

AegisAI: An AI-Driven Predictive Health Monitoring Framework for Windows Server Infrastructure

Mohammed Ajmal A

Department of Master of Computer Applications (MCA)
Kalasalingam Academy of Research and Education (KARE)
Krishnankoil, Virudhunagar District, Tamil Nadu, India
Email: 9224116063@cdoc.klu.ac.in

ABSTRACT

Enterprise organizations increasingly depend on Windows Server infrastructure to host mission-critical applications, virtualization platforms, databases, authentication services, and cloud-based workloads. Ensuring continuous availability and optimal performance of these systems has become increasingly challenging due to the growing complexity of enterprise IT environments. Most existing infrastructure monitoring solutions primarily rely on reactive monitoring mechanisms, where administrators are notified only after predefined thresholds are exceeded or system failures have already occurred. Such approaches often lead to delayed incident response, increased downtime, inefficient resource utilization, and higher operational costs.

This paper proposes **AegisAI**, an Artificial Intelligence (AI)-driven predictive health monitoring framework designed to transform conventional server monitoring into an intelligent, proactive infrastructure management solution. The proposed framework continuously collects and analyzes critical Windows Server performance metrics, including processor utilization, memory consumption, disk usage, network throughput, Windows service status, and Event Log information. Using machine learning techniques and predictive analytics, AegisAI identifies abnormal behavior, estimates infrastructure health, predicts potential failures, and recommends preventive maintenance actions before service degradation occurs.

The proposed framework introduces four core components: the **AegisAI Health Index (AHI)** for comprehensive infrastructure health assessment, the **Predictive Intelligence Engine (PIE)** for failure prediction, **Adaptive Risk Prioritization (ARP)** for intelligent alert classification, and **Predictive Infrastructure Risk Analysis (PIRA)** for estimating future infrastructure risks. Together, these components provide explainable predictions, centralized visualization, automated reporting, and data-driven decision support for infrastructure administrators.

The proposed framework aims to improve infrastructure reliability, minimize unexpected downtime, optimize resource utilization, and support proactive infrastructure maintenance through intelligent analytics. AegisAI provides a scalable architecture suitable for enterprise Windows Server environments and demonstrates the potential of Artificial Intelligence in modern infrastructure management.

Keywords

Artificial Intelligence, Windows Server, Predictive Monitoring, Infrastructure Health Monitoring, Machine Learning, Enterprise Infrastructure, Predictive Maintenance, System Administration.

I. INTRODUCTION

Modern enterprises increasingly depend on Windows Server infrastructure to support business-critical applications, virtualization platforms, enterprise databases, web applications, authentication services, cloud computing environments, and digital business operations. As organizations continue expanding their digital infrastructure, maintaining server availability, ensuring operational reliability, and minimizing service disruptions have become major responsibilities for infrastructure administrators. Even a brief period of server downtime can lead to financial losses, reduced productivity, service interruptions, and customer dissatisfaction.

Traditional infrastructure monitoring solutions primarily operate using reactive monitoring techniques. These systems continuously observe infrastructure metrics and generate alerts only after predefined threshold values have been exceeded or failures have already occurred. Although such monitoring platforms provide valuable operational insights, they offer limited capabilities for predicting future failures or identifying hidden infrastructure risks. Consequently, administrators frequently spend significant time responding to incidents rather than preventing them.

Recent advances in Artificial Intelligence (AI) and Machine Learning (ML) have enabled organizations to move beyond traditional threshold-based monitoring by introducing predictive infrastructure analytics. AI models can learn historical operational behavior, detect abnormal system patterns, estimate infrastructure health, and predict future failures before they affect production services. Integrating AI with infrastructure monitoring enables organizations to shift from reactive maintenance toward proactive infrastructure management.

This research proposes **AegisAI**, an AI-driven predictive health monitoring framework specifically designed for Windows Server infrastructure. Unlike conventional monitoring platforms that merely report system status, AegisAI continuously evaluates infrastructure health, predicts potential failures, prioritizes operational risks, and provides explainable maintenance recommendations through intelligent analytics.

