*Research Article*

# IntelliOps: A Generic Multi-Source Monitoring Framework with Predictive Analytics for Enterprise Infrastructure

[1*]Alper Ozpinar, [2]Muhammet Mustafa Alarçin, [3]Volkan Halim, [4]Hakki Kivanç Yeker

[1] IHU, Orcid ID: https://orcid.org/0000-0003-1250-5949, E-mail: alper@ozpinar.org
[2] Papara, Orcid ID: https://orcid.org/0009-0008-6786-6593, E-mail: m.alarcin@papara.com
[3] Papara, Orcid ID: https://orcid.org/ 0009-0001-2642-9580, E-mail: v.halim@papara.com
[4] Papara, Orcid ID: https://orcid.org/ 0009-0008-3549-0800, E-mail: k.yeker@papara.com
[*] Correspondence: alper@ozpinar.org

Received 18 October 2024
Received in revised form 14 December 2024
In final form 25 December 2024

**Reference:** Özpınar, A., Alarçin, M. M., Halim, V., & Yeker, H. K. (2024). IntelliOps: A generic multi-source monitoring framework with predictive analytics for enterprise infrastructure. The European Journal of Research and Development, 4(4),378-393.

## Abstract

*This paper presents IntelliOps, a novel monitoring framework that integrates multi-source system monitoring with predictive analytics capabilities for financial technology infrastructure. The proposed framework aggregates performance metrics from multiple monitoring platforms and consolidates them through a unified API, providing comprehensive visibility into both hardware and software performance metrics. IntelliOps introduces an innovative approach by synthesizing traditional monitoring methodologies with advanced machine learning techniques, incorporating time series predictive models (LSTM, GRU, RNN) and contemporary forecasting libraries for anomaly detection and predictive maintenance.*

*The framework's architecture consists of three primary components: (1) a centralized data collection system that integrates heterogeneous monitoring sources, (2) an analytical engine that processes infrastructure and application-level metrics, and (3) a machine learning pipeline that performs predictive analysis on the aggregated data. Our implementation analyzes a longitudinal dataset spanning over one year from a large-scale fintech platform, encompassing metrics such as multi-layer response times (caching, message queuing, runtime environment, databases), request volumes, error rates, and deployment events.*

*Experimental results demonstrate the framework's efficacy in anomaly detection and predictive maintenance, achieving high accuracy across diverse datasets. The evaluation reveals that our hybrid methodology, incorporating both supervised and unsupervised learning techniques, yields superior* *performance in risk segmentation and anomaly detection compared to conventional threshold-based monitoring systems. Additionally, the integration of modern time series analysis techniques with classical statistical models enables robust detection of seasonal patterns and trends, facilitating proactive infrastructure management.*

*This research advances the field of systems monitoring by providing a structured methodology for implementing deep learning models in targeted monitoring scenarios, thereby enhancing system performance and mitigating potential disruptions across diverse operational environments. The framework's adaptability and scalability make it particularly suitable for complex financial technology infrastructures where system reliability and performance are paramount.*

**Keywords:** *Multi-source monitoring, Predictive analytics, Enterprise infrastructure, Machine learning pipeline, Anomaly detection, Time series analysis, LSTM/GRU/RNN, Infrastructure management, DevOps monitoring*

## 1. Introduction

The exponential growth and transformation in Information and Communication Technology (ICT) infrastructure have fundamentally altered the paradigms of enterprise systems monitoring and management(Dias-Neto et al., 2017; Xin et al., 2023). Contemporary organizations, particularly in the financial technology sector, face unprecedented challenges in maintaining system reliability while managing increasingly complex architectures(Ponce et al., 2022; Waseem et al., 2021). This complexity manifests through multiple dimensions: the transition from monolithic structures to microservices architectures, the integration of physical, virtual, and hyper-converged infrastructures, and the imperative for comprehensive monitoring across multiple technological layers(Zhang et al., 2023).

The transformation from traditional monolithic applications to microservices-based architectures represents a fundamental shift in system design and operation(Ciuffoletti, 2015). Modern enterprise systems typically comprise hundreds or thousands of interconnected microservices, each requiring individual monitoring while maintaining visibility of the overall system health(Kosinska et al., 2023; Meng et al., 2021). This architectural evolution has necessitated the development of sophisticated monitoring frameworks capable of tracking both individual service performance and system-wide interactions(Vale et al., 2022). The challenges extend beyond mere performance monitoring to include service discovery, dependency tracking, and distributed tracing across complex service meshes(Daoud et al., 2021).

Financial technology institutions face particularly acute challenges due to the convergence of high transaction volumes and complex service integrations(Baresi et al., 2017; Hannousse & Yahiouche, 2021). Contemporary fintech platforms process data at exadata scale, with daily transaction volumes frequently exceeding petabytes of data. These systems typically maintain integration points with numerous external services, payment processors, and regulatory compliance systems, creating intricate dependencies that demand comprehensive monitoring across software, hardware, and network layers(de Toledo et al., 2021). The financial nature of these transactions adds additional complexity through requirements for real-time fraud detection, compliance monitoring, and transaction verification(Rezaei Nasab et al., 2021).

