WCRG Global Research
WORLD COMET RESEARCH GROUP

International Journal of Artificial Intelligence & Digital Transformation
Vol. 1, No. 1, 2018
Doi: 10.67228/30713315/IJAIDT-2018PI9V6T
PP. 01-11

Original Article

AI/ML Techniques for Predictive Maintenance of Cloud Infrastructure

Dr. T. Rakesh

Assistant Professor, Department of Economics, University of Delhi, New Delhi, India.

Received: 03-04-2018 Revised: 03-24-2018 Accepted: 04-03-2018 Published: 04-05-2018

ABSTRACT

Cloud infrastructure plays a pivotal role in delivering uninterrupted computing services across the globe. However, unexpected failures in critical components such as cooling systems, networking hardware, and power supply units can result in significant downtime and economic loss. This paper investigates the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques for predictive maintenance in cloud data centers. By analyzing historical sensor data, system logs, and performance metrics, various ML models—ranging from classical approaches like Random Forests to deep learning models such as LSTM and Autoencoders—are deployed to predict potential failures and trigger maintenance alerts. The paper also explores challenges like data imbalance, real-time inference, and deployment at scale in cloud environments. Experimental evaluations demonstrate the efficacy of ML models in identifying fault patterns, thereby improving operational reliability, reducing maintenance costs, and minimizing service disruptions.

KEYWORDS

Predictive Maintenance, Cloud Infrastructure, Machine Learning, Anomaly Detection, Data Center Reliability, Time Series Forecasting, Fault Detection, Lstm, Edge Monitoring, Operational AI.

ISSN: 3071-3315
PP: 01-11.

1. INTRODUCTION

1.1. Motivation for Predictive Maintenance in Cloud Environments

Cloud data centers form the critical backbone of modern digital services, supporting everything from streaming platforms and financial transactions to AI-powered applications. As global reliance on cloud services intensifies, ensuring the reliability and efficiency of the underlying infrastructure becomes paramount. Traditional reactive or time-based maintenance approaches often lead to unplanned downtime, performance degradation, and increased operational costs. Predictive maintenance, driven by AI/ML, offers a proactive solution by forecasting potential failures before they occur. This capability is essential for avoiding service disruptions, optimizing asset lifecycles, and enhancing the operational health of cloud facilities.

1.2. Economic and Service-Level Impact of Infrastructure Failures

Unscheduled failures in cloud infrastructure components such as cooling systems, networking hardware, or power supplies can have severe financial and reputational consequences. Even a few minutes of data center downtime can translate to millions of dollars in lost revenue, especially for service providers operating at hyperscale. Beyond the economic loss, failures compromise service-level agreements (SLAs), reduce customer satisfaction, and may even lead to contract penalties or legal issues. The cascading effect of hardware faults on hosted virtual machines and dependent services makes failure prevention not just a technical necessity but also a business imperative.

1.3. Role of AI/ML in Addressing the Issue

Artificial Intelligence (AI) and Machine Learning (ML) present a transformative approach to infrastructure maintenance by enabling intelligent monitoring, pattern recognition, and anomaly detection. Unlike static rule-based systems, ML models can adapt to the dynamic behavior of hardware systems, learn from historical and real-time data, and predict potential breakdowns with greater accuracy. These technologies can be deployed to analyze sensor data from temperature probes, vibration monitors, network packet traces, and log files to forecast failure events before they happen. By automating fault detection and optimizing maintenance schedules, AI/ML dramatically improves operational reliability and cost efficiency.

1.4. Contributions of this Paper

This paper presents a comprehensive study of AI/ML techniques applied to predictive maintenance within cloud infrastructure, focusing specifically on components such as cooling systems and networking equipment. We propose an end-to-end pipeline involving data collection, preprocessing, model training, and real-time inference for failure prediction. Our contribution includes the evaluation of multiple machine learning models, including supervised, unsupervised, and deep learning approaches, with a comparative analysis of their effectiveness in different maintenance scenarios. Additionally, we discuss challenges such as data quality, model deployment in cloud-native environments, and scalability. The findings aim to serve as a blueprint for integrating intelligent maintenance systems into modern cloud operations.

ISSN: 3071-3315
PP: 01-11.

