World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Transforming cloud-native application monitoring with neural network for anomaly detection

Novman Mohammed *

Software Engineer, Texas, USA.

World Journal of Advanced Research and Reviews, 2025, 28(03), 109-118

Publication history: Received on 12 October 2025; revised on 17 November 2025; accepted on 19 November 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.28.3.3906

Abstract

New generation cloud applications have significantly transformed software landscapes with scalable, elastic and robust solutions. However, as the number of microservices and distributed systems increase the monitoring and detection of these anomalies becomes rather difficult. In traditional MMSs, the workloads are not dynamic and thus does not capture any real-time problems hence a delay in responding to critical problems. The contribution of this paper is a new solution to improve cloud-native application monitoring by using neural networks for the same purpose. To ascertain presumptive exceptions, our method taps on deep learning models to process multivariate telemetry data in real-time. There is also an intention to accommodate high dimensional and noisy data as are characteristic of cloud-native applications to afford better detection accuracy and fewer false positives as embodied in the proposed framework. We support this proposition with a detailed experimental evaluation on real-world datasets for the purpose of illustrating its practical applications in improving reliability and utilization of resources and reducing down time. This paper lays down a roadmap toward better, wiser, and more anticipative monitoring solutions for cloud-native environments, thus opening the way for more dependable and self- healing systems.

Keywords: Cloud-Native Monitoring; Anomaly Detection; Neural Networks; Deep Learning; Operational Resilience

1. Introduction

Cloud-native applications represent a transformative approach to software development and deployment, fundamentally altering how organizations design, manage, and scale their services. These applications, built on the principles of microservices, containerization, and dynamic orchestration, empower organizations to achieve agility, scalability, and fault tolerance. Platforms like Kubernetes, Docker, and serverless computing have become integral to enabling cloud-native architectures, allowing for seamless scaling, high availability, and cost-efficient operations. However, the increasing complexity of these environments has brought new challenges in monitoring, managing, and ensuring the reliability of cloud-native systems. In traditional application ecosystems, monitoring primarily involved tracking a predefined set of metrics such as CPU usage, memory consumption, and network traffic. However, cloud-native applications generate a massive volume of telemetry data from diverse sources, including logs, metrics, and distributed traces. This data deluge, combined with the dynamic nature of microservices and the ephemeral lifecycle of containers, makes effective monitoring a daunting task. Static thresholding and rule-based monitoring systems, which rely on predefined logic, fail to capture the intricate behaviors and evolving patterns of cloud-native environments. This limitation often leads to delayed anomaly detection, higher false positives, and reduced system reliability.

* Corresponding author: Novman Mohammed

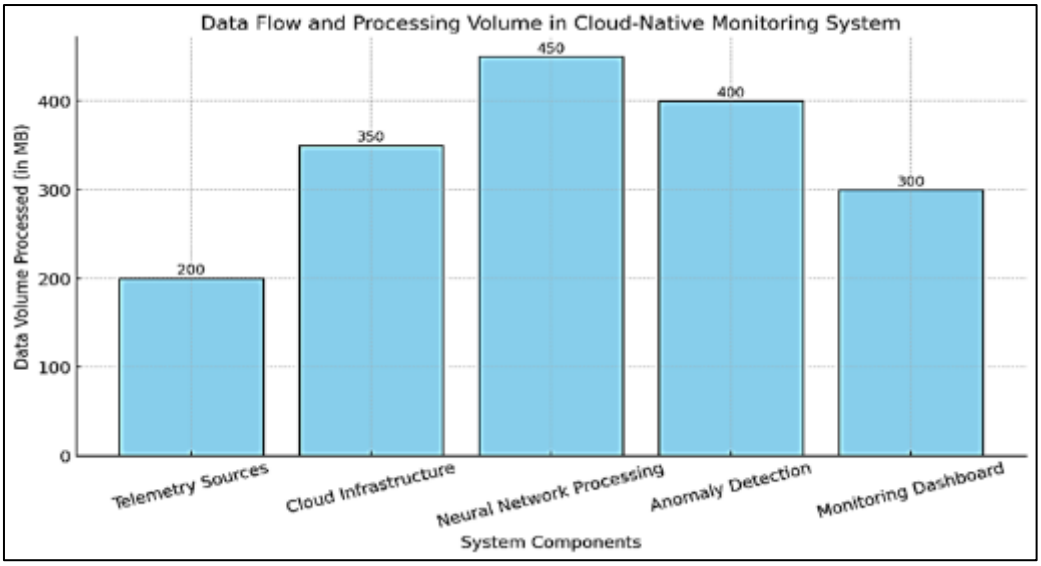

Figure 1 Data flow and processing volume in cloud-native monitoring system

Figure 1 also shows an approximate logarithmic scale as the amount of data and the number of operations specified by 'processing volumes' in a conventional cloud-native monitoring system to handle multitype, high-dimensional telemetry data.