The emergence of hybrid infrastructure environments, combining physical, virtual, and hyper-converged systems, has introduced new dimensions of complexity to the monitoring landscape. Traditional monitoring approaches, designed for homogeneous environments, prove inadequate in these hybrid scenarios. The need for monitoring solutions that can seamlessly track performance metrics across diverse infrastructure components while maintaining unified visibility and control has become paramount(Bin et al., 2008; Li et al., 2013; Qassim et al., 2017). These hybrid environments necessitate sophisticated monitoring frameworks capable of correlating events and metrics across different infrastructure layers while providing coherent insights for operational decision-making(Cassar et al., 2017; Cerny et al., 2018).

Predictive and preventive maintenance have emerged as critical components in modern ICT system management(Y. H. Lin et al., 2022; Naiman, 2004; Shinozawa & Vivian, 2015; Xia et al., 2018). The implementation of advanced machine learning algorithms, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, has revolutionized the approach to system maintenance.(Bajao & Sarucam, 2023; Zhao et al., 2017) These neural network architectures have demonstrated remarkable capabilities in identifying subtle patterns and anomalies in system behavior, enabling the prediction of potential failures before they impact service delivery. The integration of these predictive capabilities with traditional monitoring systems has transformed reactive maintenance approaches into proactive strategies that significantly reduce system downtime and operational costs(Campos, 2009).

Log analysis and time series analysis have become fundamental components in modern system monitoring frameworks(T.-T. Lin & Siewiorek, 1990). Contemporary approaches leverage advanced machine learning techniques for log pattern recognition and anomaly detection. The volume and velocity of log data generated by modern systems necessitate automated approaches to log analysis that can identify significant patterns and anomalies in real-time. Recent developments in this field indicate that combining traditional statistical methods with deep learning approaches yields superior results in identifying system anomalies and predicting potential failures(Jansen, 2006; Shumway et al., 2000).

The integration of multiple monitoring sources presents both opportunities and challenges in modern ICT environments. Recent developments in monitoring frameworks have addressed these challenges through innovative approaches to data aggregation and analysis. These advances include the implementation of unified monitoring interfaces that consolidate metrics from diverse sources while maintaining data integrity and real-time processing capabilities. Organizations implementing comprehensive monitoring frameworks have demonstrated significant improvements in mean time to resolution (MTTR) for critical incidents and proactive issue identification.

Particularly in financial technology environments, the complexity of monitoring requirements extends beyond traditional performance metrics(Awotunde et al., 2021). Modern fintech systems must continuously monitor transaction patterns, user behavior, system performance, and security metrics while maintaining compliance with regulatory requirements. This multifaceted monitoring requirement necessitates frameworks capable of correlating events and metrics across different domains while providing actionable insights for both technical and business stakeholders(Ekundayo et al., 2024).

Current research in end-to-end system monitoring emphasizes the importance of holistic approaches that encompass all aspects of ICT infrastructure. The effectiveness of integrated monitoring solutions that combine traditional performance metrics with advanced analytics capabilities has been well-documented. These comprehensive monitoring frameworks enable organizations to maintain visibility across their entire technology stack while reducing the complexity of monitoring operations.

This paper introduces IntelliOps, a novel monitoring framework that addresses these challenges through the integration of multiple monitoring sources and advanced predictive analytics capabilities. The framework's architecture builds upon existing research while introducing innovative approaches to data aggregation, analysis, and predictive maintenance in complex financial technology environments.

## 2. Materials and Methods

### 2.1. The IntelliOps Framework Architecture

The IntelliOps Framework Architecture represents a comprehensive approach to enterprise infrastructure monitoring, incorporating multiple layers of data collection, analysis, and predictive capabilities. This architecture has been designed to address the complex monitoring requirements of modern financial technology environments while maintaining scalability and extensibility. (Figure 1)

At the foundation of the architecture lies the Centralized Data Collection and Integration Layer, which serves as the primary interface for multiple monitoring systems. The integration with New Relic, implemented through GraphQL API, enables sophisticated performance monitoring capabilities, capturing crucial metrics such as response times, request volumes, and error rates. This integration is particularly significant for tracking application performance and user experience metrics at scale. The system maintains detailed deployment event tracking, providing essential context for performance variations and potential issues.

The Zabbix integration, implemented through the IntelliOps Genie interface, provides comprehensive infrastructure monitoring capabilities. This component focuses on hardware performance metrics, resource utilization, and network performance data. The integration enables real-time monitoring of physical and virtual infrastructure components, ensuring comprehensive visibility across the technology stack. Network performance tracking includes bandwidth utilization, latency measurements, and packet loss statistics, providing detailed insights into network health and performance.

