigates the risk of human-induced errors in the configuration process. Our research focuses on bridging these substantial gaps, emphasizing the core principles of efficiency, reliability, and automation.

The Automated Self-Healing Maintenance System is our overarching goal is to pave the way for the evolution of maintenance systems tailored to Windows Server environments. Through an emphasis on proactive error identification and resolution, as well as real-time misconfiguration management, our research questions revolve around novel approaches to enhance system reliability and efficiency, ultimately contributing to the realization of a robust and dependable IT infrastructure. In bridging these research gaps, we are committed to pioneering a new era in maintenance systems that can efficiently adapt to the dynamic challenges posed by modern IT environments.

III. METHODOLOGIES

A) Predictive Errors

1) Identify High-Performance Usage Services

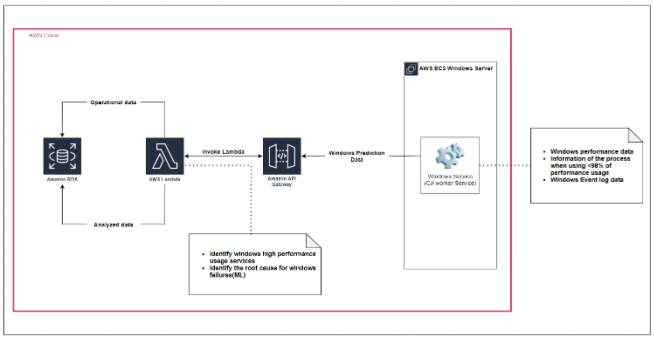

Figure 1: Predict Errors

This part of the component is designed to efficiently monitor and track the high-performance usage services that have a significant impact on the performance of the Windows Server.

The core of this system is a dedicated Windows service that runs at regular intervals, specifically every 30 seconds. The Windows service is responsible for constantly monitoring the server's performance metrics, including CPU utilization, memory consumption, and disk I/O. When any of these performance metrics surpass the critical threshold of 98%, the Windows service immediately initiates the tracking process.

Upon detecting a performance anomaly, the Windows service identifies the top high-performance usage services that are causing the server's performance to exceed the designated

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

threshold. It collects relevant data about these services and stores this information in a database for further analysis and reporting. In high-performance usage events, when detected (i.e., performance exceeding 98%), trigger a Lambda function in the AWS cloud environment. This Lambda function is responsible for interacting with an RDS database, where it retrieves and aggregates data about high-performance usage services. Subsequently, web-based dashboard repots inform relevant users about the identified high-performance usage services.

2) Predict Errors using Event Logs

This section of the component serves the purpose of pinpointing the reasons behind errors, utilizing machine learning techniques and historical data to isolate the root causes of past errors. By leveraging this historical data, the component can forecast future errors.

The Continual collection and storage of Windows server event log data. These logs contain critical information about system events, errors, and warnings. The system maintains a persistent connection to an RDS (Relational Database Service) to store this event log data, establishing an extensive historical record of server events.

When an error event occurs within the Windows server environment, the Windows service in the Windows server triggers the Lambda function within the AWS cloud. This Lambda function plays a pivotal role in executing a comprehensive analysis process, underpinned by a decision tree model (classification model) that leverages Machine Learning (ML) techniques. This ML-based classification model scrutinizes event log descriptions, searching for keywords and patterns that are indicative of error causes. By employing ML, the model can discern the most probable event log entry responsible for the error, effectively isolating the root cause.

Once the root cause is identified, the system stores this prediction within the RDS database. Additionally, it communicates this information to users through a web-based dashboard, providing insights into the probability percentage of the predicted error. If the same error cause recurs, the component proactively informs users through the dashboard, indicating specific services that are now in warning status, thereby enabling timely intervention and maintenance. This methodology ensures that errors are not only identified but also predicted and communicated efficiently to enhance the self-healing capabilities of the overall system.

