



## Article

# Optimization Techniques and Evaluation for Building an Integrated Lightweight Platform for AI and Data Collection Systems on Low-Power Edge Devices

Woojin Cho , Hyungah Lee  and Jae-hoi Gu \*

Energy Environment IT Convergence Group, Plant Engineering Center, Institute for Advanced Engineering, Yongin 17180, Republic of Korea; wooju\_1@iae.re.kr (W.C.); lhaeve@iae.re.kr (H.L.)

\* Correspondence: jaehoi@iae.re.kr; Tel.: +82-31-330-7870

**Abstract:** Amidst an energy crisis stemming from increased energy costs and the looming threat of war, there has been a burgeoning interest in energy conservation and management worldwide. Industrial complexes constitute a significant portion of total energy consumption. Hence, reducing energy consumption in these complexes is imperative for energy preservation. Typically, factories within similar industries aggregate in industrial complexes and share similar energy utilities. However, they often fail to capitalize on this shared infrastructure efficiently. To address this issue, a network system employing a virtual utility plant has been proposed. This system enables proactive measures to counteract energy surplus or deficit through AI-based predictions, thereby maximizing energy efficiency. Nevertheless, deploying conventional server systems within factories poses considerable challenges. Therefore, leveraging edge devices, characterized by low power consumption, high efficiency, and minimal space requirements, proves highly advantageous. Consequently, this study focuses on constructing and employing data collection and AI systems to utilize edge devices as standalone systems in each factory. To optimize the AI system for low-performance edge devices, we employed the integration-learning AI modeling technique. Evaluation results demonstrate that the proposed system exhibits high stability and reliability.

**Keywords:** edge device; low-power computing; virtual plant utility; energy management; AI; deep learning



**Citation:** Cho, W.; Lee, H.; Gu, J.-h. Optimization Techniques and Evaluation for Building an Integrated Lightweight Platform for AI and Data Collection Systems on Low-Power Edge Devices. *Energies* **2024**, *17*, 1757. <https://doi.org/10.3390/en17071757>