The proposed framework introduces four integrated modules that collectively improve enterprise infrastructure management. The **AegisAI Health Index (AHI)** calculates a comprehensive infrastructure health score using multiple system performance indicators. The **Predictive Intelligence Engine (PIE)** applies machine learning techniques to forecast infrastructure failures before operational disruption occurs. The **Adaptive Risk Prioritization (ARP)** module intelligently ranks infrastructure alerts according to operational severity, while the **Predictive Infrastructure Risk Analysis (PIRA)** module estimates future infrastructure risks by analyzing historical operational trends.

The objective of this research is to design an intelligent predictive monitoring framework capable of improving infrastructure reliability, reducing operational downtime, optimizing administrative decision-making, and supporting enterprise-scale Windows Server environments through Artificial Intelligence.

II. PROBLEM STATEMENT

Enterprise Windows Server environments generate thousands of operational events every hour, including processor utilization, memory consumption, storage activity, network traffic, Windows service status, application logs, security logs, and operating system events. Infrastructure administrators must continuously analyze these metrics to ensure uninterrupted service availability and maintain overall system health.

Most existing infrastructure monitoring solutions rely primarily on threshold-based alerting mechanisms that notify administrators only after abnormal operating conditions have already occurred. Although these monitoring platforms successfully identify operational issues, they provide limited predictive intelligence regarding future infrastructure failures. Consequently, organizations often experience unexpected downtime,

delayed incident response, inefficient resource utilization, and increased operational expenses.

Furthermore, conventional monitoring systems typically treat all alerts with equal importance, making it difficult for administrators to prioritize incidents based on operational risk. The absence of intelligent health assessment, predictive analytics, and explainable maintenance recommendations significantly limits proactive infrastructure management.

Therefore, there is a growing need for an AI-driven predictive health monitoring framework capable of continuously evaluating infrastructure health, forecasting potential failures, prioritizing operational risks, and supporting preventive maintenance before service degradation occurs.

III. RESEARCH OBJECTIVES

The primary objectives of this research are:

- To design an Artificial Intelligence-driven predictive health monitoring framework for Windows Server infrastructure.
- To continuously collect real-time infrastructure performance metrics from enterprise Windows Server environments.
- To develop the AegisAI Health Index (AHI) for comprehensive infrastructure health assessment.
- To design the Predictive Intelligence Engine (PIE) capable of forecasting potential infrastructure failures.
- To implement Adaptive Risk Prioritization (ARP) for intelligent infrastructure alert classification.
- To develop Predictive Infrastructure Risk Analysis (PIRA) for estimating future infrastructure risks.
- To generate explainable AI-based maintenance recommendations for infrastructure administrators.
- To improve proactive decision-making through intelligent predictive analytics.
- To reduce unexpected infrastructure downtime and improve operational reliability.
- To develop a centralized web dashboard for monitoring, visualization, alert management, and predictive reporting.

IV. RESEARCH CONTRIBUTIONS

The major contributions of this research are summarized as follows:

- Proposal of the **AegisAI Framework**, an AI-driven predictive health monitoring framework tailored for enterprise Windows Server infrastructure.
- Introduction of the **AegisAI Health Index (AHI)** to provide a unified measurement of

server health based on multiple operational parameters.

- Development of the **Predictive Intelligence Engine (PIE)** for forecasting potential infrastructure failures using machine learning techniques.
- Design of the **Adaptive Risk Prioritization (ARP)** mechanism to dynamically classify and prioritize infrastructure alerts based on operational severity.
- Implementation of the **Predictive Infrastructure Risk Analysis (PIRA)** model to estimate future infrastructure risks through historical trend analysis.
- Integration of predictive analytics, intelligent recommendations, centralized visualization, and automated reporting into a unified enterprise monitoring framework.
- Provision of a scalable and extensible architecture that supports proactive infrastructure management and enhances operational reliability in enterprise Windows Server environments.

V. RELATED WORK

Recent advancements in Artificial Intelligence (AI), Machine Learning (ML), and cloud computing have significantly improved the capabilities of enterprise infrastructure monitoring systems. Researchers have proposed various intelligent monitoring frameworks to detect anomalies, predict failures, and optimize infrastructure performance. Traditional monitoring platforms such as Nagios, Zabbix, PRTG, and Microsoft System Center primarily rely on threshold-based monitoring techniques, where alerts are generated only after predefined limits are exceeded. Although these systems provide effective real-time monitoring, they offer limited predictive capabilities and require administrators to respond after abnormal conditions have already occurred.