B) Automated Error Resolution and Configuration Management

1) Automated Error Resolution

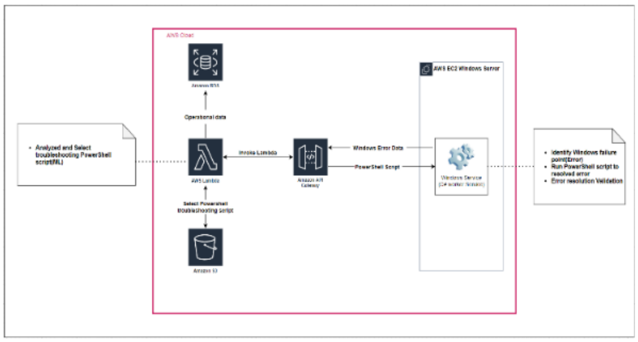

Figure 2: Automated Error Resolution

The "Automated Error Resolution" component in the Automated Self-Healing Maintenance System for Windows Server is fundamentally focused on the technical aspects of error resolution within the Windows server environment. This systematic approach revolves around the efficient handling of error events through a series of well-defined steps.

Error resolution begins with the detection of errors within the Windows server. To achieve this, a dedicated Windows service continuously monitors the system, identifying error events as they occur. When an error is detected, this service promptly collects comprehensive information about the failure points, including specific details about the error. This critical error data is then transmitted to an RDS (Relational Database Service), where it is centrally stored for further analysis and resolution. To facilitate this analysis, an AWS Lambda function is triggered within the AWS cloud environment. This Lambda function plays a crucial role in processing the error information.

The heart of the analysis process lies in a classification model that employs the Decision Tree Algorithm, a Machine Learning (ML) technique. This classification model in lambda function intelligently analyzes and pinpoints the optimal PowerShell script to address a detected error. The classification model plays a crucial role in associating error information with the corresponding PowerShell commands within the script. Once the appropriate PowerShell script is selected, the lambda function passes it over to a server-hosted Windows service. This service then executes the PowerShell script, confirming whether the error has been rectified.

All operational data, including the results of the error resolution process, as well as detailed logs, are diligently recorded in the RDS database for reference and analysis. This data serves as the foundation for a web-based dashboard that provides users with valuable insights into the efficiency of the

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

error resolution process, enabling a proactive approach to system maintenance and ensuring the smooth operation of the Windows server environment.

2) Configuration Management

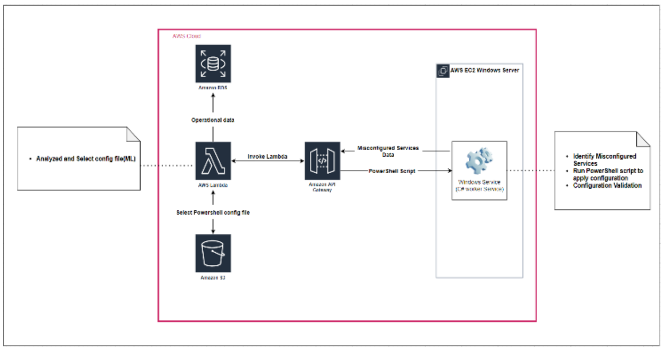

Figure 3: Configuration Management

The "Configuration Management" component within the Automated Self-Healing Maintenance System for Windows Server is defined by its technical underpinnings, meticulously designed to automate the configuration management process for Windows Server. This process encompasses a structured sequence of steps to ensure seamless standardization and efficient management of Configuration on services as required by users.

The system's operation begins with the user adding standard configuration data for specific services. This configuration data that the user provided is stored within the RDS (Relational Database Service). The RDS, acting as the central repository for standard configuration information, serves as the bridge between user-defined standards and the system's automated management process.

The Windows service, a core component of the system, continuously monitors the standard configuration status of specific services in the Windows server. This monitoring occurs at regular intervals, with the service checking the configuration data every 30 seconds. The objective is to promptly identify any service that falls into a misconfigured status, potentially compromising system integrity.