Academic Editor: Paulo Santos

Received: 20 February 2024

Revised: 30 March 2024

Accepted: 3 April 2024

Published: 6 April 2024



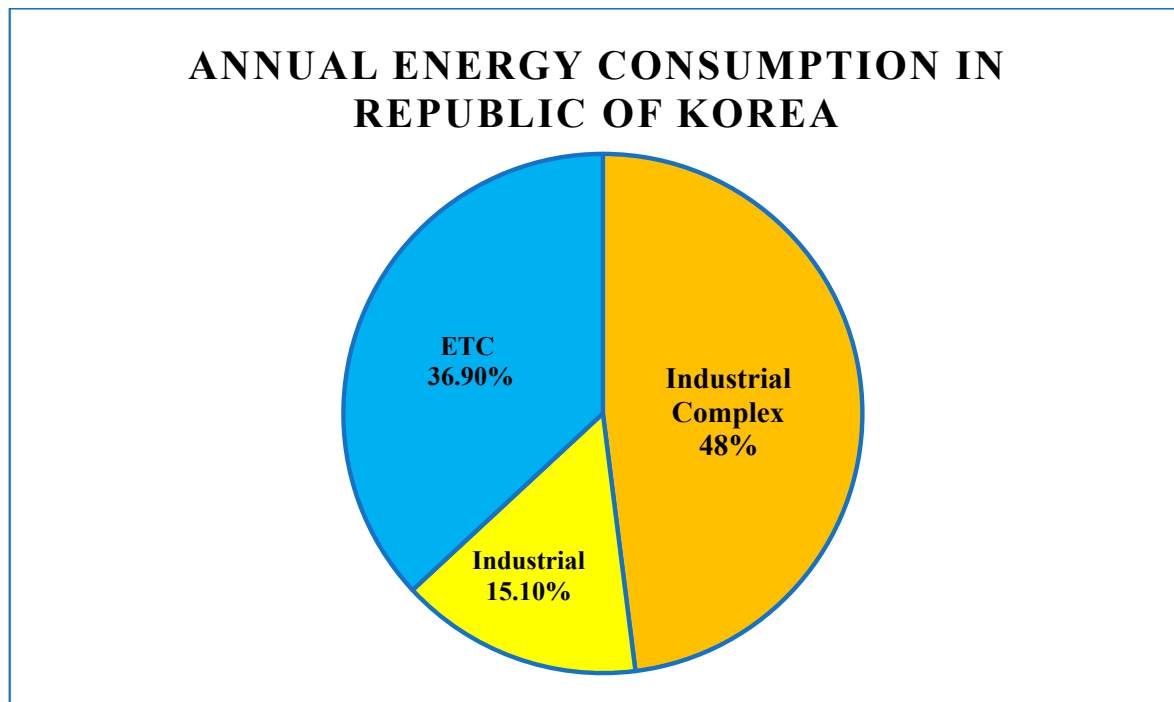
**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Amidst an energy crisis fueled by escalating energy costs and the specter of war, global interest in energy conservation and management has surged. In response, the government of the Republic of Korea has initiated various efforts to conserve energy. One such initiative involves the implementation of the Factory Energy Management System (FEMS), mandated by the Third Energy Master Plan for energy-intensive facilities consuming more than 100,000 tons of oil equivalent (TOE), effective from 2025 onward. Furthermore, the Korean government is actively promoting FEMS adoption among factories consuming less than 100,000 TOE [1]. Notably, energy-related policies in Korea are predominantly industry-centric, given that the industrial sector, particularly industrial complexes, accounts for a significant portion of the nation's energy consumption. As shown in Figure 1, industrial complexes alone contributed to approximately 48% of the Republic of Korea's total energy consumption [2]. Hence, it becomes imperative to implement energy-saving strategies and management protocols within these energy-intensive industrial complexes to bolster energy efficiency and conservation.

Within these industrial complexes, factories specializing in similar industries often coalesce. Despite this proximity, many factories fail to leverage shared energy utilities effectively. Instead, they resort to self-generation or individual contracts with energy production entities. This decentralized approach to energy procurement proves disadvantageous as it

precludes the preparedness for potential issues arising from energy surpluses or shortages. Moreover, energy trade under such arrangements typically occurs through contracts with high unit prices [3].



**Figure 1.** Energy consumption in the Republic of Korea in 2018.

A network system has been devised to address these challenges by leveraging a virtual utility plant. Within this network system, factories within an industrial park that share the same energy utilities establish a stable supply chain, utilizing common facilities to stabilize the provision of energy utilities. Moreover, this concept facilitates energy trade between energy-producing companies and consumer companies by forecasting energy demand and supply and employing routing algorithms. This approach optimizes energy utilization and minimizes waste [4].

To effectively utilize an energy utility-sharing network such as the VUP network system, artificial intelligence AI-based techniques are essential for predicting energy production and demand across common facilities, energy production, and trading entities. The conventional method for deploying these techniques involves each company utilizing its own on-premise servers.

However, deploying FEMS or VUP simulators in industrial complexes using conventional server systems necessitates additional equipment, such as air conditioners and dehumidifiers, to regulate temperature and humidity for server management. Nonetheless, this approach consumes a significant amount of energy. Moreover, given the spatial constraints inherent in many factories within industrial complexes, setting up these systems using conventional server infrastructure poses numerous limitations.

To address these challenges, a method such as Software-as-a-Service (SaaS) can be considered. While SaaS offers easy setup with low initial costs, it entails ongoing fixed expenses [5]. Additionally, concerns about security may arise due to the sharing of operational and sensor data among factories [6,7].