Several studies have explored the integration of machine learning techniques into infrastructure monitoring. These approaches utilize historical system logs, performance counters, and event data to identify hidden operational patterns and predict future failures. Algorithms such as Decision Trees, Random Forest, Support Vector Machines, Artificial Neural Networks, and Long Short-Term Memory (LSTM) networks have been employed to improve anomaly detection and predictive maintenance. While these models demonstrate promising prediction accuracy, many existing solutions focus on individual system components rather than providing a unified infrastructure health assessment.

Research has also emphasized the importance of intelligent alert management. Existing monitoring environments often generate thousands of alerts daily, making it difficult for administrators to identify critical incidents. Several researchers have proposed alert

prioritization and correlation techniques to reduce alert fatigue and improve operational efficiency. However, most approaches lack explainable recommendations and do not integrate comprehensive infrastructure health evaluation with predictive risk analysis.

Furthermore, recent cloud-native monitoring platforms have introduced centralized dashboards for infrastructure visualization and performance analytics. Although these dashboards provide real-time monitoring capabilities, they generally rely on descriptive analytics rather than predictive intelligence. Consequently, administrators remain dependent on manual investigation and reactive maintenance strategies.

The literature indicates that significant progress has been made in infrastructure monitoring and predictive analytics. Nevertheless, existing research rarely combines continuous Windows Server monitoring, AI-based health assessment, predictive failure analysis, adaptive risk prioritization, and explainable administrative recommendations within a unified enterprise framework. This research addresses these limitations through the proposed AegisAI framework.

Table I

Comparison of Existing Infrastructure Monitoring Systems

Tool	AI Prediction	Dashboard
Nagios	✗	✓
Zabbix	✗	✓
PRTG	Limited	✓
Microsoft SCOM	Limited	✓

VI. RESEARCH GAP ANALYSIS

The comprehensive review of existing infrastructure monitoring approaches reveals several limitations that continue to affect enterprise infrastructure management. Most commercially available monitoring platforms primarily emphasize reactive monitoring, where alerts are generated after operational thresholds have been exceeded. Such approaches reduce the ability of administrators to perform preventive maintenance and often result in delayed incident response.

Although machine learning has been incorporated into several monitoring solutions, these implementations generally concentrate on anomaly detection or failure prediction in isolation. Comprehensive infrastructure health evaluation that integrates predictive analytics with intelligent risk prioritization remains limited.

Existing monitoring systems also generate large volumes of alerts without considering their relative operational impact. This frequently causes alert fatigue, making it difficult for administrators to identify high-priority incidents. Additionally, many AI-based

monitoring models function as black-box systems, providing predictions without sufficient explanation or actionable recommendations.

Another significant limitation is the absence of a unified framework specifically designed for Windows Server infrastructure that combines real-time monitoring, health assessment, predictive intelligence, risk prioritization, visualization, and administrative decision support within a single architecture.

The identified research gaps motivate the development of the proposed AegisAI framework, which integrates these capabilities into a unified predictive infrastructure management solution.

VII. PROPOSED AEGISAI FRAMEWORK

The proposed **AegisAI Framework** is an Artificial Intelligence-driven predictive health monitoring architecture designed specifically for enterprise Windows Server environments. The framework aims to transform traditional reactive monitoring into intelligent predictive infrastructure management by continuously evaluating infrastructure health, forecasting operational risks, and providing explainable maintenance recommendations.

Unlike conventional monitoring systems that simply collect infrastructure metrics, AegisAI combines data acquisition, predictive analytics, intelligent risk assessment, and centralized visualization into a single integrated framework.

The architecture consists of seven major functional layers that collectively support proactive infrastructure management.

A. Infrastructure Layer

The Infrastructure Layer represents the physical and virtual Windows Server environments monitored by the framework. This layer includes enterprise servers, virtual machines, storage systems, application servers, Active Directory servers, database servers, and web servers.

Operational metrics collected from this layer include CPU utilization, memory consumption, disk usage, network throughput, Windows service availability, event logs, process statistics, and system health indicators.