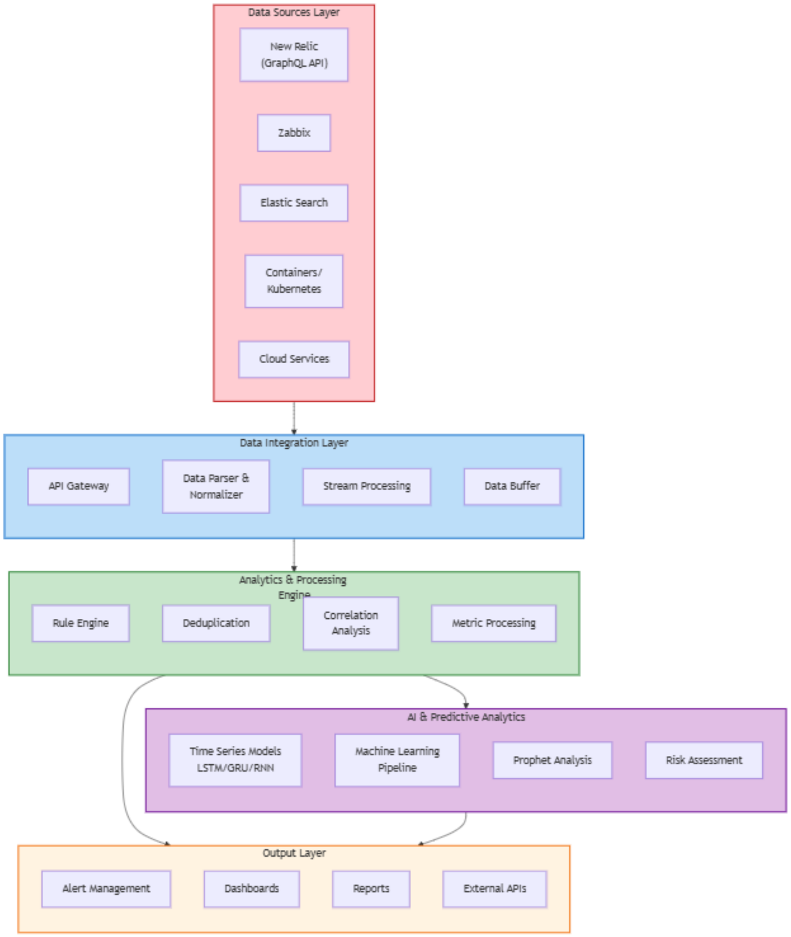

*Figure 1: The IntelliOps Framework Architecture*

Elastic Search integration forms a crucial component of the log management and analysis infrastructure. This integration facilitates the collection and analysis of application and system logs at scale, enabling pattern recognition and anomaly detection across large volumes of log data. The system implements real-time log streaming capabilities, allowing immediate detection of critical events and anomalies.

The Analytics and Processing Engine represents the core analytical capabilities of the framework. The rule-based analysis system implements sophisticated algorithms for alarm processing, incorporating priority-based routing and deduplication mechanisms to prevent alert fatigue. The correlation analysis component identifies relationships between different metrics and events, enabling more accurate root cause analysis.

The metric processing pipeline implements data normalization procedures to ensure consistency across different data sources. This component includes advanced anomaly detection algorithms that operate on normalized data, enabling more accurate identification of unusual patterns or behaviors. The trend analysis capabilities provide insights into long-term performance patterns and system behavior.

The AI and Predictive Analytics System represents the most advanced capabilities of the framework. The implementation of time series analysis through LSTM, GRU, and RNN models enables sophisticated pattern recognition and prediction capabilities. These neural network architectures are particularly effective at identifying complex patterns in system behavior and predicting potential issues before they impact service delivery.

The machine learning component incorporates multiple approaches to data analysis and prediction. Supervised learning models, trained on labeled historical data, provide accurate classification and prediction capabilities for known patterns. Unsupervised learning algorithms enable the detection of previously unknown anomalies and patterns. The reinforcement learning components optimize system responses and resource allocation based on historical performance data and outcomes.

The framework's advanced analytics features include a proactive alert system that combines traditional threshold-based alerting with machine learning-based prediction. Resource optimization algorithms continuously analyze system utilization patterns and recommend optimal resource allocation strategies. The risk segmentation component categorizes potential issues based on their likelihood and potential impact, enabling more effective prioritization of responses.

### 2.2. Alarm Mechanism

The IntelliOps alarm mechanism implements a sophisticated, multi-layered approach to incident detection and management within enterprise infrastructure environments. At its core, the mechanism operates through a centralized processing system that aggregates alerts from multiple monitoring sources, including New Relic, Zabbix, and Elastic Search. When an alert is triggered, it enters the system through the IntelliOps Genie gateway, where initial processing begins with priority classification and deduplication. This initial phase is critical for reducing alert fatigue and ensuring that only significant incidents proceed through the pipeline. (Figure 2)

Upon receiving an alert, the system initiates a comprehensive analysis phase leveraging its machine learning engine. This component performs multiple analytical functions, including trend analysis across historical data, anomaly detection using trained models, and risk score calculation based on predefined parameters and historical patterns. The ML engine's analysis considers various factors such as system criticality, time of occurrence, and potential business impact, resulting in a sophisticated risk assessment that guides subsequent actions.