For a more fundamental solution, ARM-based edge devices present a viable option. This approach entails replacing conventional server systems with ARM-based edge devices. Unlike conventional servers, which consume several thousand watts of power, each ARM-based device consumes only tens of watts. Their lower power consumption reduces the occurrence of overheating issues, thereby easing device management burdens. Further-

more, ARM-based devices occupy significantly less space, approximately 40 to 50 times less than conventional server systems, thus optimizing space utilization. However, ARM-based embedded devices typically offer lower performance than conventional server systems, impacting processing throughput. Additionally, they utilize ARM-based application processors (APs) rather than conventional x86- or AMD64-based central processing units (CPUs). This imposes constraints and limits the scope of support they can provide.

Notably, AI learning often requires high graphics processing unit (GPU) performance, which has historically posed challenges for embedded devices with lower performance capabilities.

Recent studies have been exploring methods to overcome various limitations, including constraints related to space, environment, and energy, using edge devices. A key focus of this research is leveraging the capabilities of low-performance edge devices, strategically positioned closer to data collection points [8]. Related studies examine strategies for distributing data processing by relocating segments of existing AI systems to edge devices. However, these efforts primarily aim to complement existing systems rather than replace them entirely. Consequently, they neither operate autonomously nor fully resolve challenges such as external data access.

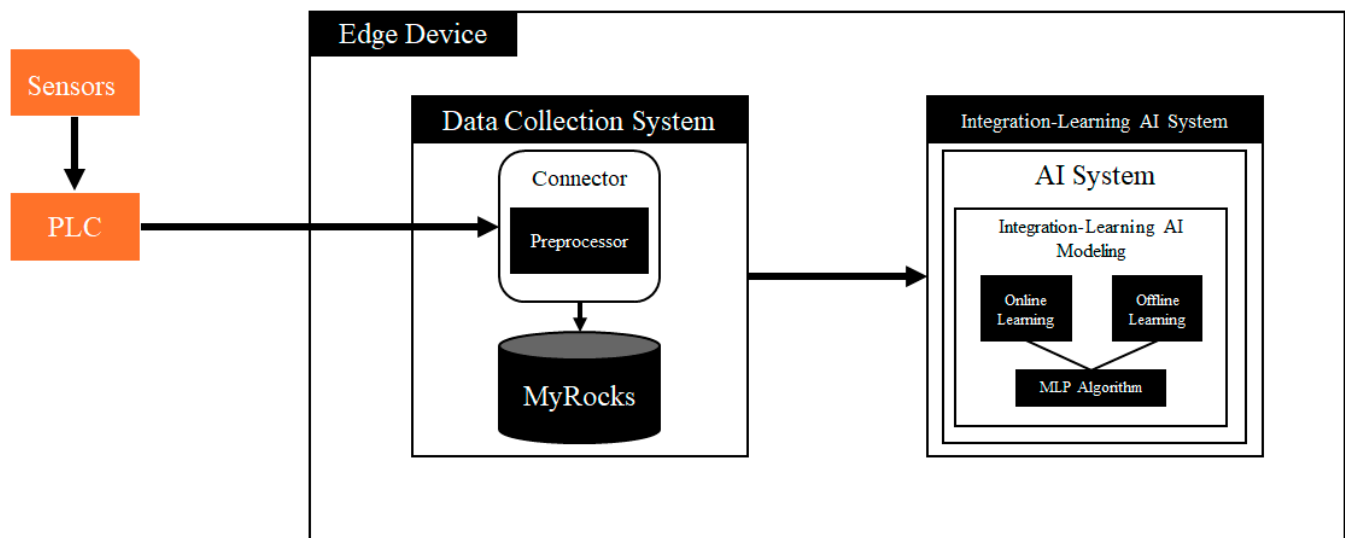
Additionally, the utilization of TinyML, which offers benefits such as low power consumption, real-time operation, and high accuracy, has been studied [9,10]. Nonetheless, these devices may struggle with processing large volumes of data in real time owing to their constrained specifications. Specifically, they may not be suitable for environments characterized by significant fluctuations in data production, necessitating frequent model updates.