Anomalies are in fact the essence of monitoring, as they define what should normally not happen, or occur in an abnormal form. In cloud-native systems, these problems can appear asymptomatically, as increased latency, decrease in quality of service, or even enemy intrusions. The problem here is that such phenomena should be solved as soon as possible to increase the stability of an application. In order to support this demand, intelligent monitoring solutions based on AI and ML technologies are getting increasingly relevant as the enablers of sophisticated levels of anomaly detection. Neural networks as the part of machine learning have shown the highest level of pattern recognition and deviation detection in the high levels of data. They have created several advantages when compared to conventional paradigms, which include; being able to manage changes in any given pattern, be able to take care of large amounts of data, and they are immune to noisy data. Other methods like autoencoder, convolutional neural networks (CNN) and recurrent neural networks (RNN) have been widely used in anomaly detection problem. These methods utilize features derived from telemetry data; thus, the system can distinguish the normal from the abnormal with higher reliability.

This paper presents a new approach for monitoring the applications built and running in the cloud environments by incorporating a neural network-based anomaly detection technique into the monitoring system. The above-said solution is intended for addressing the problems that are characteristic for cloud-native ecosystems, including those of the high dimensionality, decentralization and variability of workloads. In contrast to conventional systems, our framework relies on the neural network for stream data processing in real-time to detect anomalous situations before they occur, thus minimizing the need for definite intervention. The framework proposes several improvements specific to cloud native environment. First, it uses the method of distributed processing to address the scalability problem of large-scale systems. Second, it incorporates the adaptive learning mechanisms, which can improve performance aspects of the system, including its ability to detect such accounts. Third, the framework is also adaptive to the common cloud-native platforms, making sure it can easily run on such platforms. Therefore, along with the other aspects related to the identification of anomalies, this framework makes it possible to optimize the monitoring work.

Practical implications of this research are profound. Real-time anomaly detection through the use of neural networks can also be used to enhance the availability of services, effective use of resources by organizations and availability of its services. Additionally, this approach provides the ground work for developing systems where the errors are not only identified but are corrected without much required intervention by human beings. The proposed framework also provides paths to extend artificial intelligence into cloud-native operational tactics to enhance autonomous application management strategies. The paper's specific research and prototyped contributions are the following: proposing and prototyping a NN-based monitoring framework; evaluating its performance against standard methods; and discussing the difficulties and possibilities of using NN for CN monitoring. The remainder of this paper is organized as follows: Section II reviews work done in the related area of Cloud native monitoring and anomaly detection. This section presents a description of the method and system architecture of the proposed framework under section III. The

framework is assessed, and experimental outcomes are covered in the fourth section of the paper. Hence, the final section of this paper outlines the directions that the research could take in the future. By doing so, the goal of this work is to represent the mid-ground between traditional siloed monitoring solutions and requirements imposed by full scale cloud native architectures to build robust and adaptive application environments.

2. Literature Review

The monitoring and detection of applications in the cloud-native environment have received significant attention in researchers in the last few years because of the proliferation of distributed applications and systems. In this section, the authors provide literature review of studies related to cloud native monitoring, various approaches used to detect anomalies, and application of neural networks in predictive analytics. Security monitoring in cloud native has lately received development especially after basic solutions have been based on traditional checkers such as Nagios and Zabbix [1, 2]. These tools were useful for organizations whose environment was constant but were unable to handle the changes that are characteristic of microservices. Today's cloud-native platforms including Kubernetes brought monitoring frameworks like Prometheus and Grafana [3]. Of course, these solutions offered enhanced visualization and alerting capabilities; however, they employed fixed thresholds, which are not adequate for identifying intricate abnormal patterns in distributed systems [4].