2. BACKGROUND AND RELATED WORK

2.1. Overview of Cloud Infrastructure Components (Cooling, Networking, Power)

Cloud data centers comprise a vast array of interconnected components that ensure the seamless delivery of computational resources. These include not only computing hardware (servers, storage devices) but also essential infrastructure such as power distribution units (PDUs), network switches and routers, and thermal management systems (cooling towers, CRAC units, airflow systems). Cooling systems maintain temperature thresholds to avoid thermal damage to hardware, while networking hardware manages high-throughput, low-latency data exchange between distributed systems. The power infrastructure ensures continuous uptime with redundancy and backup mechanisms. These components work together in a highly orchestrated manner, and failure in any one of them can lead to service disruption.

2.2. Traditional Maintenance vs Predictive Maintenance

Traditionally, cloud infrastructure relies on either reactive or preventive maintenance strategies. Reactive maintenance involves addressing issues only after failure occurs, which often results in costly downtimes. Preventive maintenance follows a scheduled approach, where components are serviced or replaced at regular intervals regardless of their condition. However, this can be inefficient and lead to premature equipment replacement. Predictive maintenance, in contrast, leverages data-driven insights to assess the health of equipment and predict failures before they occur. This approach allows operators to perform maintenance only when necessary, thereby optimizing costs, extending equipment life, and avoiding unplanned outages.

2.3. Prior Research on AI/ML for Equipment Monitoring

Previous studies have demonstrated the feasibility and benefits of applying AI/ML to predictive maintenance across various industries, including manufacturing, transportation, and energy. In the cloud context, some research has applied anomaly detection techniques to temperature and power data, while others have explored supervised learning to detect early signs of component degradation. Time-series forecasting using models like LSTM has also gained traction for monitoring temperature, fan speed, and network latency. However, much of the existing work is domain-specific and lacks generalization to the complex, multi-layered environment of cloud infrastructure.

2.4. Gaps in Existing Literature

Despite growing interest, there are several gaps in current research. Firstly, most studies lack a comprehensive integration of predictive maintenance systems across multiple infrastructure domains (cooling, networking, power) in a unified model. Secondly, few papers consider real-time deployment challenges in dynamic, multi-tenant cloud environments. Thirdly, while ML models are often proposed, discussions on handling data imbalance, concept drift, and privacy concerns are limited. Lastly, there's a lack of standardized benchmarks and datasets for predictive maintenance in cloud settings, making comparative evaluation and replication difficult. This paper seeks to address these gaps by providing a holistic, implementable framework with detailed experimentation.

ISSN: 3071-3315
PP: 01-11.

3. PROBLEM STATEMENT AND OBJECTIVES

3.1. Definition of Predictive Maintenance in Cloud Context

In the context of cloud infrastructure, predictive maintenance refers to the intelligent anticipation of component failures such as network switches overheating, fans slowing down, or cooling units malfunctioning before they cause significant downtime. This is achieved by continuously collecting and analyzing telemetry data (e.g., temperature, vibration, voltage, latency) from various infrastructure components and using AI/ML models to forecast abnormal patterns. The aim is to perform just-in-time maintenance actions that are cost-effective and minimize disruption to services.

3.2. Key Research Objectives and Hypotheses

The primary objective of this research is to develop and evaluate AI/ML-based models capable of predicting failures in cloud infrastructure with high accuracy and low latency. Key questions include: Can historical telemetry data be used to forecast infrastructure faults? Which models (supervised, unsupervised, or time-series) are most effective for different component types? What are the trade-offs between prediction accuracy, computational overhead, and real-time deployability? We hypothesize that deep learning models like LSTM will outperform traditional models in time-dependent failure prediction, especially when trained on large datasets, but may be more complex to deploy at scale.

3.3. System Constraints and Assumptions

This study assumes access to sufficient historical data for training models, including sensor readings, performance logs, and maintenance records. It also assumes that cloud infrastructure provides the necessary APIs or monitoring tools (e.g., Prometheus, AWS CloudWatch) for real-time data collection. Constraints include limited labeled failure data (due to rarity of events), class imbalance, and varying sensor quality across data centers. Additionally, the proposed system is designed under the assumption that it will be deployed in a modular, cloud-native environment that supports containerization and microservices architecture.

4. DATA COLLECTION AND PREPROCESSING

4.1. Data Sources: Sensors, Logs, Network Traffic, Environmental Data

The foundation of any predictive maintenance model lies in the quality and diversity of its input data. In cloud environments, data is sourced from multiple sensors and monitoring tools embedded across infrastructure components. These include temperature sensors on CPUs and cooling units, voltage and power sensors on PDUs, network throughput and latency logs from switches and routers, as well as environmental data such as humidity and airflow. Logs generated by system monitoring tools and event managers provide additional context, such as prior maintenance actions or fault reports. Collecting data from these heterogeneous sources allows the model to build a more holistic understanding of system health and behavior.