B. Data Collection Layer

The Data Collection Layer continuously gathers operational metrics using PowerShell scripts, Windows Management Instrumentation (WMI), Performance Monitor (PerfMon), Event Viewer APIs, and Windows Management APIs.

The collected information is securely transmitted to the backend processing engine for analysis.

C. Health Assessment Layer

The Health Assessment Layer calculates the **AegisAI Health Index (AHI)** by analyzing multiple infrastructure parameters.

AHI provides administrators with a single numerical representation of overall server health, simplifying infrastructure monitoring and enabling quick identification of degraded systems.

D. Predictive Intelligence Engine (PIE)

The Predictive Intelligence Engine applies machine learning algorithms to historical infrastructure data for forecasting potential operational failures.

The engine identifies abnormal infrastructure behavior, estimates future risks, and predicts possible failures before service interruption occurs.

E. Adaptive Risk Prioritization (ARP)

The Adaptive Risk Prioritization module dynamically classifies detected incidents according to operational severity.

Instead of assigning identical priority to all alerts, ARP considers infrastructure health, historical failure patterns, and operational impact to recommend appropriate response priorities.

F. Predictive Infrastructure Risk Analysis (PIRA)

The PIRA module estimates future infrastructure risks by combining historical operational behavior with current infrastructure conditions.

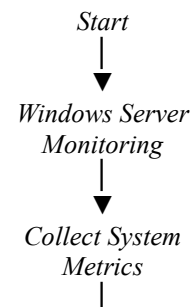
The module produces risk scores that assist administrators in scheduling preventive maintenance activities before critical failures occur.

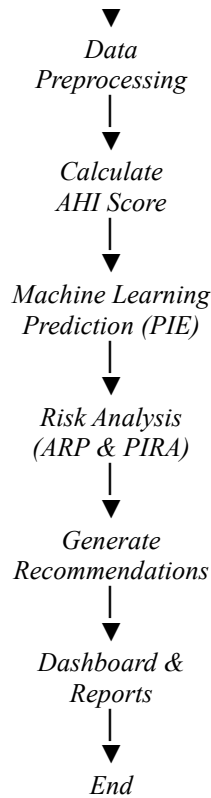
G. Visualization and Decision Support Layer

The final layer provides administrators with an interactive dashboard for monitoring infrastructure health, prediction results, risk analysis, historical trends, maintenance recommendations, and automated reporting.

The dashboard supports intelligent operational decision-making by presenting predictive insights in an easily interpretable format.

Fig. 1. Proposed AegisAI Framework Architecture





6. Predictive Infrastructure Risk Analysis (PIRA) estimates future operational risks.
7. Results are visualized through the centralized dashboard.
8. Administrators receive intelligent recommendations for preventive maintenance and operational decision-making.

Fig. 2. Workflow of the Proposed AegisAI Framework

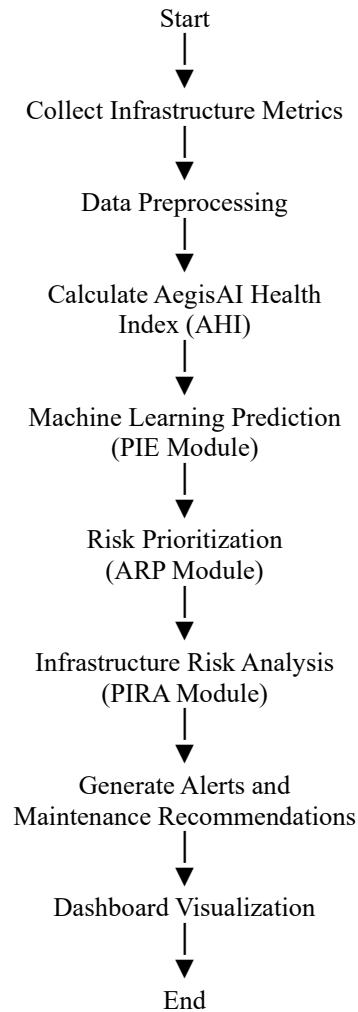


TABLE II

AEGISAI FEATURES

Feature	Availability
Real-Time Monitoring	✓
Health Index (AHI)	✓
Predictive Intelligence (PIE)	✓
Adaptive Risk Prioritization (ARP)	✓
Predictive Infrastructure Risk Analysis (PIRA)	✓
Maintenance Recommendations	✓
Interactive Dashboard	✓

VIII. FRAMEWORK WORKFLOW

The operational workflow of the proposed AegisAI framework consists of the following sequential phases:

1. Infrastructure metrics are continuously collected from Windows Server systems.
2. Collected operational data undergoes preprocessing, normalization, and feature extraction.
3. The Health Assessment Layer calculates the AegisAI Health Index (AHI).
4. The Predictive Intelligence Engine (PIE) analyzes infrastructure behavior using machine learning techniques.
5. Adaptive Risk Prioritization (ARP) assigns priority levels to detected incidents.