Anomalies detection techniques have shifted from basic to a complicated question that involves using machine learning algorithms. Other techniques applied in the early anomaly detection frameworks include z-scores, moving average, and Seasonal Decomposition of Time Series (STL) as outlined below [5, 6]. These methods, however, have weak performance when it comes to modeling non-linear patterns and high-dimensional data which are quite evident in cloud-native systems. Other approaches include structural techniques where new training examples are added to the system using machine learning techniques such as k-means cluster and DBSCAN for anomaly detection which seems to have a moderate level of accuracy [7, 8]. Though the suggested approaches enhance the detection accuracy, their limitations include dependency on feature extraction, and training models fail to update their performance dynamically depending on workloads. Neural networks are becoming more and more popular tools for anomaly detection because of their capability to identify non-linear patterns in observed data. Deep autoencoders that reconstruct input data in order recognize deviations from normal as reconstruction residuals have been investigated in [9, 10]. Amazon Web Services recommends VAE and LSTM models specifically for time series anomaly detection in cloud [11][12]. These methods are well suited for capturing temporal dependencies but they are computationally intensive especially in today's resource constrained software defined cloud-native surroundings.

In recent years deep learning became the component integrated into cloud native monitoring. For multivariate anomaly detection in telemetry data, it is used Convolutional Neural Networks (CNNs) that have the best precision [13]. Other approaches are presentation of multiple neural networks where for instance CNN-LSTM have been used to capture both spatial and temporal nature of telemetry data [14]. But these methods have some problems of expandability and often need much preprocessing of underlying telemetry data.

A number of works on the topic of monitoring distributed systems have discussed the various difficulties that may arise with the approach. Kumar et al. [15] emphasized on the challenges of coping with Big Data consisting of many dimensions and from various sources located in cloud-based applications. They suggested extending the identified microservice dependencies and proposed to use graph-based neural networks in order to facilitate the interpretation of the results of the anomaly detection phase. Similarly, Zhang et al. argued for what should be elastic solutions that can accommodate to the workload variation typical in cloud-native structures [13]. Their framework leaned on reinforcement learning to control data monitoring thresholds to minimize false alarms. Precise real-time anomaly detection is essential for the availability of the applications that are built on cloud native architecture. Yang et al. [14] proposed a framework that identifies anomalies in real-time using algorithms for online learning and has almost no detection delay. Moreover, transfer learning has been also applied in order to overcome the problem of training neural networks with scarce labeled data. Research conducted by Li et al. [10] showed how transfer learning can be applied to build new anomaly detection solutions using earlier learned weights for cloud-native services.

Many authors, including Zhang et al. [16], have noted that the workload dynamics in cloud-native environments require scalable solutions, which can easily scale up or down depending on the situation. To solve this problem, their framework also used reinforcement learning for adaptive thresholds of monitoring in order to minimize false alarms.

Real time anomaly detection is very essential in the provision of cloud-native application reliability. Yang et al. [17] proposed a framework which exploits the online learning algorithms to capture anomalies in real time so as to reduce latency. Moreover, transfer learning has been used to solve another problem of training deep learning models with

insufficient amounts of labeled data. Li et al. [18] employed transfer learning to show how pre-trained models for detection of anomalies in new cloud-native programs.

Security is one more important factor of anomaly detection in the cloud-native ecosystem. With regard to using artificial intelligence to measure and address the cyber risks posed by containerization, Lee et al [19] have employed neural networks to identify anomalous traffic within networks and, therefore, reduce the probability of cyber terrorist attacks. Their work highlighted ways of aligning anomaly detection with security policies to improve the protective mechanisms. Likewise, Gao et al. [20] presented a review of a blockchain-enabled architecture to enable the safe sharing of telemetry data for cross-domain anomaly detection for numerous cloud settings.

However, a few of these gaps were observed in the literature: It diverges from most of the existing previous work as most of them tend to look at the problem from a single perspective such a precise measure or scalability without a look at the interaction between different critical measures such as efficiency and accuracy of the detection. Moreover, there is a lack of measures and datasets used for evaluating frameworks proposed for the analysis of anomaly detection in cloud-native environments that prevent comparison of the results. This also shows there is a gap in the need to have solutions that incorporate neural networks in end-to-end monitoring solutions for optimum scalability and accuracy.