Another avenue of research involves developing predictive systems by optimizing clustering algorithms on edge devices [11]. This line of inquiry explores the potential of employing GPUs such as the Jetson Nano to deploy AI systems on edge devices, as opposed to conventional predictive systems. However, implementing this approach on edge devices with limited GPU performance poses significant challenges. These studies predominantly focus on refining artificial intelligence algorithms and often overlook aspects such as data collection, thereby limiting the advancement of comprehensive systems exclusively utilizing edge devices.

Furthermore, active research is underway on refining AI models. By employing hyperparameter auto-tuning, opportunities for enhancing AI model performance can be identified [12,13]. However, owing to the necessity of frequent model reconstruction caused by significant production fluctuations, this method proves unsuitable for edge devices owing to the high computational burden. Self-adaptive deep learning techniques can more effectively accommodate production fluctuations [14]. Nevertheless, self-adaptive deep learning is also unsuitable for edge devices with limited specifications owing to high computational demands and the potential for overfitting issues.

Additionally, studies have aimed at implementing a recommendation system using AWS as a backend platform for edge devices [15]. However, this study also explores how to utilize edge devices as supplementary tools rather than directly addressing various limitations.

Herein, we introduce an integrated lightweight platform, as depicted in Figure 2, aimed at addressing the limitations observed in prior studies. The integrated lightweight platform is designed to facilitate the independent operation of systems running on existing server infrastructure on edge devices. Sensor data collected within the platform were gathered using a programmable logic controller (PLC). This collected data underwent conversion and preprocessing utilizing the connector and preprocessor modules. Subsequently, the data were stored using a relational database management system known as MyRocks 5.6, which employs a key-value store as its backend storage engine [16].



**Figure 2.** Integrated lightweight platform on low-power edge devices architecture.

Utilizing the data stored in MyRocks, an AI system was constructed by employing integration-learning AI modeling techniques to forecast future supply and demand in factory settings. Integration-learning AI modeling utilizes online learning, characterized by lower computational costs compared to alternative algorithms, to adapt to changing data patterns in response to orders. If model retraining becomes necessary due to evolving conditions, the AI model can be retrained using offline learning techniques.

To alleviate the performance limitations of edge devices, delay techniques were employed to mitigate prediction-related load, facilitate data collection, and minimize platform interruptions, thereby enhancing stability.

To the best of our knowledge, no prior research has been conducted on developing a comprehensive system—from data collection to prediction—solely utilizing edge devices without reliance on networks or external devices.

This study conducted energy utility predictions using edge devices, aiming to develop a practical prediction system that can substitute conventional systems requiring high computing power. The reliability of edge devices was assessed by comparing their prediction duration with that of conventional systems. This research offers the potential to enhance conventional systems, which consume significant energy, by integrating edge devices. Moreover, this study contributes to the development of a lightweight platform and the improvement of edge device reliability.

## 2. Background

### 2.1. Data Collection System

A traditional method of data collection is through a process known as data acquisition (DAQ). This method encompasses converters, sensors, data collection, and programmable logic controllers (PLCs) [17], albeit in a limited sense. However, this traditional notion of data collection fails to address the expanded scope of modern concepts, which include big data, AI, high-performance computing, and data processing. Consequently, contemporary data collection systems encompass broader functionalities, such as storing data using programs such as databases, preprocessing data for applications such as AI, and conducting real-time data processing. Data collection systems, now conceptualized in this expanded manner, play a pivotal role in systems reliant on data collection. Consequently, efforts are underway to enhance the efficiency of such data collection systems by decentralizing database and data preprocessing tasks to edge devices, thereby enabling more tasks to be performed at the endpoint level.

## 2.2. Database Management System

Data collected through sensors and PLCs in the data collection system necessitate storage. Owing to the ease of managing and utilizing data with a database, this approach is often preferred over storing data in files for future reference and management purposes. Database management systems are not only integral to data collection systems but are also extensively utilized for various processes such as AI prediction and data analysis.