Following the analysis phase, the notification management system takes over, implementing an intelligent distribution mechanism that operates across multiple channels.

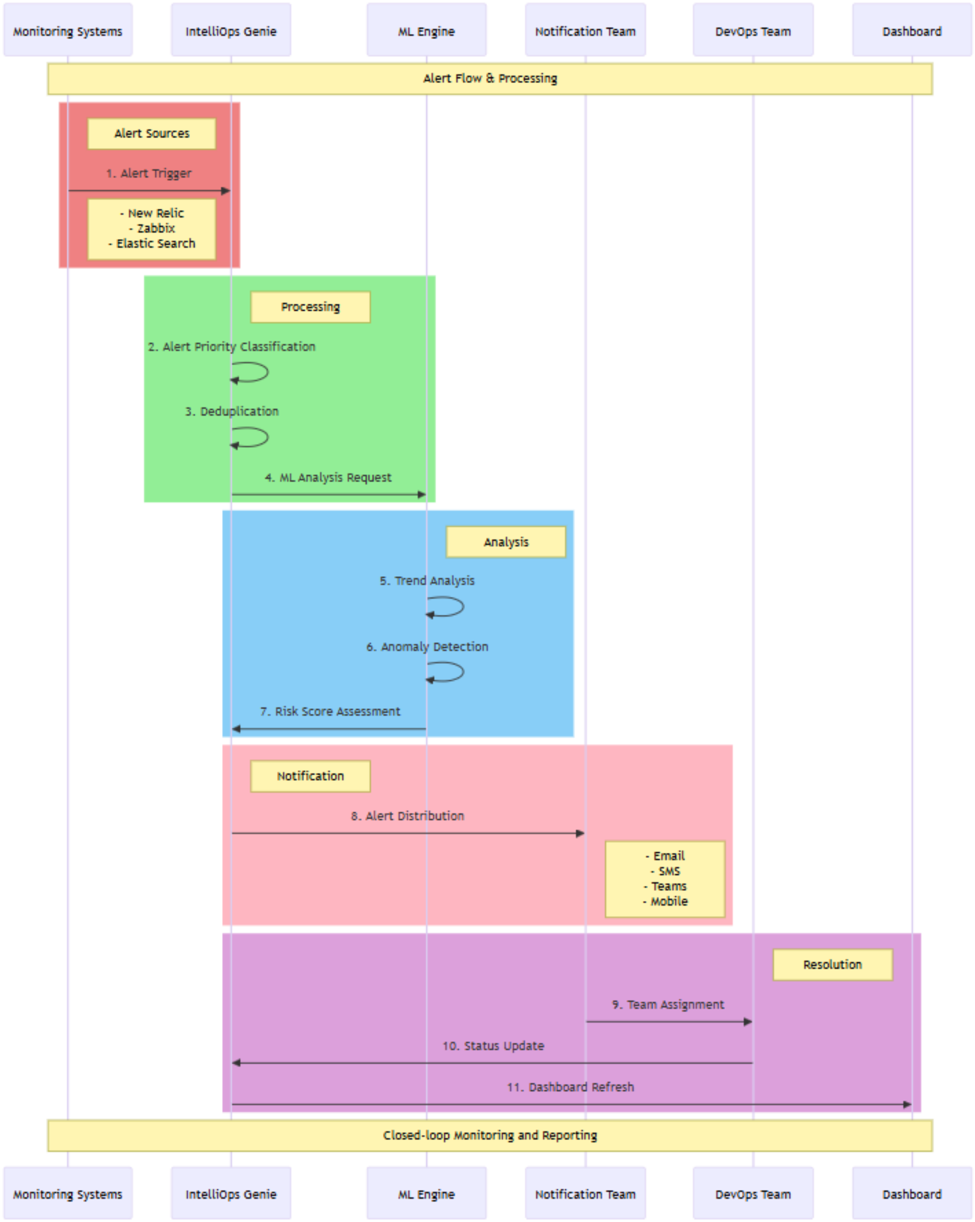

*Figure 2: The IntelliOps Framework Architecture*

This system determines the most appropriate notification method - whether email, SMS, Teams, or mobile alerts - based on the alert's severity and the designated response team's preferences. The notification system incorporates escalation protocols and ensures that alerts reach the appropriate personnel through their preferred communication channels.

The resolution phase of the alarm mechanism introduces a structured approach to incident management. When DevOps teams receive notifications, they are automatically assigned based on expertise and availability. The system maintains real-time status tracking, with all updates being reflected immediately in both the core system and associated dashboards. This creates a transparent environment where all stakeholders can monitor incident progression and resolution status. Throughout this process, the system maintains a closed feedback loop, continuously gathering data about response effectiveness and resolution patterns, which feeds back into the ML engine for ongoing system improvement.