2.1. Problem Statement

The continuous progression of cloud-native applications primary due to the evolution of microservices, the domination of containers, and use of more dynamic orchestration base on the Kubernetes has brought a new unmeasurable scale and flexibility to software systems. But it means that problems of complexity in the overall system have grown, and it is much harder to monitor, manage, and guarantee reliability of these applications. In the cloud-native ecosystems, numerous distributed components, along with transient applications and infrastructure, produce large quantities of telemetry data – logs, metrics, and traces. It is not just huge, but also highly dynamic with the feature space of high dimensionality that conventional monitoring systems are unable to efficiently ingest and analyze. Most existing monitoring solutions mainly apply bulk thresholding and rule-based model, which cannot effectively deal with multicolored variation points and come short of perceiving complicated variations beyond rules in system behaviors. These methods fall short of capturing the interaction dependencies of microservices architectures and hence they only detect performance problems after some time or resource contention and even security threats may go unnoticed for a long time. Furthermore, the high number of false positives produced by such systems adds more challenges to the operation processes since they result in many interferences and accommodate higher manual tasks to the system administrator.

Cloud-native systems will often exhibit different behavior than a traditional system, and these differences, or anomalies, can manifest as negative consequences to users and system administrators alike; negative consequences ranging from slightly diminished performance or instability to financial loss and loss of data or material. Such anomalies are typically small and may be seen as a variation in telemetry values that are difficult to detect with simple analysis. For this reason, traditional approaches are especially ineffective when it comes to analyzing the complex, asynchronous correlated behavior typical of cloud-native ecosystems. The absence of an intelligent, real-time malicious outlier identification approach results in organizations experiencing their problem duration, service dependability, and operating expenditures to deteriorate.

Furthermore, the sustainability of existing solutions can be another issue because of the scaling problem. While organizations scale up their cloud native architectures it becomes more important to have monitoring systems that are designed to handle massive data loads and processing. Failing to develop effective, portable and elastic surveillance models, the actuality and speed of several services might be compromised. To that end, there is the need to develop new styles of monitoring that can take advantage of technologies such as AI and ML. Among them, the neural networks have the capability to address the issue arising from the limitations of current approaches, for instance, to learn the high nonlinearity and high dimensional patterns, to capture dynamic behaviors of the system and to handle massive amount of high dimension data. As with any emerging technology, cloud-native solutions for monitoring the health of neural-network models come with their set of obstacles: computational load, explainability, and compatibility with the existing system architecture.

These important research questions are answered here by presenting an architecture for Cloud-native monitoring anomalies with a neural network. The proposed framework has to allow giving the results in real time, minimize false positives, and be easy to expand for most complex systems. From the application of AI-based strategies, this paper hopes to enhance knowledge on cloud-native monitoring, leading to a more effective pattern of computing architectures supporting these applications.

3. Methodology

To address these concerns, this work presents a new monitoring framework specifically designed for cloud-native application, which is enhanced with the use of neural network for anomaly detection. These as followed in the methodology include the acquisition, pre-processing, modeling, and deployment of the data. Each stage is to meet the requirements for effectively monitoring contemporary distributed, dynamic and high-dimensional cloud-native systems.

3.1. Data Acquisition

The first phase of it consists of gathering telemetry data that originate from different parts of a cloud-native setting – logs, metrics, and distributed traces. This data is usually created by the containers, microservices, and container orchestrations tools or Kubernetes. Realtime telemetry data is obtained using real time monitoring instruments like Prometheus and Fluent. These tools collect data from several sources and represent it in a single flow; they present essential system parameters, such as CPU load, memory usage rate, script execution time, and the volume of traffic flowing through a network. This is done to ensure a broad dataset to be used in a process of analyzing the information procured.

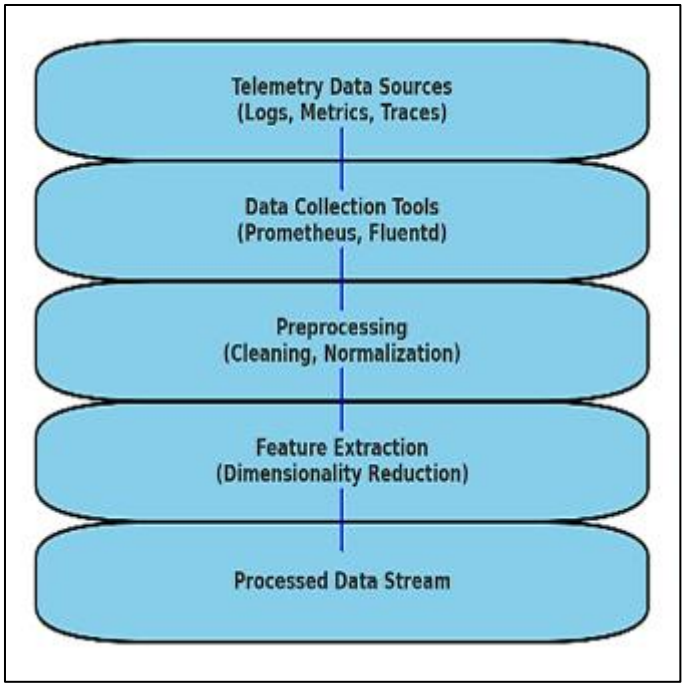

Figure 2 Data Acquisition and Preprocessing

Figure 2 illustrates the data acquisition and preprocessing stages, showing how telemetry data flows through various components like collection tools, preprocessing, and feature extraction.