Hence, a database plays a crucial role in data preservation and management. Historically, server systems were commonly employed for this purpose. However, in this study, a database management system was implemented on edge devices to address environmental constraints, such as limited space and operational conditions, as well as to mitigate power consumption issues. This study contributes to evaluating the reliability of database management systems on edge devices by assessing the operation of the AI system alongside the utilization of a database management system in embedded devices.

## 2.3. Deep Learning

An artificial neural network (ANN) with multiple hidden layers is referred to as a deep neural network (DNN), with deep learning algorithms responsible for training DNNs [18,19].

Deep learning is a subset of machine learning, which focuses on discerning the relationship between input and output data. It analyzes and identifies data features to construct a model, subsequently used for predicting new data. Furthermore, deep learning employs an ANN to process input data and forecast outcomes.

The term “deep” in “deep learning” signifies the presence of numerous stacked layers within the ANN. With this layered architecture, deep learning extracts features from extensive datasets, thereby yielding high accuracy.

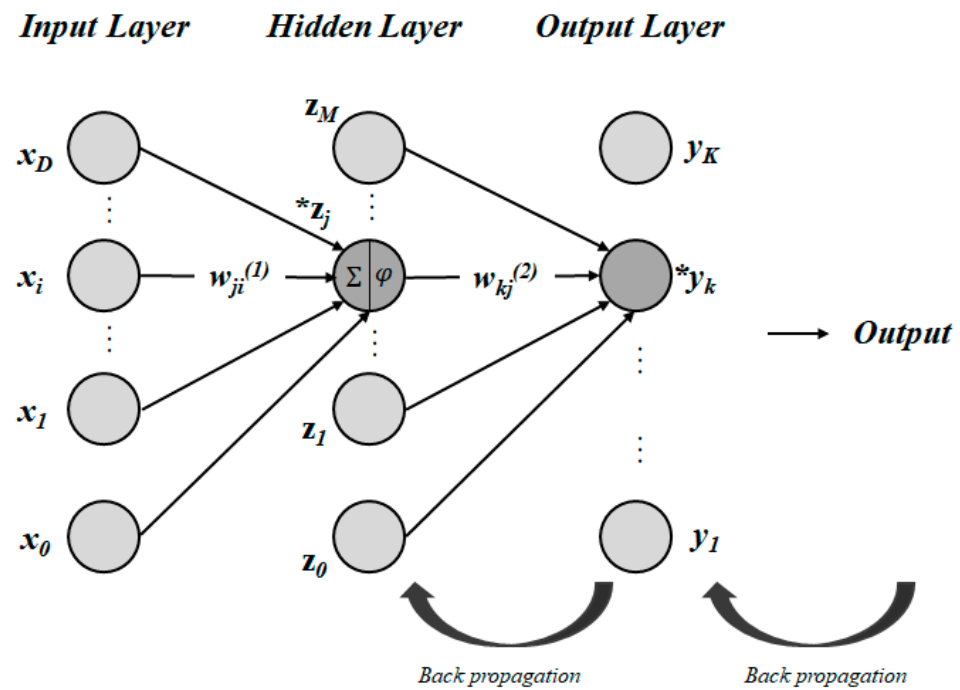
Prominent deep learning models encompass the multi-layer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN) [20,21].

### 2.3.1. MLP

MLP is a type of ANN structured with multiple layers of neurons. It serves as an algorithm capable of modeling complex non-linear relationships and addresses the limitations inherent in single-layer perceptrons. Consequently, MLP is efficiently applied to classification or regression problems and is one of the most widely utilized ANN algorithms.

MLP comprises an input layer, responsible for receiving input data, an output layer, which generates final predictions or output results, and hidden layers situated between the input and output layers. These hidden layers encompass multiple neurons interconnected with weights. Additionally, MLP operates as a feed-forward neural network (FFNN), with computations progressing from the input layer towards the output layer. Results are computed using the input data weights and activation function, with the process reiterated until reaching the output layer. Figure 3 depicts the computational process of MLP.