4.2. Data Cleaning, Feature Extraction, and Normalization

Raw infrastructure data often contains noise, missing values, or outliers due to sensor errors or data transmission faults. Therefore, data cleaning is a critical step that involves removing corrupted

ISSN: 3071-3315
PP: 01-11.

entries, imputing missing values (e.g., using forward filling or statistical methods), and filtering out anomalous spikes that are unrelated to faults. Feature extraction involves identifying and engineering meaningful variables from the raw data for example, calculating temperature gradients, time since last maintenance, rolling averages of network delay, or rate of change in fan RPM. These features are then normalized (e.g., using Min-Max scaling or Z-score normalization) to ensure consistent input for ML models and to improve convergence during training.

4.3. Handling Missing Data and Class Imbalance

One of the biggest challenges in building predictive maintenance models is the presence of class imbalance failure events are rare compared to normal operation. This can lead to biased models that are good at predicting "normal" but fail to detect "abnormal" events. Techniques such as SMOTE (Synthetic Minority Over-sampling Technique), anomaly detection models, or cost-sensitive learning can be used to mitigate this issue. Additionally, missing data is often inevitable due to sensor downtimes or logging errors. Strategies like interpolation, KNN imputation, or model-based imputation (e.g., using autoencoders) are employed to fill gaps without introducing bias. Careful attention to these preprocessing steps ensures that the downstream ML models are both robust and accurate.

Table 1: Overview of Data Collected from Cloud Infrastructure Components

Data Source	Type of Data	Frequency	Purpose
Temperature Sensors	CPU temp, Ambient temp	Every 1 min	Detect overheating/cooling issues
Network Logs	Packet loss, Latency, Throughput	Every 5 sec	Detect network degradation
Power Distribution Units (PDUs)	Voltage, Current, Power usage	Every 10 sec	Identify power fluctuations
Maintenance Logs	Repair history, Failure labels	On event	Model supervision, ground truth
Environmental Sensors	Humidity, Airflow	Every 5 min	Contextual features for model

5. ML MODELS FOR PREDICTIVE MAINTENANCE

5.1. Model Selection Rationale

Choosing the appropriate machine learning model is critical in predictive maintenance, as different models have unique strengths depending on data characteristics and the problem context. In this work, a combination of supervised, unsupervised, and time-series deep learning models is explored to address the diverse nature of cloud infrastructure data. The rationale behind this selection is to ensure broad coverage of both labeled and unlabeled scenarios, account for temporal dependencies, and effectively detect subtle degradation patterns in cloud components before they lead to failures.

5.2. Supervised Models: Random Forest, XGBoost

Supervised learning models are particularly effective when labeled historical failure data is available. Random Forest is chosen for its robustness to overfitting, ability to handle high-dimensional data, and interpretability. It works well on tabular sensor data and can provide insights into feature

ISSN: 3071-3315
PP: 01-11.

importance, aiding explainability. XGBoost, on the other hand, is a gradient-boosting algorithm known for its high accuracy and efficiency in handling large-scale structured data. It often outperforms other tree-based models in predictive tasks and is well-suited for handling missing values and imbalanced datasets a common scenario in predictive maintenance where failures are rare events.

5.3. Unsupervised Models: Autoencoders, Isolation Forest

In many real-world settings, labeled failure data is scarce or incomplete. Hence, unsupervised learning models are utilized to detect anomalies without requiring labeled examples. Autoencoders, a type of neural network, are trained to reconstruct normal behavior patterns from the input data. When the system behavior deviates significantly from the learned pattern, the reconstruction error increases, signaling a potential anomaly. Isolation Forest is another anomaly detection algorithm that isolates observations by randomly selecting features and splitting values. It is particularly effective for identifying outliers in multi-dimensional sensor data and operates efficiently even with high-dimensional inputs.

5.4. Time-Series Models: LSTM, GRU

Many infrastructure metrics such as temperature, fan speed, and network latency are inherently temporal. To capture long-term dependencies and trends in such sequences, Recurrent Neural Networks (RNNs) are used. Specifically, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures are implemented. These models are designed to retain historical context over extended time windows and can learn complex temporal patterns leading up to failures. LSTM and GRU are especially valuable for forecasting maintenance needs based on evolving sensor data, and for generating alerts before thresholds are crossed.