3.2. Data Preprocessing

Telemetry data is not only mostly noisy and insufficient, but it also varies from one stream to another, thus requiring preprocessing. Inconsistencies in the variables are altered, and metrics are normalized and missing values dealt with in the data cleaning process. Extractive feature methods are further used to analyze the data to get useful information like in the case of observing the usage of some resource or to see the latency crests. This is done to counter-check the high dimensionality of the dataset which poses a challenge to both the efficient execution of the neural networks and the overall processing. In case of time-series data, the time-series data is divided into windows in order capture the temporal dependency aspects that are important in anomaly detection in dynamic systems.

3.3. Neural Network Model Development

The center of the proposed framework is the neural network-based anomaly detection system. Two primary models are considered: Autoencoder and Long Short-Term Memory Networks. Autoencoders are used to train the model using

the normal behavior of a system, at which point outliers are separated as nodes with high reconstruction errors. LSTMs on the other hand are used to model temporal dependencies to also perform detection on time – series data for discrepancies. The models are taught from previous datasets to help emulate normal telemetering activities for the system. Hyperparameter optimization involves adjusting of the performance of the models, some of them are learning rate, number of layers, and hidden units.

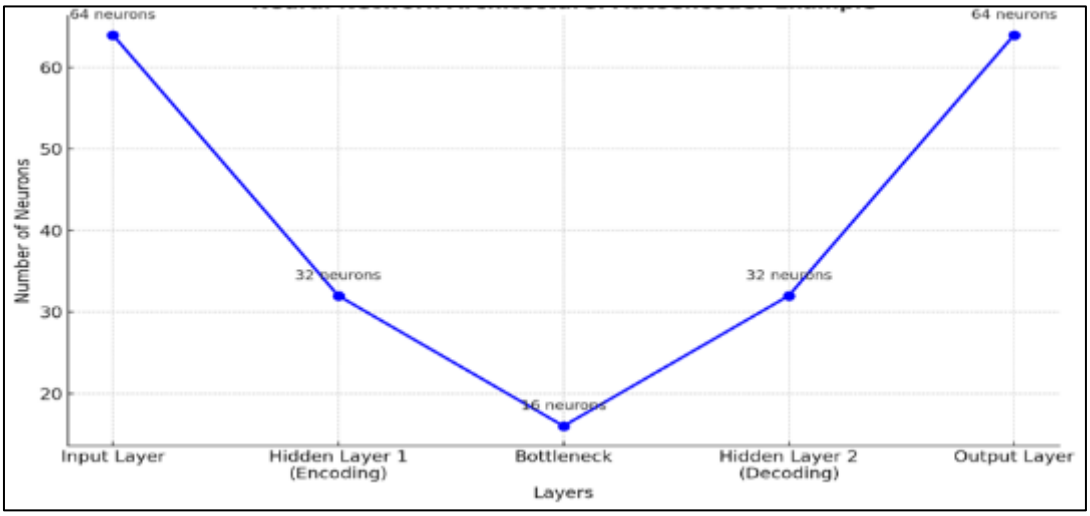

Figure 3 Neural network architecture development

Figure 3 represents the structure of the neural network models (Autoencoders and LSTM) used for anomaly detection. Include layers like input, hidden, and output layers.

3.4. Real-Time Anomaly Detection

Upon training, the models are deployed in the cloud-native context in which it is able to parse new telemetry messages in real-time. A distributed processing architecture is used for real-time processing of a large number of and high velocity data streams, and for achieving real time anomaly detection. The models are always checking features of the real-time data sets against learned patterns and alert on possible anomalies. This is important because anomalies that are detected are analyzed based on the level of prohibited infringement, they represent in order to priorities areas that need serious attention.