Upon detecting a misconfigured service, the Windows service takes immediate action. It records the details of the misconfiguration, including the specific points at which the service deviates from the predefined standards, and transmits this information back to the RDS database. Additionally, the service triggers an AWS Lambda function, which initiates the analysis of the misconfiguration issue. To resolve the misconfiguration, the system relies on a repository of multiple PowerShell scripts stored in an AWS S3 bucket. The Lambda function plays a critical role in selecting the most suitable PowerShell script for the identified misconfiguration issue. This selection process leverages a classification model underpinned by the Decision Tree Algorithm, a Machine Learning (ML) technique. The classification model systematically analyzes the misconfiguration data, considering the misconfigured points and their significance. It identifies key points within the misconfiguration information and matches these key points with the corresponding PowerShell commands within the scripts. ultimately selecting the most suitable PowerShell script as a solution to resolve the misconfiguration issue. The Selected PowerShell script is then transmitted to the Windows service operating within the Windows Server environment. The Windows service executes the script to address the misconfiguration and subsequently validates whether the issue has been successfully resolved.

Detailed operation data and logs generated throughout this process are meticulously stored within the RDS database for reference and analysis. This operational data forms the basis for a web-based dashboard that provides users with real-time insights into the efficiency of the solution, ensuring that misconfigurations are swiftly identified and rectified, thereby maintaining the integrity and performance of the Windows Server environment.

IV. DATA ANALYSIS

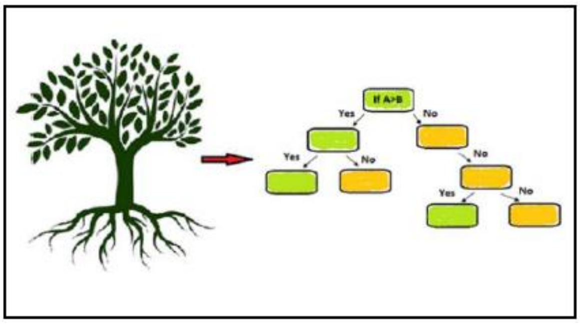

Figure 4: Decision Tree Algorithms

The Decision Tree Algorithm (DTA) serves as the backbone of the system's data analysis capabilities, offering a structured and interpretable approach to handling complex datasets. Its application varies across the three components.

The DTA scrutinizes event log descriptions, meticulously searching for keywords and patterns indicative of error causes for the error prediction component. Through the utilization of machine learning (ML), the model discerns the most probable event log entry responsible for the error, thus effectively isolating the root cause. This systematic analysis ensures that the system not only detects errors but also identifies their origins with precision [6]. In the Automated Error Resolution component, The DTA embedded within the Lambda function takes on the role of intelligently analyzing and pinpointing the optimal PowerShell script to address a detected error. The

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

classification model plays a pivotal role in associating error information with the corresponding PowerShell commands within the script. This process ensures that error resolution is not a generic approach but is tailored to the specific error at hand, optimizing the effectiveness of the resolution process [7]. For the Configuration Management component, The DTA systematically analyzes misconfiguration data. It takes into account the misconfigured points and their significance. The model identifies key points within the misconfiguration information and matches these key points with the corresponding PowerShell commands within the scripts. This methodical analysis ensures that misconfigurations are rectified precisely, maintaining system integrity, and adhering to user-defined standards [8].

Jupyter serves as an invaluable tool for illustrating the data analysis process step by step by arranging code. It provides a transparent and interactive environment for exploring and visualizing datasets, enabling researchers and administrators to gain a deeper understanding of the data being processed. With Jupyter, the analysis process becomes more transparent and interpretable, making it easier to fine-tune ML models and refine decision-making [9].

Categorizing datasets is a fundamental technique used to structure and organize data for analysis [10]. It is a process of grouping data into categories based on common characteristics. This makes it easier to understand the data and to apply the Decision Tree Algorithm effectively. In the context of this system, datasets are categorized into three main types: error logs, configuration data, and misconfiguration details. Error logs contain information about errors that have occurred in the system. Configuration data stores information about the system's configuration settings. Misconfiguration details record information about misconfigurations that have been found in the system.

