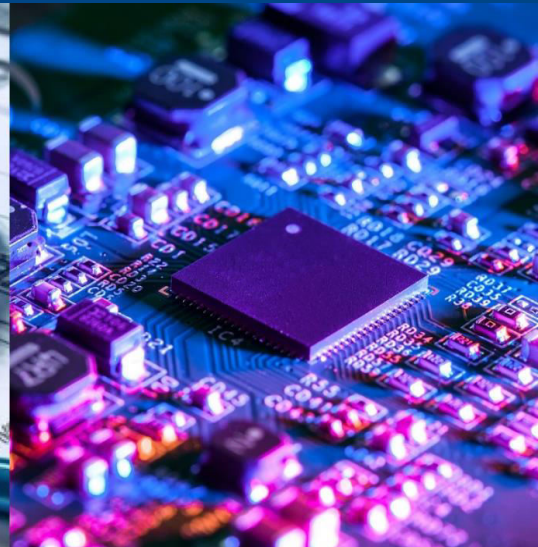
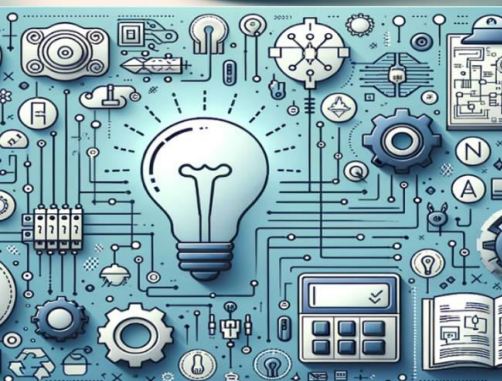




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.864

Volume 9, Issue 9, September 2026



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

SYRA: An Intelligent Windows Desktop Assistant for System Diagnosis

P. Kanchana Devi¹, Kaviyalakshmi M², SaiShambhavi Yadav G.V³, Sahitha Begum K⁴

Assistant Professor, Dept. of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College,
Madagadipet, Puducherry, India¹

UG Student, Dept. of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College,
Madagadipet, Puducherry, India^{2,3,4}

ABSTRACT: Modern Windows computers generate telemetry in form of CPU usage, memory, disk activity, network behavior, running processes, GPU load and system events. Traditional monitoring tools simply show this telemetry without any explanation of the cause and necessary action. This paper proposes SYRA, Windows desktop assistant for diagnosing and solving system performance issues. SYRA gathers telemetry and stores its snapshots with timestamps in the local SQLite database. An LSTM autoencoder identifies anomalies, while diagnostic rules, signal correlation, graph reasoning and optionally supervised prediction provide probable root cause identification. The diagnosis includes confidence score, evidence and the responsible process, if any. Any fix is never applied automatically. SYRA suggests some action to perform, waits for user approval, verifies its outcome based on telemetry data and records the incident with feedback. The system uses Electron, React, FastAPI, SQLite and TensorFlow/Keras. Functional testing confirmed that the whole workflow works as expected. For example, CPU usage dropped from 41.1% to 12.4% due to an approved process-priority modification.

KEYWORDS: System Monitoring, Root Cause Analysis, Anomaly Detection, LSTM Autoencoder, Guided Remediation

I. INTRODUCTION

Modern Windows computers run applications, background services and system processes simultaneously and generate telemetry in terms of CPU, memory, disk, network, processes, GPU and events. People typically use Task Manager and Resource Monitor, but the cause identification usually requires multiple signals combination.

Another problem is how to handle the problem after it has been found. Modifying processes and system components can potentially interfere with work and bring even more problems. The system which analyzes telemetry data, provides probable cause explanation and fixes suggestion can significantly assist a user without compromising the control. Modern approaches to anomaly detection and AIOps target mostly cloud and enterprise environments rather than desktops.

SYRA solves this problem by providing Windows telemetry collection, anomaly detection, reasoning about probable root cause, reporting confidence and evidence and user-approved solution with verification.

II. LITERATURE SURVEY

Windows offers Task Manager, Resource Monitor, Performance Monitor and Reliability Monitor for resource monitoring. These utilities allow to discover heavy resource consuming processes, but mostly show raw data without explaining the cause.

Enterprise solutions like Intune, ManageEngine and NinjaOne gather data and can apply fixes, but are targeted at IT specialists. Observability platforms gather metrics, logs, traces and events, but are focused mostly on servers and cloud-based systems.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The literature shows a transition from threshold check methods to anomaly detection and AI-based root cause analysis. Endpoint diagnosis also faces issues. While numerous tools detect the high resource consumption process, they fail to explain the reason for this. Some analyze the data separately, others don't consider risk assessment and remediation advice while very few verify if the fix helped. SYRA tries to combine these elements while keeping the user in control.

III. METHODOLOGY

SYRA applies a pipeline which consists of telemetry collection, anomaly detection, root cause reasoning, user-approved remediation and verification. SYRA runs the monitoring agent using Python FastAPI backend, while the Electron and React interface displays the results, as shown in Figure 1.