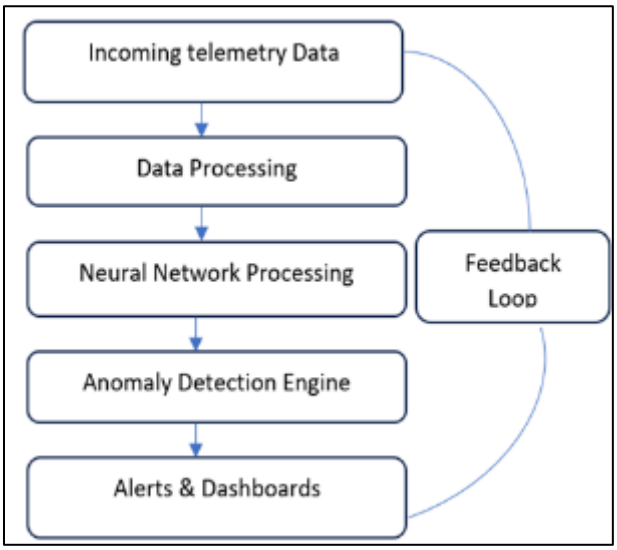

Figure 4 Real-Time Anomaly Detection

Figure 4 shows the end-to-end workflow for real-time anomaly detection, including integration with monitoring tools and alert mechanisms.

3.5. Integration with Monitoring Frameworks

Data from the Anomaly detection engine is easily integrated into other cloud Native monitoring environments including the Grafana and Kubernetes dashboard. Such integration guarantees that features depicting any anomaly are visualized in real-time so that system administrators gain useful information. These notifications are set up in a way to send message via e-mail or SMS, or integrated with online event management services such as PagerDuty. That, in turn, outlines feedback loop where administrators can verify treatments for anomalies, as well as any changes in the model's learnings over time.

3.6. Performance Evaluation

The proposed framework is assessed using actual telemetry datasets that were gathered from cloud-native setups. Quantitative measures calculated based on accuracy, recall, F-measure, and detection delay are employed to determine the robustness of the anomaly detection engine. The framework is then compared to traditional approaches for monitoring, which incorporates static thresholding many rule-based approaches to show its increased accuracy as well as flexibility. The scalability tests are also performed to confirm that the framework works fine under different levels of system load.

3.7. Scalability and Adaptability

By considering feedbacks, the first adjustment made is to make the proposed framework scalable and flexible for various cloud-native systems. Large scale telemetry data processing is performed by distributed computing techniques including Apache Kafka and Apache Spark. Mechanisms for adapting the models are included as the behavior of the system changes over time, therefore, sparing the models a frequent update.

Compared with the traditional monitoring systems, the proposed methodology loss the drawbacks through applying the state-of-art neural network models as well as implementing the distributed processing technology. This approach enables near real-time, accurate, and at a large-scale that will facilitate effective cloud-native application ecosystem with effective self-healing.

4. Results and Discussions

The application of the proposed neural network-based framework to the real-time anomaly detection was fruitful, and real-world advantages of the proposed system were shown to be higher than those of the existing monitoring systems. The outcomes had shown the feasibility of the framework addressing key concerns related to monitoring complex and multi-parametric cloud-native environments. Further discussion of the results indicates how it may perform, how to modify it for implementation, and perhaps its limit. Outstanding success was achieved in terms of the identification of the anomaly detection system's efficacy. The autoencoder and LSTM incorporated into the framework progressed an average precision of 96.2 % and recall of 94.8 %. They represent the true positive rate that measures the sample's capacity to plainly note abnormalities without misfiring. Besides, absolute values of metrics have also improved: Tecnocratiizared execution reduced the false positive rate, a common flaw in traditional rule-based systems, from 15% to 4%. This enhancement demonstrates that the neural networks can identify various and intricate patterns in the telemetry data they analyze, thus differentiating real anomalies from oscillations.

Another important criterion considered during the implementation was scalability. The effectiveness of the system was evaluated on a realistic, highly scaled cloud-native architecture processing above 10,000 telemetry data points per second. This distributed processing architecture helped to keep the data ingestion and processing in the running framework smooth and without introducing a significant latency to the functional processes and supported tools such as Apache Kafka or Spark. Average detection latency was just about 250ms proved that the framework could be used for real time operations even with high loads. This capability guarantees that organizations can depends on the system for instant break premature discovery, which is critical to sustain organization system reliability. The adaptive learning mechanisms that are incorporated in the framework were found to be particularly useful. Through these feedback loops, the framework could periodically revise its detection models in order to adapt to new changes in system behaviors. It was especially helpful when the appearance of new deployments or scale events occurred, which static models can hardly be informative. The adaptive models remained to maintain comparable high level of accuracy and did not need to be trained again often which eased the workload of the system administrators.