Each datum entering the input layer undergoes weighting using the connections to each node. Subsequently, the data pass through the  $h(\ )$  function and are transmitted as input data, denoted as  $z$ , to the hidden layer. Typically, the logistic sigmoid function or hyperbolic tangent function serves as the  $h(\ )$  function within the hidden layer. Similarly,  $z$  is weighted using the connections to the output layer, and the resulting values are transformed through the  $f(\ )$  function to produce the output. In regression problems, MLP outputs results as they are. However, for binary classification problems with two classes, 0 and 1, the results are processed through a sigmoid function. For multi-class problems, the results undergo transformation via the Softmax function.



$$z_j = h \sum_{i=0}^D w_{ji}^{(1)} x_i$$

$h$  : Activation Function (ex. Logistic sigmoid, Tanh)

$$y_k = f \left( \sum_{j=0}^M w_{kj}^{(2)} z_j \right) = \left( \sum_{j=0}^M w_{kj}^{(2)} h \left( \sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right)$$

$f(\cdot) \rightarrow I(\cdot)$  Identity function for regression

$\rightarrow \sigma(\cdot)$  Sigmoid function for binary classification

$\rightarrow \text{Softmax}$  function for multiclass classification

**Figure 3.** Calculation process of MLP.

### 2.3.2. Offline Learning

Offline learning, also known as batch learning, is a training approach that utilizes all available data for training. This method demands considerable time and resources as all data are learned simultaneously. Nonetheless, it boasts high accuracy and reliability since AI learning occurs across the entire dataset. Moreover, offline learning is efficient as it conducts matrix operations in a single batch, facilitating stable convergence of AI models.

### 2.3.3. Online Learning

Online learning, meanwhile, swiftly updates models by training them with incoming data, requiring fewer resources compared to batch learning. It involves updating the model with each execution or specific unit, enabling prompt adaptation to dynamic data changes. However, this method is susceptible to drawbacks such as noise, computational overhead, and overfitting.

## 3. AI System

The primary objective of an AI system based on edge devices for factories is to leverage AI for real-time predictions using collected data.



Hence, this section delves into the rationale behind selecting an AI model that operates on edge devices. Furthermore, it presents an architecture and methodology for integration-learning AI modeling as a benchmark for AI systems on edge devices.

### 3.1. AI Model

In this study, a previously researched AI algorithm served as the machine learning algorithm [22].

Two evaluation metrics,  $R^2$  and CvRMSE, were employed to assess the validity of the prediction models using the MLP and support vector regression (SVR) algorithms, respectively. For the SVR model, evaluation metrics were examined across three different kernels: linear, radial basis function network (RBF), and polynomial.

Based on the verification results of the models, the MLP demonstrated an  $R^2$  of 0.84 and CvRMSE of 17.35% for predicting electricity consumption, and an  $R^2$  of 0.88 and CvRMSE of 12.52% for predicting liquefied natural gas (LNG) consumption. The SVR model, when utilizing the linear kernel, exhibited an  $R^2$  of 0.72 and CvRMSE of 21.59% for predicting electricity consumption, and an  $R^2$  of 0.82 and CvRMSE of 21.59% for predicting LNG consumption. When employing the RBF kernel, the SVR model showed an  $R^2$  of 0.75 and CvRMSE of 20.52% for predicting electricity consumption, and an  $R^2$  of 0.88 and CvRMSE of 17.01% for predicting LNG consumption. Finally, with the polynomial kernel, the SVR model yielded an  $R^2$  of 0.71 and CvRMSE of 22.10% for predicting electricity consumption, and an  $R^2$  of 0.82 and CvRMSE of 21.58% for predicting LNG consumption.

Table 1 summarizes the results of the prediction models for each applied algorithm.

**Table 1.** Comparison of results of prediction models.

		MLP	SVR		
			Linear	RBF	Polynomial
Electricity	$R^2$	0.84	0.72	0.75	0.71
	CvRMSE	17.35%	21.59%	20.52%	22.10%
LNG	$R^2$	0.88	0.82	0.88	0.82
	CvRMSE	12.52%	21.59%	17.01%	21.58%

Based on the results, the model's validity is highest when applying MLP for predicting both electricity and LNG consumption.