5.5. Model Architecture and Training Approach

Each model is tailored to fit the specific characteristics of the dataset. For Random Forest and XGBoost, a grid search is performed to optimize hyperparameters such as tree depth, learning rate, and number of estimators. For Autoencoders, a symmetric neural network with bottleneck architecture is employed, where the reconstruction error serves as the anomaly score. LSTM/GRU networks are structured with multiple hidden layers and trained on sequences of historical sensor readings, using techniques such as sliding windows to prepare the data. The training process involves dividing the dataset into training, validation, and test sets, using early stopping to prevent overfitting, and employing regularization techniques like dropout and batch normalization.

5.6. Evaluation Metrics (Precision, Recall, F1-score, AUC, etc.)

Given the class imbalance and critical nature of fault prediction, multiple evaluation metrics are used to assess model performance. Precision measures the proportion of correctly predicted failures out of all predicted failures, while recall (sensitivity) indicates the ability of the model to detect all actual failure events. The F1-score balances these two metrics and is especially useful in imbalanced scenarios. Area Under the ROC Curve (AUC) is also reported to evaluate the model's ability to distinguish

ISSN: 3071-3315
PP: 01-11.

between healthy and failing states across all thresholds. For time-series models, additional metrics like Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used for forecasting accuracy.

Table 2: Selected Features Used For Model Training

Feature Name	Source	Type	Description
CPU_Temperature	Temperature Sensor	Continuous	Avg. CPU temperature over last 5 mins
Fan_Speed_Deviation	Cooling System	Continuous	Std. deviation of fan speed readings
Network_Latency_Trend	Network Logs	Continuous	Slope of latency over 10-minute window
Last_Maintenance_Age	Maintenance Logs	Numeric	Days since last repair or service
Error_Log_Frequency	System Logs	Discrete	Number of errors in last hour

6. SYSTEM DESIGN AND IMPLEMENTATION

6.1. Architecture of Predictive Maintenance Pipeline

The predictive maintenance pipeline is architected as a modular and scalable system, designed to ingest real-time telemetry data, preprocess it, feed it into trained ML models, and generate actionable alerts. The pipeline begins with data acquisition from sensors and monitoring platforms, followed by a preprocessing layer that handles cleaning, feature extraction, and normalization. The processed data is then passed to the inference engine, which hosts one or more trained ML models. Detected anomalies or predicted failure probabilities are forwarded to a decision-making layer, which triggers notifications or automatic maintenance actions. The entire system is containerized using technologies like Docker and orchestrated with Kubernetes for horizontal scalability.

6.2. Real-Time Data Ingestion and Inference (e.g., using Apache Kafka, Spark, TensorFlow Serving)

To enable real-time predictive capabilities, the system uses Apache Kafka as a high-throughput, distributed data ingestion layer. Kafka streams raw sensor and log data into the pipeline with minimal latency. Apache Spark Streaming or Flink is used for real-time preprocessing and feature transformation. The trained ML models are served via TensorFlow Serving or ONNX Runtime, enabling low-latency inference at scale. These services are deployed on cloud-native infrastructure, utilizing GPU acceleration where necessary, and are designed to support both batch and streaming prediction workloads.

6.3. Integration with Cloud Monitoring Platforms (e.g., Prometheus, Azure Monitor)

For seamless operational integration, the predictive maintenance system interfaces with existing cloud monitoring platforms. For example, Prometheus can be used to scrape metrics and feed them into the Kafka stream, while Azure Monitor or AWS CloudWatch can trigger events based on model predictions. This integration allows the system to function within existing observability stacks, making it easier to adopt in production environments. Alerts and insights generated by the ML models are pushed into dashboards (e.g., Grafana) or ticketing systems (e.g., PagerDuty, ServiceNow) for incident management.

ISSN: 3071-3315
PP: 01-11.

7. EXPERIMENTAL SETUP AND RESULTS

7.1. Dataset Description

The experimental evaluation is conducted using both synthetic and real-world datasets sourced from data center operations. The dataset includes temperature readings, fan RPM, CPU/GPU utilization, network throughput, error logs, and historical maintenance records. For confidentiality and reproducibility, open datasets such as Google Cluster Data or the ASHRAE building energy datasets may be supplemented. The data is collected over a period of several months, ensuring a diverse mix of normal and degraded operating states. The dataset is then preprocessed and labeled where possible, with anomalies marked using maintenance logs and downtime reports.