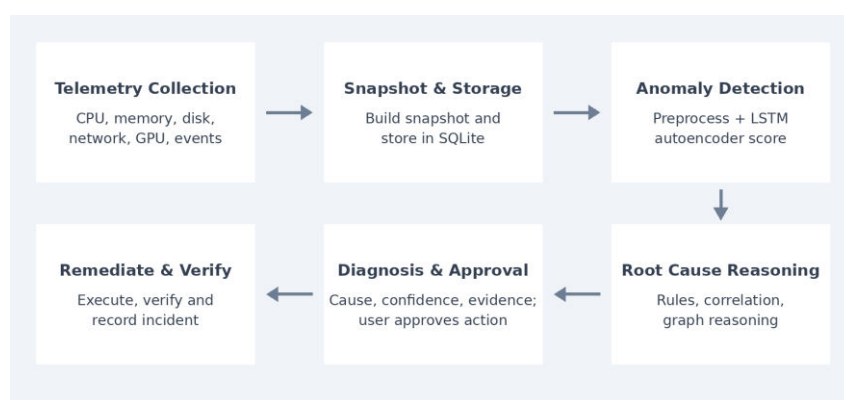


Figure 1: System Workflow

A. Telemetry Collection and Storage

The background agent continuously gathers CPU, memory, disk, network, process, GPU and Windows events data. Readings are then aggregated into snapshots and saved to SQLite database using SQLAlchemy models. Database holds system metrics, incidents and anomalies, root cause analysis results, remediations and user feedback.

B. Data Preprocessing and Anomaly Detection

Each snapshot is converted to a feature vector and recent vectors are kept as sequence. SYRA uses LSTM autoencoder for anomaly detection. This model learns how to reconstruct the input telemetry sequence. The gap between reconstruction and input serves as an anomaly score and helps decide whether the behavior is abnormal or not. The model training is done offline.

C. Root Cause Reasoning

Once the anomaly is detected, the diagnostic rules find the symptom, and correlation engine connects related signals. Then the reasoning engine builds the incident graph and rates the possible causes based on graph connections, evidence available and anomaly data. There is an option of applying supervised predictor to make the diagnosis. Rule and graph-based results are preserved unless there is a strong support in incident graph for the prediction.

D. Diagnosis, Remediation and Verification

The resulting diagnosis includes the probable cause, confidence score, evidence and affected process. If remediation is needed, SYRA generates remediation proposal with action and target. SYRA will never perform any action without user's approval. Then, SYRA compares telemetry readings before and after the action and verifies the improvement, recording the result.

E. Implementation and Incident History

Interface is implemented using React, TypeScript and Vite with Electron packaging. The backend uses Python and FastAPI. System metrics are retrieved from psutil and Windows events on local machine.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. RESULTS AND DISCUSSION

SYRA was tested as a Windows desktop application with FastAPI backend and Electron/React frontend. Evaluation concentrated on monitoring, diagnosis, remediation approval, execution, verification and incident history recording. It wasn't a formal accuracy or performance evaluation but a functional one.

In one captured state CPU usage was around 18.9%, memory usage – 83.6%, disk usage – 7.1% and network throughput – 0.02 Mbps. The diagnosis revealed excessive swapping as the probable root cause.

In another state, SYRA diagnosed CPU bottleneck with the process electron.exe and proposed to lower the priority of this process. User approval was required (Figure 2).

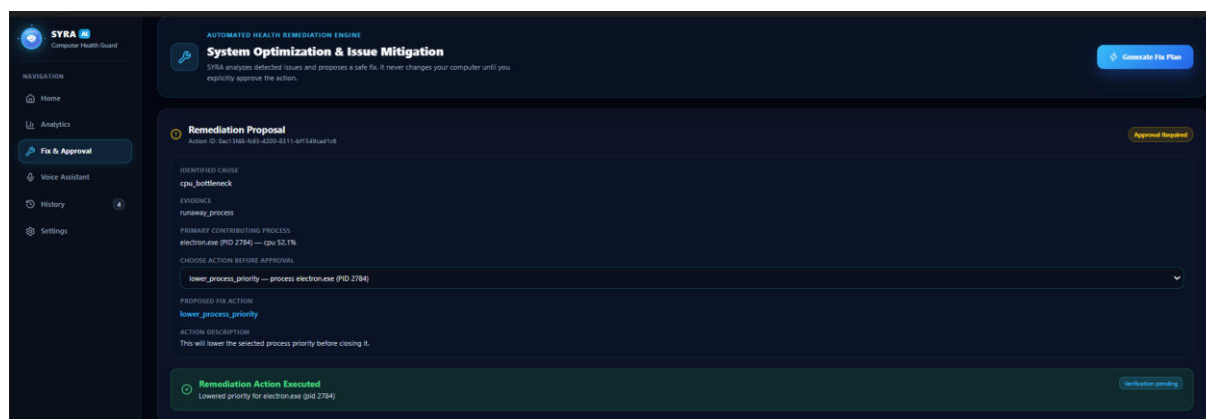


Figure 2: Remediation proposal with cause, evidence, target process and approval requirement

After the approved action, CPU usage dropped from 41.1% to 12.4%, memory usage changed from 79.6% to 79.5% and disk usage stayed at 7.1% (Figure 3). All seven evaluated functions worked correctly.

