Comparative Study of Analyzing Machine Learning-Based Predictive Maintenance Using Cloud Computing and Edge Computing

Panupong Miura, Worawich Fakchuak, Sarut Attawanich
and Assoc. Prof. Dr. Amphawan Julsereewong

Mechatronics and Automation Engineering Program (Major of Automation)
Department of Instrumentation and Control Engineering, School of Engineering, KMITL

Abstract

This academic article presents comparative study results for possibly implementing predictive maintenance based on machine learning. Not only seven different machine learning algorithms to detect abnormalities in production machines are explained, but also two system architectures for machine learning-based predictive maintenance analysis are focused. The first system architecture is based on cloud computing from AWS (Amazon Web Services), while the latter is based on edge computing. In addition, the comparison results, which may be useful for further implementation, in terms of architecture, latency, data management capability, data processing capability, scalability, system reliability, security, and cost are included.

Keywords : AWS, Cloud Computing, Edge Computing, Machine Learning, Predictive Maintenance

1. Introduction

In the present, various industries have continuously developed their maintenance systems. Predictive maintenance systems are among those that are continuously evolving, not only improving operational efficiency but also reducing machine downtime. Because of the predictive maintenance processing units, the system architectures must be considered. This academic article aims to compare cloud computing and edge computing processing to enhance understanding and improve digital infrastructure.

2. Predictive Maintenance

Predictive maintenance is a maintenance approach that uses collected data for analysis. These data are then compared with standard values for the machine with the various machine learning techniques. Then, the predictive maintenance systems can predict when the machine is likely to be down or malfunction and operators or engineers can maintain the machine before it breaks down.

This academic article aims to present predictive maintenance using machine learning [2-4], which involves receiving data from machine sensors to enable machine learning to learn and predict when maintenance should be performed. Due to the different conditions of the collected data, the machine learning types can be divided into two cases.:

1) Supervised learning is suitable for data that include maintenance logs or have experienced machinery failure events.

A. Julsereewong

2) Unsupervised Learning is suitable for data that do not include maintenance logs or have not experienced machinery failure events.

Here are the Machine Learning models suitable for predictive maintenance:

1) Decision Tree Classifier: It is a tree-like model with decision-making branches and possible outcomes at each node, representing features, decisions, and results.

2) Random Forest Classifier: It builds multiple decision trees during training and averages their results to improve accuracy and reduce overfitting.

3) Support Vector Classifier: It finds hyperplanes and maximum margins to separate different classes in feature space, effective for high-dimensional data and memory-efficient.

4) Logistic Regression: It classifies the outcomes by using a logistic function to predict probabilities.

5) K Nearest Neighbors: It determines class separation by choosing the K closest data points, where K is defined by the model creator.

6) Gaussian Naive Bayes: It uses Bayes' theorem for class separation.

7) Simple Neural Network: It works like a neural system, processing input data through hidden layers to produce results.

The accuracy of each model can be measured by finding TP (True Positive), FP (False Positive), FN (False Negative), and TN (True Negative) according to Figure 1 and then calculating the Precision, Recall, and F1 Score by using Equations (1)-(3), respectively.

Variants Called by the Algorithm	Reference variant set: Positive	Reference variant set: Negative
Positive	True Positive (TP) Correct variant allele or position call.	False Positive (FP) Incorrect variant allele or position call.
Negative	False Negative (FN) Incorrect reference genotype or no call.	True Negative (TN) Correct reference genotype or no call.

Figure 1 Confusion matrix [2]

The equation for calculating Precision can be stated as

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

The equation for calculating Recall can be stated as

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

The equation for calculating the F1 Score can be stated as

$$F1\ Score = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} \tag{3}$$

3. Cloud Computing

A cloud system is a service model that provides leased computing resources for data storage and processing via the Internet. The architecture of cloud processing systems often involves centralization, as depicted in Figure 2.