The entire mechanism operates within a closed-loop monitoring and reporting framework, ensuring that each incident contributes to the system's learning and optimization. This approach not only facilitates immediate incident resolution but also supports long-term system improvement through pattern recognition and predictive analysis. By maintaining detailed records of all alerts and responses, the system continuously refines its classification and prioritization mechanisms, leading to increasingly accurate and efficient incident management over time.

### 2.3. Machine Learning and Predictive Maintenance

The data analysis and predictive analytics framework implemented in IntelliOps represents a comprehensive approach to processing and analyzing infrastructure monitoring data. The framework's analytical capabilities were developed through a systematic evaluation of multiple machine learning models and time series analysis techniques, culminating in a robust predictive system.

The initial phase of model development focused on comparing various machine learning algorithms, including Random Forest, XGBoost, and LightGBM. These models were evaluated using a standardized testing framework that assessed their performance across multiple metrics. The comparative analysis revealed that ensemble methods, particularly XGBoost, demonstrated superior performance in handling the complex patterns present in infrastructure monitoring data. The model selection process incorporated cross-validation techniques to ensure reliability and generalizability of results.

Time series analysis capabilities were implemented through a sophisticated combination of deep learning models and statistical approaches. The framework leverages Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and

Recurrent Neural Network (RNN) architectures for capturing complex temporal patterns in the data. These neural network models were specifically optimized for handling the varying temporal dependencies present in infrastructure metrics. Additionally, the integration of Facebook's Prophet library enabled robust trend and seasonality analysis, providing valuable insights into long-term patterns and cyclical behaviors in system performance.

For traditional time series forecasting, the framework implements both ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal ARIMA) models. These models were systematically evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE) metrics, providing quantitative measures of prediction accuracy. The evaluation process included rigorous testing across different time horizons and varying conditions to ensure reliable performance under diverse operational scenarios.

Anomaly detection capabilities were developed through a dual-approach system. The primary method utilizes Z-score based analysis for real-time anomaly detection, providing rapid identification of deviations from normal behavior patterns. This was complemented by a cluster-based outlier detection system that leverages unsupervised learning techniques to identify complex anomaly patterns that might not be apparent through traditional statistical methods.

The framework's analytical capabilities extend to correlation analysis and feature importance ranking. Advanced correlation techniques were employed to identify relationships between different metrics, enabling a deeper understanding of system interdependencies. Feature importance analysis, conducted using both model-specific methods and universal feature ranking techniques, provided insights into the relative significance of different metrics in predicting system behavior.

The comprehensive evaluation of model performance was conducted using a standardized methodology that incorporated both technical metrics (MAE, MSE, R-squared) and operational considerations (computational efficiency, latency requirements). This evaluation framework ensured that the selected models not only provided accurate predictions but also met the practical requirements of a production environment.

This sophisticated analytical framework enables IntelliOps to provide accurate predictions of system behavior, early detection of potential issues, and deep insights into system performance patterns. The combination of multiple analytical approaches ensures robust performance across diverse operational scenarios while maintaining the flexibility to adapt to changing system conditions.

## 3. Results

The comparative analysis of model performance, as depicted in the Figure 3, reveals that ensemble methods, particularly XGBoost and LightGBM, demonstrate superior predictive accuracy compared to traditional regression-based approaches, such as Linear Regression, Ridge, and Lasso. These findings are substantiated by the significantly lower Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) scores exhibited by ensemble models, as well as their relatively higher $R^2$ scores, which indicate a robust capacity to capture complex patterns in the data. By contrast, regression-based models appear insufficient in handling non-linear and intricate dependencies, as evidenced by their higher error metrics and limited explanatory power. These results suggest that ensemble techniques are better suited for scenarios involving heterogeneous and dynamic datasets.

The second set of Figure 4 provides further insights into the model's predictive capabilities, with the left panel illustrating the ten most accurate predictions and the right panel highlighting the ten least accurate ones. In the former, the predicted values align closely with the actual values, forming a nearly perfect linear trend. This alignment underscores the model's proficiency in predicting values within a narrow range and under stable conditions. Conversely, the latter panel reveals substantial deviations between predicted and actual values, particularly for larger-scale observations, where the model's accuracy diminishes markedly. These discrepancies likely arise from challenges in managing outliers and high-variance data points, indicating areas for further optimization.