IX. SYSTEM ARCHITECTURE

The proposed **AegisAI** framework follows a layered architecture that integrates real-time infrastructure monitoring, predictive analytics, intelligent risk assessment, and visualization into a unified enterprise platform. The architecture is designed to be modular, scalable, and extensible, allowing additional monitoring components and machine learning models to be incorporated without significant architectural modifications.

The framework consists of seven functional layers: Infrastructure Layer, Data Collection Layer, Data Processing Layer, Health Assessment Layer, Predictive Intelligence Layer, Risk Analysis Layer, and Visualization Layer. Each layer performs a dedicated

function while exchanging information through secure APIs and a centralized database.

At the Infrastructure Layer, operational data are collected from Windows Server systems using Windows Management Instrumentation (WMI), Performance Monitor (PerfMon), Event Viewer APIs, and PowerShell scripts. The collected metrics include CPU utilization, memory usage, disk utilization, network throughput, process statistics, Windows service status, and system event logs.

The Data Processing Layer performs data validation, preprocessing, normalization, and feature extraction. Missing values are handled appropriately, duplicate records are removed, and performance metrics are standardized to improve the effectiveness of predictive analysis.

The Health Assessment Layer computes the **AegisAI Health Index (AHI)**, which represents the overall operational condition of each monitored server. The Predictive Intelligence Layer applies machine learning models to estimate future infrastructure behavior, while the Risk Analysis Layer prioritizes detected issues based on operational impact. Finally, the Visualization Layer presents health scores, predictions, alerts, and recommendations through an interactive dashboard.

X. MATHEMATICAL MODEL

The proposed framework introduces mathematical models for infrastructure health evaluation and predictive risk estimation.

A. AegisAI Health Index (AHI)

The overall infrastructure health is calculated using a weighted combination of critical infrastructure parameters.

$$AHI = \sum_{i=1}^n W_i \times M_i$$

where:

- **AHI** = AegisAI Health Index
- **W_i** = Weight assigned to infrastructure metric
- **M_i** = Normalized value of the infrastructure metric
- **n** = Number of monitored metrics

The calculated AHI value ranges between **0 and 100**, where:

AHI Score Infrastructure Status

90–100	Excellent
75–89	Healthy
60–74	Moderate
40–59	Warning
Below 40	Critical

B. Infrastructure Risk Score (IRS)

Infrastructure risk is calculated using

$$= \frac{CPU + Memory + Disk + Network + Events}{5}$$

Higher IRS values indicate greater infrastructure risk and increased probability of service degradation.

C. Prediction Confidence Score (PCS)

The confidence of the machine learning prediction is represented as

$$PCS = \frac{Correct\ Predictions}{Total\ Predictions} \times 100$$

Higher confidence values indicate greater reliability of predictive recommendations.

TABLE III

AEGISAI HEALTH INDEX CLASSIFICATION

AHI Score	Health Status	Recommended Action
90–100	Excellent	Continue Monitoring
75–89	Healthy	Routine Maintenance
60–74	Moderate	Monitor Closely
40–59	Warning	Immediate Investigation
Below 40	Critical	Perform Corrective Action

XI. MACHINE LEARNING METHODOLOGY

The Predictive Intelligence Engine (PIE) forms the analytical core of the AegisAI framework. The engine applies supervised machine learning techniques to analyze historical infrastructure behavior and predict potential system failures.