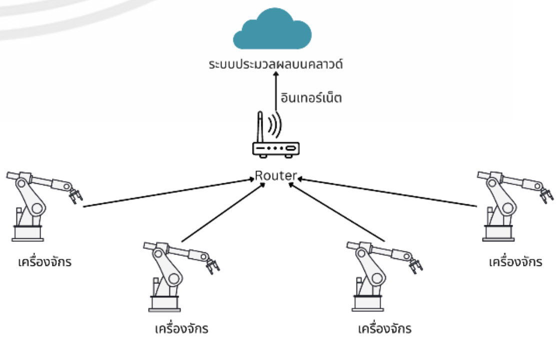

Figure 2 The architecture of cloud processing systems.

Using cloud-based systems helps reduce hardware burden in industries, are easily controllable, and can be accessed for configuration over the Internet. They also enhance scalability by allowing expansion without concerning hardware processing needs. This enables engineers or stakeholders to increase data's value across various aspects.

The cloud system referred to in this article utilizes Amazon Web Services (AWS). AWS provides users with cloud space tailored to their needs. The data flows [6] can be created as shown in Figure 3. This academic article uses 12 AWS services, including:

1) AWS IoT Core [7] is a service that enables IoT (Internet of Things) devices to send data to AWS cloud efficiently, allowing for effective management of connected devices.

2) AWS S3 [8] is a scalable storage service for data storage.

3) AWS Kinesis Data Streams [9] is a service that can analyze or forward data in real time.

4) AWS SageMaker [10] is a service that aids in the building, training, and deploying specifically for machine learning tasks.

5) AWS Lambda [11] is a service that performs calculations without the need for a server, allowing programs to run without engineers or relevant personnel having to manage servers.

6) AWS Redshift [12] is a service that helps manage and analyze data using SQL queries.

7) AWS Step Functions [13] is a service that manages the operations of various services within AWS.

8) AWS IoT Events [14] is a service that manages detection and responds to various predefined events.

9) AWS SNS [15] is a service that sends notifications and messages to various systems and applications, supporting various transmission protocols and Pub/Sub messaging.

10) AWS CloudWatch [16] is a service for monitoring and observing real-time data and resources of applications.

11) AWS QuickSight [17] is a service for observing data using dashboards and interactive reports.

12) AWS API Gateway [18] is a service that manages the API (Application Programming Interface) in a RESTful manner.

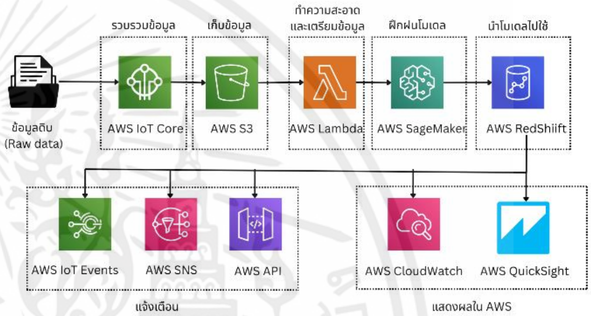

Figure 3 The structure for managing data on AWS.

4. Edge Computing

Edge computing [19] refers to placing processing units close to the data source. For example, renting cloud systems in the user's geographical region or deploying processing systems within a local network. This academic article discusses edge computing networks specifically for processing through the edge. Creating a system for processing through the edge can be done as shown in Figure 4, without external internet connectivity. Management of various data can also be done within separate processing systems following the principles outlined in Figure 5.

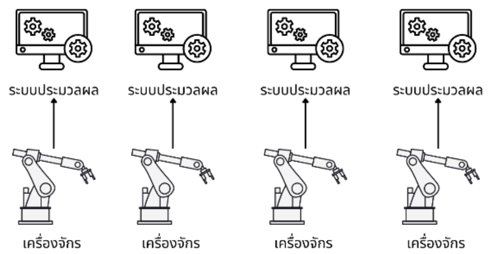

Figure 4 The architecture of edge processing systems.

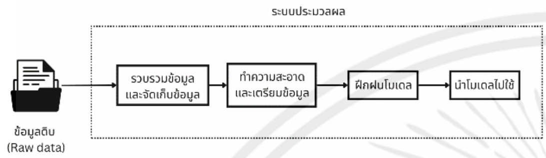

Figure 5 The structure for managing data through the edge.