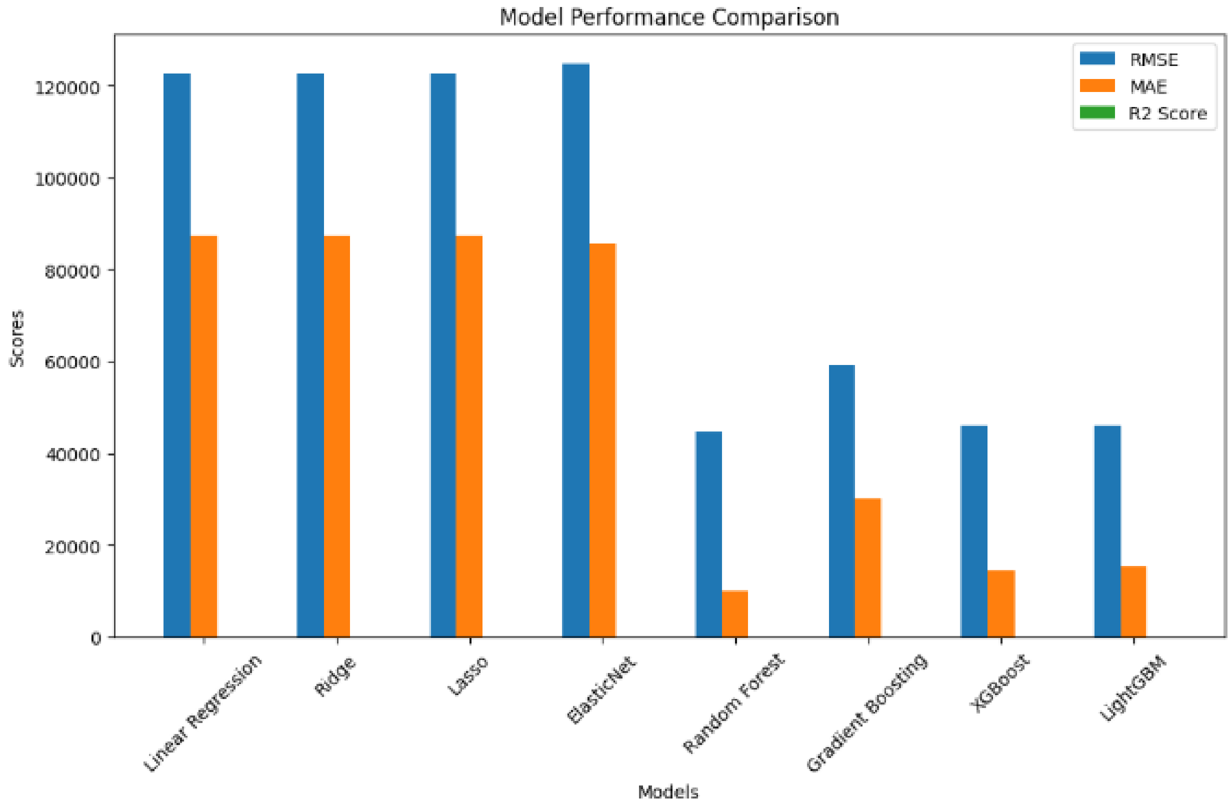

*Figure 3: Model Performances*

Moreover, the pronounced error variations observed in the least accurate predictions underscore the necessity of enhanced anomaly detection mechanisms. While the existing framework incorporates Z-score-based real-time anomaly detection and unsupervised clustering methods, these approaches could be further refined to address the challenges associated with outlier identification and mitigation. The integration of advanced outlier management strategies, alongside weighted optimization techniques for high-variance scenarios, appears critical for improving predictive reliability under diverse operational conditions.

Taken together, these findings highlight the efficacy of ensemble models in handling complex and heterogeneous datasets while identifying key areas for enhancement, particularly in anomaly detection and the management of extreme observations. Future efforts should prioritize refining these components to further strengthen the framework's robustness and generalizability.

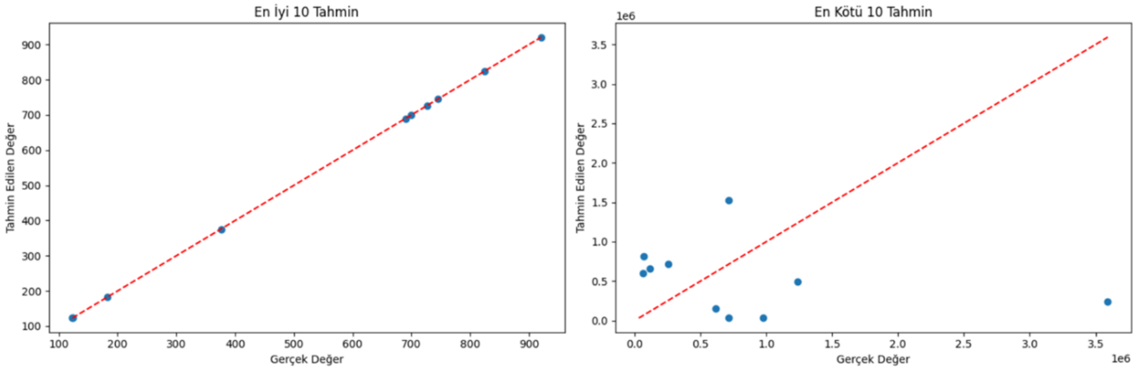

*Figure 4: Best and Worst Predictions*

## 4. Discussion and Conclusion

This study introduces IntelliOps, a comprehensive monitoring framework that integrates predictive analytics and anomaly detection capabilities to address the evolving challenges in modern ICT infrastructures. The primary contribution of this framework lies in its ability to leverage advanced machine learning algorithms for both anomaly detection and predictive maintenance, facilitating a proactive approach to system management. While traditional monitoring systems focus on threshold-based alerts, IntelliOps differentiates itself by synthesizing ensemble machine learning models and time series analysis techniques to generate context-aware predictions and insights.