Furthermore, MLP is notable for its capacity to update weights. This study aimed to ensure the reliability of the AI system's construction by reflecting real-time data weights and reconstructing the system when model retraining is necessary. Hence, the decision was made to utilize the MLP algorithm in this study.

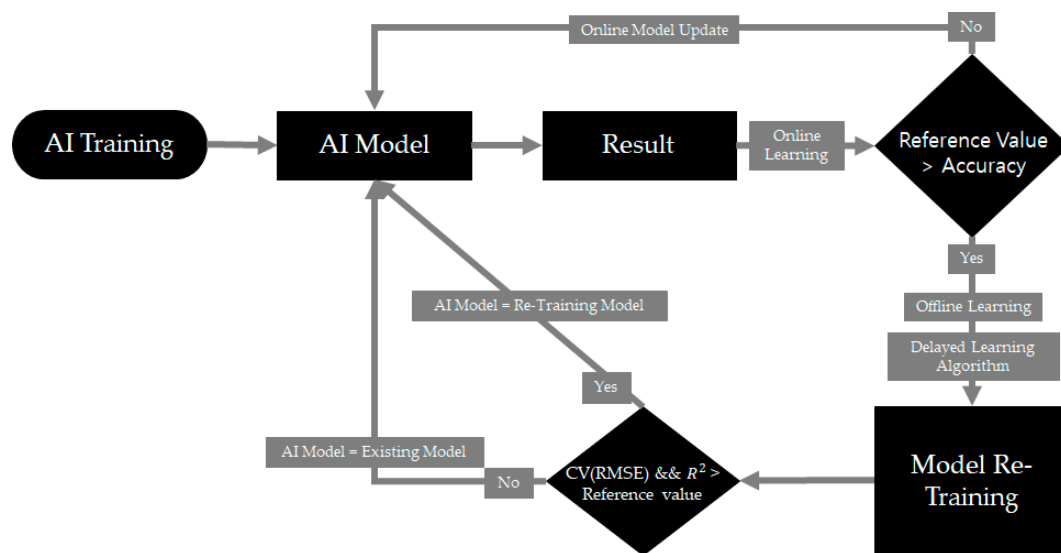
The MLP model was developed and analyzed using Python 3.9.7, Tensorflow 2.3.0, Keras 2.4.3, Sklearn 1.0.2, Pandas 1.4.1, Numpy 1.19.5, and Matplotlib 3.5.1. The dataset was split into training and test data, with a ratio of 90% for training data and 10% for test data. Daily data spanning approximately 3 years were used for the electricity usage prediction model, and daily data covering around 3 months were utilized for the LNG usage prediction model. Data preprocessing involved the application of Minmax Scaling and Standard Scaling methods. Additionally, the previously developed predictive models served as base models before undergoing hyperparameter optimization. All hyperparameters were set to default settings and analyzed. Validation for each MLP-based prediction model was conducted by comparing predicted values with actual measured values over a 1-month period.

### 3.2. Integration-Learning AI Modeling

When employing AI models to predict energy management systems or demand and supply within an industrial complex, various factors such as process or task changes must

be considered. Therefore, the AI model must be sensitive to data fluctuations and ensure the continual checking of the reliability and accuracy of predicted values.

Many AI models currently utilized in systems requiring real-time data reflection often utilize online learning to manage real-time data streams or large datasets. Online learning offers the advantage of promptly reflecting changes as they occur. However, setting up large server systems in factories implementing online learning is challenging owing to spatial, cost, and energy constraints. To address these challenges, a solution has been devised: replacing large server systems with low-performance edge devices, known as integration-learning AI modeling. Integration-learning AI modeling combines online and offline learning. It primarily operates via online learning, which updates the AI model in real time. However, if overfitting or bias occurs due to continuous weight updates and the average accuracy of the AI model drops below a threshold value, offline learning is initiated to retrain the AI model using all collected data up to that point, thereby mitigating bias in the data. The proposed integration-learning AI modeling system is illustrated in Figure 4.