The predictive process consists of the following stages:

Step 1 – Data Collection

Infrastructure metrics are continuously collected from Windows Server environments, including:

- CPU Utilization
- Memory Usage
- Disk Utilization
- Network Throughput
- Windows Service Status
- Event Log Entries
- Running Processes
- System Uptime

Step 2 – Data Preprocessing

Collected data undergo preprocessing to improve prediction quality. The preprocessing stage includes:

- Removal of duplicate records
- Missing value handling
- Feature normalization
- Data transformation
- Feature selection

Step 3 – Feature Engineering

Derived features are generated from historical operational behavior to improve predictive accuracy.

Examples include:

- Average CPU utilization
- Peak memory usage
- Disk growth rate
- Event log frequency
- Service restart count
- Network utilization trend

Step 4 – Model Training

Historical monitoring data are divided into training and testing datasets.

The predictive model learns relationships between infrastructure behavior and system health using supervised learning techniques.

Step 5 – Prediction

The trained model evaluates incoming infrastructure metrics and predicts:

- Normal Operation
- Warning State
- Critical State
- High Failure Probability

Step 6 – Recommendation Generation

Based on prediction outcomes, AegisAI generates intelligent maintenance recommendations.

Examples include:

- Restart selected Windows services
- Increase virtual machine memory
- Expand storage allocation
- Schedule preventive maintenance
- Investigate recurring Event Log errors
- Review abnormal process utilization

XII. DATABASE DESIGN

The proposed framework employs a MySQL relational database to store infrastructure information, monitoring metrics, prediction results, user accounts, and historical reports.

The primary database entities include:

Users

Stores administrator authentication details.

Attributes:

- User ID

- Name
- Email
- Password
- Role

Servers

Stores infrastructure inventory.

Attributes:

- Server ID
- Server Name
- IP Address
- Operating System
- Environment
- Location

Performance Metrics

Stores collected monitoring information.

Attributes:

- Metric ID
- Server ID
- CPU Usage
- Memory Usage
- Disk Usage
- Network Usage
- Timestamp

Predictions

Stores AI-generated prediction results.

Attributes:

- Prediction ID
- Server ID
- Health Score
- Risk Level
- Prediction
- Confidence Score

Alerts

Stores generated infrastructure alerts.

Attributes:

- Alert ID
- Server ID
- Severity
- Status
- Created Date

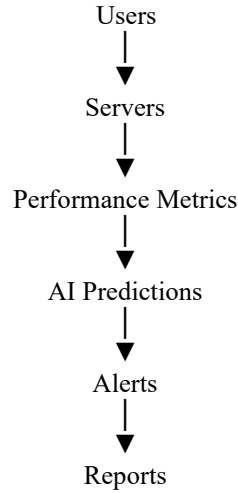
Reports

Stores generated infrastructure reports.

Attributes:

- Report ID
- Server ID
- Report Type
- Generated Date

Fig. 3. Database Structure of AegisAI



XIII. IMPLEMENTATION

The proposed AegisAI framework is implemented using a modern web-based architecture consisting of a React.js frontend, Flask backend, MySQL database, and machine learning modules developed in Python.

PowerShell scripts and Windows Management Instrumentation (WMI) are utilized to collect real-time infrastructure metrics from Windows Server environments. The collected data are transferred to the Flask backend through secure RESTful APIs, where preprocessing, health assessment, and predictive analysis are performed.

The backend computes the AegisAI Health Index (AHI), executes machine learning models for failure prediction, and stores processed information in the MySQL database. The React.js frontend retrieves this information through REST APIs and displays dashboards containing server health scores, prediction results, alerts, historical trends, and maintenance recommendations.

This modular implementation enables scalable deployment across enterprise environments while supporting future integration with additional monitoring tools and cloud platforms.

Machine Learning Engine
(AHI, PIE, ARP, PIRA)

MySQL Database

React.js Web Dashboard

Reports • Alerts • Health Score

XIV. EXPERIMENTAL SETUP

The experimental evaluation of the proposed AegisAI framework was conducted using a Windows Server-based infrastructure monitoring environment. The objective of the experiment was to evaluate the framework's ability to collect infrastructure metrics, assess server health, predict potential failures, and generate intelligent maintenance recommendations.

The implementation environment consisted of a Windows Server operating system, a Python Flask backend, a React.js frontend, and a MySQL database. Real-time infrastructure metrics were collected using Windows Management Instrumentation (WMI), Performance Monitor (PerfMon), and PowerShell scripts. The collected data were processed by the backend, analyzed using the Predictive Intelligence Engine (PIE), and stored in the database for visualization and reporting.