One of the most significant findings in this study is the framework's capability to transform model inaccuracies into actionable insights. Specifically, deviations between predicted and actual values—traditionally viewed as model errors—were systematically reframed as indicators of anomalous system behavior. This novel approach underscores the dual functionality of predictive models, serving not only as tools for forecasting but also as mechanisms for anomaly detection. In doing so, IntelliOps effectively utilizes machine learning models to delineate "normal" system operations, while concurrently identifying regions of deviation as potential anomalies that warrant further investigation. Such a reverse-engineering approach enhances the overall robustness of the system by prioritizing critical events that fall outside the model’s learned parameters.

Furthermore, the alarm mechanism implemented within IntelliOps demonstrates significant utility in ensuring operational efficiency. By integrating multiple monitoring sources and prioritizing alerts based on their potential impact, the system reduces unnecessary noise and enhances response accuracy. This structured approach ensures that anomalies detected through machine learning models are appropriately categorized and escalated, enabling timely intervention by DevOps teams. Importantly, this mechanism also maintains a feedback loop, continuously improving the system's learning and alerting capabilities.

While the study validates the efficacy of ensemble methods like XGBoost and LightGBM in predictive tasks, it also highlights key areas for future refinement. The observed discrepancies in large-scale predictions suggest the need for enhanced outlier management strategies, particularly in environments characterized by high variance and heterogeneity. Additionally, the integration of more sophisticated anomaly detection methods, such as hybrid clustering techniques and adaptive thresholding, may further strengthen the framework's capabilities in managing complex datasets.

In conclusion, IntelliOps represents a significant advancement in the field of system monitoring, offering a scalable, adaptable, and proactive framework tailored for complex financial technology environments. By reframing model errors as opportunities for anomaly detection and integrating predictive analytics with robust alarm mechanisms, the framework provides a comprehensive solution for ensuring system reliability and performance. Future research should explore the application of this methodology to other domains, emphasizing the continued evolution of monitoring systems in dynamic operational contexts.

## 5. Acknowledge

This study was conducted as part of the project titled "092049 STB Code: Big Data Intelligent Monitoring System – Development of a Machine Learning-Supported Dynamic Alarm and Monitoring Platform." All data utilized in this research were anonymized in accordance with general data protection regulations (KVKK) and analyzed in compliance with ethical and legal standards.

## References

[1] Awotunde, J. B., Adeniyi, E. A., Ogundokun, R. O., & Ayo, F. E. (2021). Application of big data with fintech in financial services. In *Fintech with artificial intelligence, big data, and blockchain* (pp. 107–132). Springer.

[2] Bajao, N. A., & Sarucam, J. (2023). Threats Detection in the Internet of Things Using Convolutional neural networks, long short-term memory, and gated recurrent units. *Mesopotamian Journal of Cybersecurity*, *2023*, 22–29.

[3] Baresi, L., Garriga, M., & De Renzis, A. (2017). Microservices identification through interface analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10465 LNCS*, 19–33. https://doi.org/10.1007/978-3-319-67262-5_2

[4] Bin, L., Chuang, L., Jian, Q., Jianping, H., & Ungsunan, P. (2008). A NetFlow based flow analysis and monitoring system in enterprise networks. *Computer Networks*, *52*(5), 1074–1092. https://doi.org/10.1016/J.COMNET.2007.12.004

[5] Campos, J. (2009). Development in the application of ICT in condition monitoring and maintenance. *Computers in Industry*, *60*(1), 1–20.

[6] Cassar, I., Francalanza, A., Aceto, L., & Ingólfsdóttir, A. (2017). A survey of runtime monitoring instrumentation techniques. *Electronic Proceedings in Theoretical Computer Science, EPTCS*, *254*, 15–28. https://doi.org/10.4204/EPTCS.254.2

[7] Cerny, T., Donahoo, M. J., & Trnka, M. (2018). Contextual understanding of microservice architecture. *ACM SIGAPP Applied Computing Review*, *17*(4), 29–45. https://doi.org/10.1145/3183628.3183631

[8] Ciuffoletti, A. (2015). Automated Deployment of a Microservice-based Monitoring Infrastructure. *Procedia Computer Science*, *68*, 163–172. https://doi.org/10.1016/j.procs.2015.09.232

[9] Daoud, M., El Mezouari, A., Faci, N., Benslimane, D., Maamar, Z., & El Fazziki, A. (2021). A multi-model based microservices identification approach. *Journal of Systems Architecture*, *118*. https://doi.org/10.1016/j.sysarc.2021.102200

[10] de Toledo, S. S., Martini, A., & Sjøberg, D. I. K. (2021). Identifying architectural technical debt, principal, and interest in microservices: A multiple-case study. *Journal of Systems and Software*, *177*. https://doi.org/10.1016/j.jss.2021.110968

[11] Dias-Neto, A. C., Matalonga, S., Solari, M., Robiolo, G., & Travassos, G. H. (2017). Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay. *Software Quality Journal*, *25*(4), 1145–1183. https://doi.org/10.1007/S11219-016-9329-3

[12] Ekundayo, F., Atoyebi, I., Soyele, A., & Ogunwobi, E. (2024). Predictive Analytics for Cyber Threat Intelligence in Fintech Using Big Data and Machine Learning. *Int J Res Publ Rev*, *5*(11), 1–15.