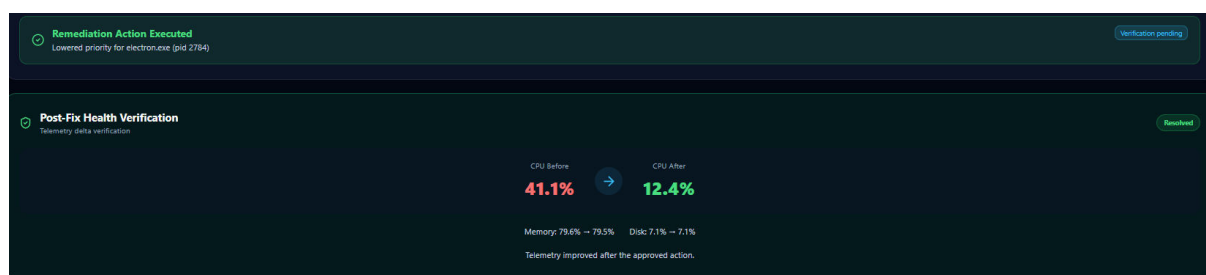


Figure 3: Post-fix health verification before and after the approved remediation

The results show that all parts of SYRA work as one workflow. However, the example with CPU reduction can not be used as a performance benchmark. These tests do not prove model accuracy, generalization and performance on all Windows machines. Anomalies and root cause test cases with labels are required to evaluate anomaly and root cause accuracy. The diagnosis quality depends on collected data, rules, model files and thresholds, while remediation quality depends on system state and action chosen.

V. CONCLUSION

SYRA is a Windows desktop assistant for monitoring system performance and aiding the user in diagnosing the problem. SYRA gathers data on CPU, memory, disk, network, processes, GPU and events and then diagnoses those data via an LSTM autoencoder, diagnostic rules, signal correlation and graph-based reasoning. The diagnoses come



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

with evidence and the guilty process, if any. Any fix suggested by the system needs to be approved by the user, who will then verify its effect.

The experiment showed that the entire process from monitoring to fixing, approval, and verification to logging of the incident was feasible. For future research, we plan to evaluate the system for anomaly detection and root cause reasoning on labeled data sets, improving the models based on the feedback and controlled retraining. In addition, Windows-specific signals and fixes, extended verification period and better monitoring intervals, alert and data retention parameters need to be evaluated.

REFERENCES

- [1] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," Proc. 2017 ACM SIGSAC Conf. Computer and Communications Security, pp. 1285–1298, 2017, doi: 10.1145/3133956.3134015.
- [2] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An Online Log Parsing Approach with Fixed Depth Tree," 2017 IEEE Int. Conf. Web Services (ICWS), pp. 33–40, 2017, doi: 10.1109/ICWS.2017.13.
- [3] H. Mi, H. Wang, Y. Zhou, M. R. Lyu, and H. Cai, "Toward Fine-Grained, Unsupervised, Scalable Performance Diagnosis for Production Cloud Computing Systems," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1245–1255, 2013, doi: 10.1109/TPDS.2013.21.
- [4] W. Gu, R. Zhong, G. Yu, X. Sun, J. Liu, Y. Huo, Z. Chen, J. Zhang, J. Gu, Y. Yang, and M. R. Lyu, "KPIRoot+: An Efficient Integrated Framework for Anomaly Detection and Root Cause Analysis in Large-Scale Cloud Systems," Empirical Software Engineering, vol. 31, no. 2, Art. no. 28, 2026, doi: 10.1007/s10664-025-10769-0.
- [5] A. Aziz and K. Munir, "Anomaly Detection in Logs Using Deep Learning," IEEE Access, vol. 12, pp. 176124–176135, 2024, doi: 10.1109/ACCESS.2024.3506332.
- [6] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang, Z. Wu, X. Hu, P. S. Yu, and Y. Li, "A Survey of AIOps in the Era of Large Language Models," ACM Computing Surveys, vol. 58, no. 2, Art. no. 44, pp. 1–35, 2025, doi: 10.1145/3746635.
- [7] Microsoft, "About Performance Counters," Microsoft Learn.
- [8] Microsoft, "Using Performance Counters," Microsoft Learn.
- [9] Microsoft, "Troubleshoot Performance Problems in Windows," Microsoft Learn.
- [10] Microsoft, "Logging Guidelines," Microsoft Learn.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com