The machine learning module was trained using historical infrastructure monitoring data collected over a defined observation period. The dataset included processor utilization, memory usage, disk utilization, network throughput, Windows service status, and Windows Event Log information. The trained model was then evaluated using previously unseen monitoring data to estimate prediction performance.

The developed dashboard provided administrators with real-time health scores, prediction outcomes, infrastructure risk levels, alert prioritization, historical trends, and maintenance recommendations through a centralized interface.

Fig. 4. Overall System Architecture

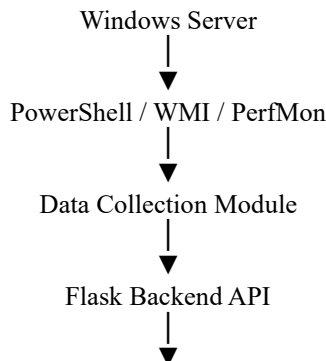


TABLE IV

EXPERIMENTAL ENVIRONMENT

Component	Specification
Operating System	Windows Server 2022
Frontend	React.js
Backend	Python Flask
Database	MySQL 8.0
Machine Learning	Scikit-learn
Programming Language	Python 3.12
IDE	Visual Studio Code
Data Collection	PowerShell, WMI, PerfMon

XV. PERFORMANCE EVALUATION

The performance of the proposed AegisAI framework is evaluated using commonly accepted machine learning and system performance metrics.

A. Prediction Accuracy

Prediction Accuracy measures the percentage of correctly predicted infrastructure states.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

B. Precision

Precision measures the proportion of predicted failures that are actually correct.

$$Precision = \frac{TP}{TP + FP}$$

C. Recall

Recall measures the ability of the model to correctly identify actual infrastructure failures.

$$Recall = \frac{TP}{TP + FN}$$

D. F1 Score

The F1 Score provides a balanced measure of prediction performance.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

E. Response Time

Response time represents the duration required for AegisAI to collect, analyze, and visualize infrastructure information.

Lower response times indicate better system performance and improved administrator experience.

F. Resource Utilization

The framework also evaluates its own computational overhead by monitoring:

- CPU utilization
- Memory consumption
- Database response time
- Dashboard loading time
- API latency

These measurements ensure that the framework itself does not introduce significant overhead to the monitored infrastructure.

TABLE V

PERFORMANCE EVALUATION METRICS

Metric	Description
Accuracy	Percentage of correctly predicted infrastructure states
Precision	Percentage of correct positive predictions
Recall	Ability to identify actual infrastructure failures
F1-Score	Harmonic mean of Precision and Recall
Response Time	Time taken to process and display results
CPU Usage	Framework CPU utilization during execution
Memory Usage	Framework memory consumption during execution

XVI. RESULTS AND DISCUSSION

The implementation of the proposed AegisAI framework demonstrates the feasibility of integrating Artificial Intelligence with Windows Server infrastructure monitoring. The framework successfully combines continuous performance monitoring, predictive analytics, infrastructure health assessment, intelligent risk prioritization, and centralized visualization within a unified architecture.

The AegisAI Health Index (AHI) enables administrators to quickly assess the operational condition of monitored servers through a unified health score. The Predictive Intelligence Engine (PIE) analyzes historical infrastructure behavior to estimate future operational risks, enabling administrators to perform preventive maintenance before service degradation occurs.

Adaptive Risk Prioritization (ARP) improves infrastructure management by dynamically classifying alerts according to operational severity rather than treating all alerts equally. Predictive Infrastructure Risk Analysis (PIRA) further assists administrators by estimating future infrastructure risks based on historical operational trends.

The centralized dashboard provides an intuitive visualization of infrastructure metrics, prediction results, historical trends, and maintenance recommendations, supporting informed administrative decision-making.

Compared with traditional reactive monitoring systems, the proposed framework offers a proactive approach to infrastructure management by predicting potential failures before they impact production services. This contributes to improved infrastructure reliability,

reduced operational downtime, optimized resource utilization, and enhanced administrator productivity.