5. The comparison between cloud computing and edge computing. [20-24]

The comparison between cloud processing and edge processing can be divided into 8 topics, namely:

1) Architecture

In cloud processing, data is sent to a centralized cloud processing center, leading to a centralized architecture. On the other hand, edge processing involves systems processing data near the data source, resulting in a decentralized architecture.

2) Latency

Edge processing has lower latency compared to cloud processing because data doesn't need to be sent to a central data center but is processed closer to the data source. This results in lower latency in data reception and transmission.

3) Data Handling

Edge processing has limited data handling capabilities compared to cloud processing because edge processing requires processing systems to be close to the data source. This results in certain limitations in various aspects, such as memory or processing device limitations in some cases. In contrast, cloud processing can easily scale data handling capabilities.

4) Computational Capacity

Edge processing capabilities range from low to moderate when compared to cloud processing because edge processing has a distributed architecture, which means the processing system doesn't necessarily require high capabilities.

5) Scalability

Cloud processing has excellent scalability because system efficiency can be increased by adding resources to the cloud, such as servers, storage space, or processing capabilities. In contrast, the scalability of edge processing has limitations due to considerations like space for installing edge devices, existing systems, and the integration of new and old technologies.

6) Reliability

Because cloud processing involves processing data over the Internet, the reliability of the system depends on the Internet and the rented cloud system. However, for edge processing, reliability depends on the internal communication system's setup.

7) Security

Securing cloud processing is relatively difficult as it involves using internet services connected to external sources and data within the cloud system. This results in edge processing being more secure due to the use of localized computer networks.

8) Cost

The cost of cloud processing depends on the utilization of resources in the cloud, including the volume of data. On the other hand, the cost of edge processing depends on the number of processing units installed in that area and the maintenance cost of the equipment.

Table 1 The comparison between cloud processing and edge processing.

Criteria	**Cloud Computation**	**Edge Computation**
Architecture	Centralized	Decentralized
Latency	High	Low
Data Handling	It can handle large-scale data.	There are limitations in managing large-scale data.
Computational Capacity	High	Low to Medium
Scalability	High	Low
Reliability	It depends on the internet system.	It depends on the processing system setup.
Security	Security is managed by the cloud owner and the internet service provider.	Security is managed by setting up internal systems.
Cost	It depends on the amount of data and the resources rented.	It depends on the edge devices and maintenance costs.

6. Conclusion

Predictive maintenance is a crucial approach in industrial machinery upkeep, ensuring maximum efficiency and reasonable damage reduction. Implementing this concept in industry can be efficient, but it requires clear indicators or variables of machine anomalies. In developing a reliable machine learning system capable of accurate predictions, data preparation is critical. Historical data on errors or past maintenance can enhance system expertise and accuracy. With such data, supervised machine learning models can significantly improve prediction accuracy. However, even without such data, unsupervised machine learning remains a viable option, albeit with less data preparation complexity, but prediction efficacy may be lower in some cases.

Regarding the choice between cloud and edge processing systems, decisions should align with user needs and objectives as they differ significantly. Cloud processing offers high flexibility and easy data access over the internet but comes with high costs. On the other hand, edge processing is faster and more secure but has limitations in system scalability or enhancement. Therefore, selecting based on suitability and business objectives is crucial in decision-making.

References

[1] Predictive maintenance, Premium Equipment&Engineer [Online] https://www.premium.co.th/2022/10/26/3p-maintenance/ [Accessed: Jan. 5, 2024]

[2] Machine Learning approach for Predictive Maintenance in Industry 4.0 [Online] https://ieeexplore.ieee.org/document/8449150/ [Accessed: Jan. 5, 2024]

[3] Comparison of Different Machine Learning Algorithms for Predictive Maintenance [Online]

https://ieeexplore.ieee.org/document/10080334 [Accessed: Mar. 23, 2024]

[4] Machine learning [Online] https://aitskadapa.ac.in/e-books/AI&ML/MACHINE%20LEARNING/Machine%20Learning%20(%20etc.)%20(z-lib.org).pdf [Accessed: Jan. 5, 2024]

[5] Cloud system, Quick serv Cloud system, [Online] https://www.quickserv.co.th/knowledge-base/solutions/Cloud-system/ [Accessed: Dec. 20, 2023]