7.2. Benchmarking of Models on Prediction Accuracy and Latency

Each model is trained and tested under controlled conditions, and its performance is evaluated on two critical dimensions: prediction accuracy and inference latency. LSTM and GRU models demonstrate superior performance on long-term temporal trends, achieving high recall in predicting fan failures and thermal anomalies. XGBoost provides strong performance on tabular sensor data with excellent precision. Autoencoders and Isolation Forests show promise in detecting novel failure types. In terms of latency, traditional models (e.g., Random Forest) offer faster inference times, making them more suitable for real-time edge deployment, while deep learning models are more computationally intensive but offer better accuracy.

7.3. Case Studies: Cooling System Failure Prediction, Network Card Degradation

To validate the practical effectiveness of the models, two case studies are conducted. In the first, predictive models are trained to detect early signs of cooling system failure using temperature deltas, fan RPM trends, and historical anomalies. The model successfully identifies abnormal behavior patterns hours before temperature thresholds are violated. In the second case study, network card degradation is predicted based on packet loss, latency spikes, and retransmission errors. The ML model detects subtle performance declines that precede hardware failure, allowing timely replacement. These studies highlight the potential of predictive maintenance in reducing mean time to repair (MTTR) and avoiding critical incidents.

7.4. Comparison with Traditional Threshold-Based Monitoring

The proposed AI/ML-based approach is benchmarked against conventional threshold-based monitoring systems commonly used in cloud environments. Traditional systems generate alerts only when predefined limits are breached, often missing gradual degradations or false positives from noisy data. In contrast, the ML models provide early warning signals, even when all system metrics are technically within "safe" bounds. The result is a significantly reduced number of false alarms and improved lead time for maintenance actions. This demonstrates the superiority of intelligent systems in handling complex, nonlinear, and interdependent infrastructure behavior.

ISSN: 3071-3315
PP: 01-11.

8. CHALLENGES AND LIMITATIONS

8.1. Scalability in Multi-Tenant Environments

One of the primary challenges in implementing predictive maintenance at cloud scale is multi-tenancy. A single physical resource may host workloads from multiple clients with varying performance profiles, making it difficult to isolate faults and identify causality. Additionally, the volume of telemetry data from large-scale environments can quickly become overwhelming. Scalable architectures using distributed computing and federated learning may be needed, but these bring their own complexity in deployment and management.

8.2. Model Drift and Re-Training Frequency

Cloud infrastructure is dynamic, with constant updates in hardware, software, and operational policies. As a result, ML models may experience concept drift a decline in performance over time due to changes in data distributions. Without regular retraining, the models may become outdated and unreliable. Implementing automated retraining pipelines and model monitoring systems is essential, but it increases operational overhead and demands robust data versioning and labeling practices.

8.3. Security and Privacy Concerns

Collecting and processing infrastructure telemetry raises concerns related to security and data privacy. For instance, logs may contain sensitive configuration data or customer information that must be protected during processing and storage. Ensuring data encryption, access control, and compliance with data protection regulations (e.g., GDPR, ISO 27001) is vital. Furthermore, adversarial attacks on ML models such as feeding manipulated input data to avoid detection pose a growing risk and must be addressed with robust validation and monitoring.

8.4. Generalization to Unseen Infrastructure Types

Models trained on data from specific hardware configurations or environmental conditions may not generalize well to different infrastructure types. For example, a cooling failure predictor trained on one data center may perform poorly in another with different airflow design or hardware models. This lack of generalizability limits the scalability and portability of predictive maintenance systems. Techniques like transfer learning, domain adaptation, and model retraining on new environments are needed to address this challenge, but these approaches are still emerging and require further exploration.

9. FUTURE WORK

9.1. Use of Federated Learning across Multiple Data Centers

One of the key limitations in deploying predictive maintenance at scale is the challenge of accessing and centralizing data from geographically distributed data centers, especially when data privacy and compliance constraints are present. In future research, federated learning (FL) can be explored as a solution to this issue. FL allows multiple data centers to collaboratively train a shared global model while keeping raw data localized and secure on-site. This approach ensures compliance with regional data governance laws and significantly reduces the risk of data breaches. Moreover, by

ISSN: 3071-3315
PP: 01-11.

leveraging the diversity of infrastructure data from various locations, FL models can achieve better generalizability and robustness. Implementing FL across cloud environments would require optimized communication protocols, secure model aggregation techniques, and strategies to deal with heterogeneous data distributions (non-IID data).