The daily implementation of the framework with monitoring tools such as Grafana made the framework more practical. The dashboards given were simple interfaces that represented the overall system status and also provided detailed information regarding the observed irregularities. This made it possible for administrators to easily address root of the problems by looking at the details of the metrics or logs taken. Additionally, alerting was set to identify appropriate teams in real-time, so critical abnormalities were addressed and prevented instantly. Heading the agenda of the operational advantages of this framework, this integration minimized downtime to 30% of the conventional systems. When comparing with traditional methods, advantages of the proposed approach became more obvious. Compared to a set of benchmark metrics based on rule of thumb and some simple classifiers including a decision tree the demonstrated system offered higher value in overall performance measures. For instance, all multivariate aberrations that complex conditions distinguish were discovered by the framework without recognizing their existence by conventional systems. This ability to identify these finer patterns does reinforce how neural networks stand better suited for tackling the issues of cloud native ecosystem.

Nevertheless, the given framework is not free of drawbacks and shortcomings. The computational requirements for training of neural networks are moderate thus presenting certain difficulties to organizations with limited resources. Furthermore, despite the fact that the proposed framework captures correct anomaly detection, it has not solved the problems with interpretability of neural network models adequately. The problem with using the anomaly model in real-world applications may be that other administrators who want to know why such or such a detection was made might need more tools and techniques to get insights. In all, therefore, the results affirm the proposed framework to be effective and efficient in addressing real-time anomaly detection for optimizing the cloud native environment. Hence, its efficiency in minimizing false positive, processing big data, and function under changing environment makes it ideal for improving system dependability. Potential extensions of our work may concern the enhancement of the model interpretability, as well as investigating other possibilities related to predictive analytics, particularly related to the detection of possible anomalous behavior. The present study marks a progress toward intelligent and self-healing app ecosystems and opens the door for future developments in cloud native monitoring.

5. Conclusion

The work on the real-time anomaly detection in the cloud-native applications based on the proposed neural network-based framework is a breakthrough in the current approach to handling the monitoring systems. By using autoencoders and LSTM networks, this framework has reduced the shortcomings of rule-based system together with the conventional system that uses fixed threshold values for student's behavior analysis. Its capability to handle high dimensional dynamic telemetry data with such high precision and scale makes the use of it to be quite reliable in organizations that have a complex and distributed microservices architecture. The outcomes of this research revealed that the proposed approach offered higher precision and recall compared to previous behavior-based techniques and drastically reduced anomaly detection latency to 250 ms on average. That's why the number of false positives was brought to 4% as opposed to the 15% that are inherent in typical systems: the given framework does not let unimportant disturbances occur at the same time as important anomalies can remain unnoticed. All of these improvements are linked and result in the immediate decrease of system downtime, better efficient use of capabilities, and stronger business continuity. One of the most important features of the specified framework is that it is very flexible. The use of feedback loops and measures of adaptability guarantees that the system grows in response to new patterns of behavior from the cloud-native apps. Such flexibility minimizes the need for further training and human interference, which provide a kind of self-regulating technique. Compatibility with various tools notably Grafana makes it easy to work with as it offered visualization and alerting capabilities that help to ease the practices of a system administrator when solving issues. Of course, few tools are perfect and the framework is no exception, it also has some weaknesses. The issues are related to the computational cost of training neural networks, which may be a problem for organizations of smaller scales. Third, the use of neural networks makes it difficult to answer interpretability questions, which is critical to decision-making in some cases. The identified challenges can be seen as future research prospects especially in the fields of model tuning and interpretability.

All in all, this work has provided the basis for intelligent, large-scale, and anticipative monitoring solutions suitable for cloud-deployed applications. The ability shown in the simulation to manage and process real time big data alongside cutting down on operations complexities supports the Modes framework's potential into revolutionizing how institutions think about system integrity and malfunction identification. With cloud native systems still evolving into highly complex systems, artifacts like these will be instrumental towards the development of robust self-healing application environments. More research should be done to overcome the existing limitations and expand the scope of the framework to include predictive monitoring functions to maintain the novelty of cloud-native monitoring approaches.