Labeling datasets is another crucial technique that aids in organizing and understanding data. It is the process of assigning labels to each data point in a dataset. These labels can be numerical or categorical. In the system, labels are used to identify and classify different types of errors, configurations, or misconfigurations. logs dataset will be labeled with the type of error that occurred, such as a syntax error or a runtime error. Configuration data will be labeled with the name of the configuration setting, such as the port number or the database name. Misconfiguration details will be labeled with the type of misconfiguration, such as a missing setting or an incorrect setting. The labeling process ensures that the Decision Tree Algorithm can make accurate decisions and provide tailored solutions based on the labeled data categories [11].

V. RESULTS

Evaluating the performance of our automated self-healing maintenance system for Windows Server, we conducted a scientific evaluation employing both quantitative and qualitative measures. Our evaluation process involved the training, validation, and testing of machine learning models, as well as assessing their effectiveness in real-world scenarios.

We initiated the evaluation process by collecting real-world data related to errors and misconfigurations in Windows Server environments. The data included incidents occurring in Internet Information Services (IIS), Active Directory, and database services. This dataset served as the foundation for training our machine learning models. We leveraged Jupyter notebooks to train and validate each model.

During the initial stages of model training, there were instances of multiple output loss situations. Through continuous and iterative model training, we successfully mitigated these issues.

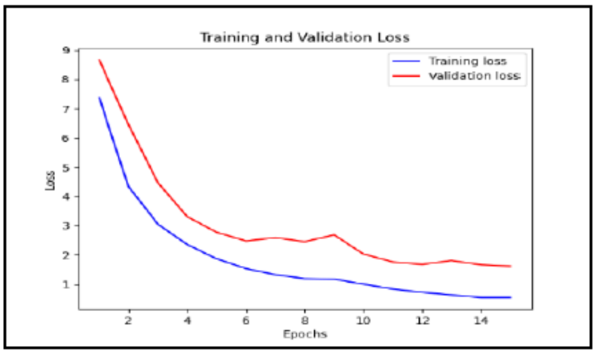

Figure 5: Overall Training and Validation Loss

Our efforts culminated in significant enhancements in the accuracy of each model. This improvement was a strong indicator that our models accurately identified the root causes of errors, selected appropriate resolutions, and applied the correct configurations for misconfiguration incidents.

We integrated the trained machine learning models into our automated self-healing maintenance system. After fine-tune the maintenance system we require to validate the effectiveness of our maintenance system, we employed previously unseen datasets, containing records of errors and misconfigurations incidents within IIS, Active Directory, and database services. These datasets were used to assess whether the models within our system could perform as anticipated. In the initial stages of testing, we encountered minor instances of output loss. However, after conducting multiple rounds of testing, we consistently observed the models performing in alignment with our expectations.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

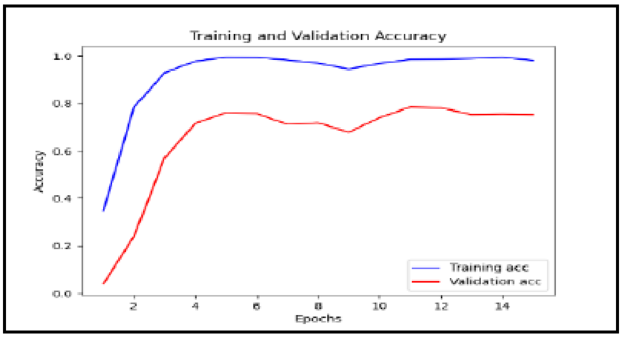

Figure 6: Overall Training and Validation Accuracy

Comparing the training and validation statistics, we noted a significant reduction in the count of output losses after repeated training and testing cycles. This improvement was accompanied by an increase in the overall accuracy of our models. The statistical results obtained from this process serve as compelling evidence that our maintenance system has demonstrated a high degree of success in meeting its intended objectives.

This scientific evaluation underlines the practical significance of our work and serves as a noteworthy contribution to the existing body of knowledge. Our solution not only improves the dependability and efficiency of Windows Server environments but also stands as a model for proactively addressing issues and maintaining standardized settings, thereby enhancing the overall system reliability, cost-effectiveness, and user satisfaction.