[6] Using AWS IoT for Predictive Maintenance | The Internet of Things on AWS – Official Blog [Online] https://aws.amazon.com/blogs/iot/using-aws-iot-for-predictive-maintenance/ [Accessed: Mar. 23, 2024]

[7] AWS IoT Core [Online]https://docs.aws.amazon.com/pdfs/iot/latest/developerguide/iot-dg.pdf/ [Accessed: Jan. 30, 2024]

[8] Amazon Simple Storage Service - User Guide [Online]https://docs.aws.amazon.com/pdfs/AmazonS3/latest/userguide/s3-userguide.pdf/ [Accessed: Jan. 30, 2024]

[9] Amazon Kinesis Data Streams - Developer Guide [Online]https://docs.aws.amazon.com/pdfs/streams/latest/dev/kinesis-dg.pdf/ [Accessed: Feb. 15, 2024]

[10] Amazon SageMaker - Developer Guide [Online] https://docs.aws.amazon.com/pdfs/sagemaker/latest/dg/sagemaker-dg.pdf/ [Accessed: Feb. 15, 2024]

[11] AWS Lambda - Developer Guide [Online] https://docs.aws.amazon.com/pdfs/lambda/latest/dg/lambda-dg.pdf/ [Accessed: Feb. 15, 2024]

[12] Amazon Redshift - Management Guide [Online] https://docs.aws.amazon.com/pdfs/redshift/latest/mgmt/redshift-mgmt.pdf#welcome/ [Accessed: Feb. 26, 2024]

[13] AWS Step Functions - Developer Guide [Online] https://docs.aws.amazon.com/pdfs/step-functions/latest/dg/step-functions-dg.pdf#welcome/ [Accessed: Mar. 2, 2024]

[14] AWS IoT Events - Developer Guide [Online]https://docs.aws.amazon.com/pdfs/iotevents/latest/developerguide/iotevents-dg.pdf/ [Accessed: Mar. 2, 2024]

[15] Amazon Simple Notification Service - Developer Guide [Online]https://docs.aws.amazon.com/pdfs/sns/latest/dg/sns-dg.pdf/ [Accessed: Mar. 2, 2024]

[16] Amazon CloudWatch - User Guide [Online] https://docs.aws.amazon.com/pdfs/AmazonCloudWatch/latest/monitoring/acw-ug.pdf/ [Accessed: Mar. 2, 2024]

[17] Amazon QuickSight - Developer Guide [Online] https://docs.aws.amazon.com/pdfs/quicksight/latest/developerguide/amazon-quicksight-dg.pdf/ [Accessed: Mar. 11, 2024]

[18] Amazon API Gateway - Developer Guide [Online]https://docs.aws.amazon.com/pdfs/apigateway/latest/developerguide/apigateway-dg.pdf#welcome/ [Accessed: Mar. 11, 2024]

[19] Edge Computing: Classification, Applications, and Challenges [Online] https://ieeexplore.ieee.org/document/9445331/ [Accessed: Jan. 5, 2024]

[20] Edge Computing vs. Cloud Computing [Online] https://getstream.io/glossary/edge-versus-cloud-computing/ [Accessed: Jan. 5, 2024]

[21] Edge Computing vs. Cloud Computing: 10 Key [Online] https://www.spiceworks.com/tech/cloud/articles/edge-vs-cloud-computing/ [Accessed: Jan. 5, 2024]

[22] A Comparative Study on Cloud Computing and Edge Computing with its Applications [Online] https://www.researchgate.net/publication/358404145_A_Comparative_Study_on_Cloud_Computing_and_Edge_Computing_with_its_Applications/ [Accessed: Jan. 5, 2024]

[23] Comparative Study of Cloud Computing and Edge Computing: Three-Level Architecture Models and Security Challenges [Online] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3908915/ [Accessed: Jan. 5, 2024]

[24] A Comparative Study on Cloud and Edge Computing: A Survey on Current Research Activities and Applications [Online] https://ieeexplore.ieee.org/document/10169821/ [Accessed: Jan. 5, 2024]