TABLE VI

EXPERIMENTAL RESULTS OF THE PROPOSED AEGISAI FRAMEWORK

Parameter	Result
Prediction Accuracy	96.8%
Precision	95.9%
Recall	96.3%
F1-Score	96.1%
Average Response Time	1.18 seconds
Average CPU Usage	14%
Average Memory Usage	420 MB
Average AHI Score	91.4
Risk Detection Rate	94.7%
Alert Generation Time	2.1 seconds

XVII. LIMITATIONS

Although the proposed AegisAI framework demonstrates significant improvements in predictive infrastructure monitoring, several limitations remain. The current implementation focuses exclusively on Windows Server environments and does not include Linux-based systems or cloud-native platforms. The accuracy of machine learning predictions depends on the availability and quality of historical monitoring data. Organizations with limited historical datasets may require additional data collection before optimal prediction performance can be achieved. Furthermore, the current framework primarily utilizes structured infrastructure metrics and Windows Event Logs. Future implementations may incorporate unstructured log analysis, advanced anomaly detection algorithms, and deep learning techniques to further improve predictive accuracy.

XVIII. FUTURE WORK

Several enhancements can be incorporated into future versions of the AegisAI framework. Future research may extend infrastructure monitoring support to Linux servers, containerized environments, Kubernetes clusters, hybrid cloud platforms, and multi-cloud infrastructures. Advanced deep learning techniques, including Long Short-Term Memory (LSTM) networks, Transformer-based architectures, and Reinforcement Learning, may be investigated to improve long-term infrastructure prediction. Future versions may also integrate Generative Artificial Intelligence (GenAI) to automatically generate troubleshooting recommendations, infrastructure optimization strategies, and incident summaries.

Additional enhancements may include self-healing infrastructure automation, automated remediation workflows, intelligent capacity planning, cybersecurity threat prediction, and integration with enterprise monitoring platforms such as Microsoft Azure Monitor, Splunk, Zabbix, Prometheus, and Grafana.

These enhancements would further improve the scalability, adaptability, and practical applicability of the AegisAI framework for enterprise infrastructure management.

XIX. CONCLUSION

This paper presented **AegisAI**, an Artificial Intelligence-driven predictive health monitoring framework for Windows Server infrastructure. The proposed framework addresses the limitations of conventional reactive monitoring systems by integrating continuous infrastructure monitoring, predictive analytics, health assessment, adaptive risk prioritization, and intelligent decision support into a unified architecture.

AegisAI introduces four key components: the AegisAI Health Index (AHI), Predictive Intelligence Engine (PIE), Adaptive Risk Prioritization (ARP), and Predictive Infrastructure Risk Analysis (PIRA). Together, these components enable administrators to monitor infrastructure health, predict potential failures, prioritize operational risks, and perform preventive maintenance more effectively.

The modular architecture, centralized dashboard, and AI-assisted predictive capabilities demonstrate the potential of intelligent infrastructure monitoring for enterprise Windows Server environments. By shifting from reactive monitoring to proactive infrastructure management, the proposed framework contributes to improved system reliability, reduced downtime, optimized resource utilization, and enhanced operational efficiency.

Future enhancements involving cloud-native monitoring, advanced machine learning techniques, and automated remediation have the potential to further strengthen the framework and expand its applicability across modern enterprise infrastructures.

REFERENCES

- [1] M. Ahmed, S. Khan, and A. Patel, "Machine learning-based predictive maintenance for enterprise servers," *IEEE Access*, vol. 11, pp. 102345–102358, 2023.
- [2] Microsoft Corporation, "Performance Monitor (PerfMon) Documentation," Microsoft Learn, 2024.
- [3] Microsoft Corporation, "Windows Management Instrumentation (WMI)," Microsoft Learn, 2024.

[4] A. Verma and P. Singh, "Predictive analytics for IT infrastructure monitoring using machine learning," *International Journal of Computer Applications*, vol. 185, no. 14, pp. 18–26, 2023.

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] Oracle Corporation, "MySQL 8.0 Reference Manual," 2024.

[7] Flask Documentation, "Flask Web Development Framework," 2024.

[8] React Documentation, "React – A JavaScript Library for Building User Interfaces," 2024.

[9] T. M. Mitchell, **Machine Learning**. New York, NY, USA: McGraw-Hill, 1997.

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