9.2. Integration with Edge Computing for Localized Inference

With the growth of edge computing, much of the data generated by cloud infrastructure can be processed closer to the source, reducing latency and bandwidth usage. Future implementations of predictive maintenance should incorporate edge-based inference systems capable of performing real-time anomaly detection and early warning signal generation directly on edge devices or gateways. This would not only enhance response times but also reduce the load on centralized servers. Edge-integrated ML pipelines would be especially valuable for monitoring localized components such as cooling fans, power units, and network interface cards that generate high-frequency sensor data. Challenges include deploying lightweight, resource-efficient models that can operate under limited compute capacity, ensuring model updates across distributed nodes, and maintaining synchronization with the central cloud environment.

9.3. Autonomous Maintenance Orchestration Using RL Agents

Another promising avenue for future work is the development of autonomous maintenance systems driven by Reinforcement Learning (RL) agents. These agents could be trained to make optimal maintenance decisions—such as scheduling downtime, replacing components, or adjusting workloads—based on environmental feedback and learned policies. Unlike static rule-based systems, RL agents can continuously learn from operational outcomes, adapting to changing workloads, hardware aging, and environmental conditions. For instance, an RL-based orchestration system could learn to delay or expedite maintenance actions depending on workload criticality and component risk level. This approach would also enable dynamic resource allocation, where workloads are intelligently migrated away from nodes predicted to fail. While promising, RL systems in this context require careful reward design, large-scale simulation environments for safe training, and mechanisms to ensure stability and safety in real-world operations.

10. CONCLUSION

10.1. Summary of Findings

This research has presented a comprehensive exploration of AI/ML techniques for predictive maintenance within cloud infrastructure, focusing on critical components such as cooling systems and networking hardware. By leveraging supervised models like Random Forest and XGBoost, unsupervised methods such as Autoencoders and Isolation Forest, and temporal models like LSTM and GRU, we demonstrated that AI systems can successfully detect early signs of equipment degradation and predict failures before they occur. Our work also highlighted the importance of high-quality data preprocessing, the role of real-time data pipelines, and the integration with cloud-native observability tools for effective deployment.

ISSN: 3071-3315
PP: 01-11.

10.2. Impact on Cloud Reliability and Cost-Efficiency

The adoption of predictive maintenance driven by AI/ML significantly enhances cloud reliability by minimizing unexpected outages and improving response time to potential failures. Moreover, it contributes to cost-efficiency by optimizing maintenance schedules, reducing unnecessary component replacements, and extending the usable lifespan of infrastructure. The ability to act proactively rather than reactively translates into better service availability, improved SLA adherence, and reduced operational expenses. As cloud infrastructure continues to scale in complexity and size, such intelligent systems will become essential to maintaining resilience and efficiency.

10.3. Final Thoughts on Industrial Applicability

From a practical standpoint, the findings of this research are highly relevant to both cloud service providers and enterprises operating private or hybrid cloud environments. By providing a scalable and modular architecture, this approach can be integrated into existing monitoring and DevOps pipelines with minimal disruption. While challenges such as data scarcity, privacy, and model maintenance remain, the long-term benefits of predictive maintenance when combined with federated learning, edge computing, and autonomous orchestration make it a promising field for both academic research and industrial deployment. Ultimately, AI-powered maintenance systems have the potential to redefine the operational backbone of modern cloud ecosystems.

REFERENCES

[1] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
[2] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD, 785–794. https://doi.org/10.1145/2939672.2939785
[3] Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. Proceedings of MLSDA, 4–11. https://doi.org/10.1145/2689746.2689747
[4] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. IEEE ICDM, 413–422. https://doi.org/10.1109/ICDM.2008.17
[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
[6] Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. https://arxiv.org/abs/1406.1078
[7] Sculley, D., et al. (2015). Hidden technical debt in machine learning systems. NeurIPS, 28. https://papers.nips.cc/paper_files/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
[8] Zaharia, M., et al. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56–65. https://doi.org/10.1145/2934664
[9] Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9
[10] Mao, H., et al. (2016). Resource management with deep reinforcement learning. HotNets XV. https://doi.org/10.1145/2987550.2987555
[11] Ahmad, S., et al. (2017). Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262, 134–147. https://doi.org/10.1016/j.neucom.2017.04.070
[12] Feki, M. A., et al. (2017). Smart factory architecture for predictive maintenance. Procedia Computer Science, 109, 1150–1155. https://doi.org/10.1016/j.procs.2017.05.446