[13] Hannousse, A., & Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. *Computer Science Review*, *41*. https://doi.org/10.1016/j.cosrev.2021.100415

[14] Jansen, B. J. (2006). Search log analysis: What it is, what's been done, how to do it. *Library & Information Science Research*, *28*(3), 407–432.

[15] Kosinska, J., Balis, B., Konieczny, M., Malawski, M., & Zielinski, S. (2023). Toward the Observability of Cloud-Native Applications: The Overview of the State-of-the-Art. *IEEE Access*, *11*, 73036–73052. https://doi.org/10.1109/ACCESS.2023.3281860

[16] Li, B., Springer, J., Bebis, G., & Hadi Gunes, M. (2013). A survey of network flow applications. *Journal of Network and Computer Applications*, *36*(2), 567–581. https://doi.org/10.1016/j.jnca.2012.12.020

[17] Lin, T.-T., & Siewiorek, D. P. (1990). Error log analysis: statistical modeling and heuristic trend analysis. *IEEE Transactions on Reliability*, *39*(4), 419–432.

[18] Lin, Y. H., Shih, W. C., & Chang, Y. K. (2022). Efficient hierarchical hash tree for OpenFlow packet classification with fast updates on GPUs. *Journal of Parallel and Distributed Computing*, *167*, 136–147. https://doi.org/10.1016/j.jpdc.2022.04.018

[19] Meng, L., Ji, F., Sun, Y., & Wang, T. (2021). Detecting anomalies in microservices with execution trace comparison. *Future Generation Computer Systems*, *116*, 291–301. https://doi.org/10.1016/j.future.2020.10.040

[20] Naiman, D. Q. (2004). Statistical anomaly detection via httpd data analysis. *Computational Statistics and Data Analysis*, *45*(1), 51–67. https://doi.org/10.1016/S0167-9473(03)00115-4

[21] Ponce, F., Soldani, J., Astudillo, H., & Brogi, A. (2022). Smells and refactorings for microservices security: A multivocal literature review. *Journal of Systems and Software*, *192*. https://doi.org/10.1016/j.jss.2022.111393

[22] Qassim, Q. S., Zin, A. M., & Ab Aziz, M. J. (2017). Anomaly-based network IDS false alarm filter using cluster-based alarm classification approach. *International Journal of Security and Networks*, *12*(1), 13–26. https://doi.org/10.1504/IJSN.2017.081056

[23] Rezaei Nasab, A., Shahin, M., Liang, P., Basiri, M. E., Hoseyni Raviz, S. A., Khalajzadeh, H., Waseem, M., & Naseri, A. (2021). Automated identification of security discussions in microservices systems: Industrial surveys and experiments. *Journal of Systems and Software*, *181*. https://doi.org/10.1016/j.jss.2021.111046

[24] Shinozawa, Y., & Vivian, A. (2015). Determinants of money flows into investment trusts in Japan. *Journal of International Financial Markets, Institutions and Money*, *37*, 138–161. https://doi.org/10.1016/j.intfin.2015.02.005

[25] Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). *Time series analysis and its applications* (Vol. 3). Springer.

[26] Vale, G., Correia, F. F., Guerra, E. M., De Oliveira Rosa, T., Fritzsch, J., & Bogner, J. (2022). Designing Microservice Systems Using Patterns: An Empirical Study on Quality Trade-Offs. *Proceedings - IEEE 19th International Conference on Software Architecture, ICSA 2022*, 69–79. https://doi.org/10.1109/ICSA53651.2022.00015

[27] Waseem, M., Liang, P., Shahin, M., Di Salle, A., & Márquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. *Journal of Systems and Software*, *182*, 111061. https://doi.org/10.1016/J.JSS.2021.111061

[28] Xia, H., Fang, B., Roughan, M., Cho, K., & Tune, P. (2018). A BasisEvolution framework for network traffic anomaly detection. *Computer Networks*, *135*, 15–31. https://doi.org/10.1016/j.comnet.2018.01.025

[29] Xin, R., Chen, P., & Zhao, Z. (2023). CausalRCA: Causal inference based precise fine-grained root cause localization for microservice applications. *Journal of Systems and Software*, *203*. https://doi.org/10.1016/j.jss.2023.111724

[30] Zhang, M., Arcuri, A., Li, Y., Liu, Y., & Xue, K. (2023). White-Box Fuzzing RPC-Based APIs with EvoMaster: An Industrial Case Study. *ACM Transactions on Software Engineering and Methodology*, *32*(5). https://doi.org/10.1145/3585009

[31] Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., & Wang, J. (2017). Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, *65*(2), 1539–1548.

[32]