VI. CONCLUSION

This research project's "Automated Self-Healing Maintenance System for Windows Server" includes an innovative approach to improving the dependability and robustness of Windows Server settings. This revolutionary technology is made up of three key components: "Predictive Errors," "Automated Error Resolution," and "Configuration Management," all of which have been methodically built to handle crucial areas of system maintenance and administration.

The system demonstrates its capacity to foresee difficulties using machine learning approaches in the "Predictive Errors" component. It detects mistakes, determines fundamental causes using past data, and anticipates prospective faults, promoting a proactive approach to system maintenance.

The component "Automated Error Resolution" shows the system's agility and efficiency in error handling. It finds faults quickly, chooses appropriate resolutions using machine learning methods, and implements them using PowerShell scripts. This component speeds up the error resolution process, ensuring that system operations are not disrupted. "Configuration Management" demonstrates the system's dedication to keeping a standardized and properly configured Windows Server environment. It guarantees that services adhere to user-defined criteria by continuous monitoring and ML-driven analysis, reducing misconfigurations and enhancing system integrity.

These three components work together to offer a comprehensive and forward-thinking approach to Windows Server maintenance. The solution not only automates key activities but also provides users with real-time information via a user-friendly web-based dashboard by integrating cutting-edge technology such as machine learning and AWS cloud services.

The "Automated Self-Healing Maintenance System for Windows Server" serves as a beacon of efficiency, resilience, and innovation in assuring the sustained health and performance of these systems as enterprises increasingly rely on Windows Server environments for their operations. It lays the groundwork for a future in which self-healing and self-optimizing IT ecosystems are the norm, improving key infrastructure dependability and efficiency.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to our mentors and specialists, whose invaluable guidance and wisdom have steered our research towards fruitful outcomes. Our deepest appreciation also goes to the industry professionals, consultants, and peers who generously engaged in discussions and provided critical feedback, playing a pivotal role in shaping our research methodology. Their collective insights and expertise have been instrumental in the success of this endeavor.

REFERENCES

[1] A.Sahai, S. Ranjan, and A. Joshi, "Automated Self-Healing of Web Servers," Proceedings of the 2016 ACM SIGOPS Asia Conference on Operating Systems, 2016.

[2] A.Gupta, A. Gupta, and R. Buyya, "Automated Self-Healing of Database Systems Using Machine Learning," Proceedings of the 2017 IEEE International Conference on Cloud Engineering, 2017.

[3] S. Zhang, H. Zhang, and J. Zhao, "Automated Self-Healing of Cloud Computing Systems Using Machine Learning and Rule-Based Systems," Proceedings of the 2018 IEEE International Conference on Cloud Computing, 2018.

[4] S. Kaur, S. Singh, and M. Goyal, "Automated Self-Healing of Virtual Machines in Cloud Environment,"

© 2023-2017 IRJIET All Rights Reserved

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 282-289, October-2023

https://doi.org/10.47001/IRJIET/2023.710036

Proceedings of the 2019 IEEE International Conference on Cloud Computing, 2019.

[5] Y. Liu, X. Zhang, and J. Sun, "Automated Self-Healing of Software Systems Using Reinforcement Learning," Proceedings of the 2020 IEEE International Conference on Cloud Computing, 2020.

[6] James, Gareth, et al. "An introduction to statistical learning." Springer, 2013.

[7] Hastie, Trevor, et al. "The elements of statistical learning." Springer, 2009.

[8] Friedman, Jerome H. "Regularization paths for generalized linear models." Journal of Machine Learning Research 10 (2009): 3309-3336.

[9] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12 (2011): 2825-2830.

[10] Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann.

[11] Zhang, Z., & Yang, Q. (2012). Categorizing and labeling datasets for decision tree algorithm. Expert Systems with Applications, 39(10), 9621-9629.

Citation of this Article:

Ruchiranga G.K.N., Bandara G.K.A.H., Rathnayake L.A.N.M., Ganepola G.A.T.S., Anjalie Gamage, Amali Gunasinghe, “Automated Self-Healing Maintenance System for Windows Server” Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 10, pp 282-289, October 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.710036

****

© 2023-2017 IRJIET All Rights Reserved