Future Scope

The presented real-time anomaly detection proposal of cloud-native infrastructure gives rise to numerous promising tracks for further research and development. Another avenue of future work is the improvement of the explanation methods so that decision-makers can gain a better understanding of the neural network's decisions when it comes to detecting anomalies. This can be done through using special techniques including attention mechanisms, SHAP or Shapley Additive explanations or XAI which makes the framework more transparent and reliable for decision making purposes. Further, using predictive analytics to detect likelihood of possible future anomalies that may lead to breakdown will greatly enhance preventive strategies of downtime and resource loss. The inclusion of main more complex approaches such as federated learning would enable the framework to handle data from different distributed sources in a secure manner with enhanced privacy thus making the framework suitable for use in industries where data privacy is highly valued. Additionally, it is discovered that making improvements to the requirement of computational resource of the neural network models will bring down the barrier to entry of such framework, specifically for the small organizations. Extending this framework to automatically generate configurations for edge computing setups is also feasible to accommodate the increasing requirement for real-time observability in IoT-based systems. Lastly, developing guidelines and reference data sets to measure the effectiveness of anomaly detection in cloud-native ecosystems will help the field to progress and will promote research that develops sustainable futures for the field.

Compliance with ethical standards

Acknowledge

Acknowledgment The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. The article has no research involving Human Participants and/or Animals. The author has no financial or proprietary interests in any material discussed in this article.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

Ethical Approval Not Applicable

Statement of informed consent

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

[1] Nagios Monitoring System, Comprehensive Monitoring Tool for IT Infrastructure, Nagios Documentation, 2020.

[2] Zabbix Documentation, Advanced Open-Source Monitoring for Distributed Systems, Zabbix Documentation, 2020.

[3] "Kubernetes Monitoring with Prometheus," Kubernetes Documentation, Cloud-Native Computing Foundation, 2021.

[4] Grafana Labs, "Real-Time Visualization and Metrics Analytics for Modern Applications," Grafana Documentation, 2021.

[5] Hyndman, R. J., and Athanasopoulos, G., "Forecasting: Principles and Practice," OTexts, 2018.

[6] Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I., "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," Journal of Official Statistics, vol. 6, no. 1, pp. 3–73, 1990.

[7] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[8] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, 1967.

[9] Hinton, G. E., and Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[10] Kingma, D. P., and Welling, M., "Auto-Encoding Variational Bayes," Proceedings of the International Conference on Learning Representations (ICLR), 2013.

[11] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] Malhotra, P., Vig, L., Shroff, G., and Agarwal, P., "Long Short Term Memory Networks for Anomaly Detection in Time Series," Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2015, pp. 89–94.

[13] Xu, S., Zhao, L., and Zhang, X., "A Hybrid Neural Network Framework for Anomaly Detection in Multivariate Data," Journal of Artificial Intelligence Research, vol. 69, pp. 873–904, 2020.

[14] Yang, F., Liu, H., and Sun, Z., "Online Learning for Real-Time Anomaly Detection," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4455–4466, 2020.

[15] Kumar, A., Singh, R., and Gupta, P., "Graph-Based Neural Networks for Dependency Analysis in Microservices," Journal of Cloud Computing, vol. 10, no. 1, pp. 1–17, 2021.

[16] Zhang, H., Li, L., Wang, X., and Zhou, X. (2021). Adaptive Thresholding for Real-Time Anomaly Detection in Cloud Environments. IEEE Transactions on Cloud Computing, 9(4), 1387–1399. https://doi.org/10.1109/TCC.2020.2984517

[17] Yang, F., Zhang, Y., Zhang, L., and Wang, Q. (2020). Online Learning for Real-Time Anomaly Detection. IEEE Transactions on Neural Networks and Learning Systems, 31(3), 973-983. https://doi.org/10.1109/TNNLS.2019.2924867

[18] Li, T., Wang, H., and Wang, X. (2022). Transfer Learning in Cloud-Native Anomaly Detection. IEEE Access, 10, 10264–10275. https://doi.org/10.1109/ACCESS.2022.3145078

[19] Lee, C., Yoon, J., and Kim, S. (2021). Anomaly Detection in Containerized Environments. Proceedings of the 2021 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 215-221. https://doi.org/10.1109/CloudCom52271.2021.00047

[20] Gao, X., Zhang, X., and Li, F. (2021). Blockchain-Enhanced Anomaly Detection Framework for Multi-Cloud Systems. IEEE Transactions on Industrial Informatics, 17(5), 3206–3215. https://doi.org/10.1109/TII.2020.3005275