**Figure 4.** Integration-learning AI modeling architecture.

Factory prediction systems must incorporate both online and offline learning because factory production is characterized not only by its time-series nature but also by its strong dependency on produced items. Thus, sensitivity to the most recent data is crucial, necessitating the use of online learning. However, past data can also provide valuable insights, and online learning tends to gradually favor more recent data, leading to bias. Therefore, offline learning should also be employed to counteract this drawback.

Several issues need to be addressed to effectively utilize integration-learning AI modeling, which combines online and offline learning, on edge devices. First, it may be challenging to implement integration-learning AI modeling on edge devices due to their limited computational performance. Additionally, there may be a latency issue where the prediction process extends beyond the completion of the next prediction due to high inference latency. Furthermore, there could be device-related issues during offline learning, potentially hindering model training.

To address these challenges, the model was optimized to be lightweight for online learning. Comparative analysis with an existing model revealed no significant difference in sensor data prediction. Although prediction speed was slower compared to a conventional server system, there were no bottlenecks observed in prediction and online learning for both the conventional server system and edge devices. However, bottlenecks occurred



during data retrieval. Moreover, the CPU-based MLP model demonstrated low load during prediction, online learning, and backpropagation.

In the case of offline learning, there is a possibility of increased model training time as data accumulate. To mitigate this issue, the algorithm was enhanced by incorporating a technique to delay offline learning. This technique schedules training during non-operational time slots, such as idle periods, to prevent disruption to factory operations. Additionally, if a significant amount of data accumulates, potentially leading to high load conditions, offline learning is delayed accordingly.

#### 4. Construction of Data Collection System

To conduct learning and predictions using AI, it is imperative to initially gather data for model training. Consequently, a robust data collection system becomes indispensable. Specifically, for operating a prediction system on an edge device independently, a reliable data collection system is paramount. While a previous evaluation of a data collection system on an edge device was conducted, it did not meet the reliability standards necessary for constructing an AI system and integrated platform [23]. Hence, experiments were conducted in this study to ensure the reliability of the data collection system on a lightweight platform integrating AI and data collection systems.

##### 4.1. Database Selection

The aim of this study was to ascertain whether tasks function on the edge device similarly to the conventional server system. Therefore, it was crucial to examine any issues arising when data collection and predictions are simultaneously executed. Consequently, various databases were compared to assess the time taken to load data and the resources utilized for data loading. Through this evaluation, the reliability of the database on the edge device was appraised, and the most suitable database management system was determined based on the reliability assessment results.

Three databases widely used across various domains were selected for evaluation. MySQL was chosen as a relational database management system, while InfluxDB, a time-series database management system known for its adeptness in handling time-series data, was selected, considering the characteristics of sensor data [24,25]. Finally, MyRocks, a relational database management system utilizing the RocksDB key-value store as its backend storage engine, was included. These three databases underwent comparison for evaluation. To gauge the reliability of the integration platform and AI system, collected data were employed to conduct AI predictions every 10 seconds, and the database load was monitored. Based on the results, the most suitable database management system for constructing an AI system with edge devices was determined.

##### 4.2. Evaluation

###### 4.2.1. Evaluation Setup

In this evaluation, tests were conducted using the ASUS Tinker Board 2 as the edge device [26]. This device consumes a maximum of 30 W of power. The database versions utilized for this evaluation were MySQL v5.7, MyRocks v5.6, and InfluxDB v1.8.

###### 4.2.2. Dataset

The dataset employed for the evaluation comprised data collected at second intervals from a demonstration factory. The test involved bulk inserting five million sensor data points, which consisted of 100 float and integer data points per second.

###### 4.2.3. Evaluation of Data Insertion in Edge Device

Upon bulk inserting approximately 3.7 GB of data, MyRocks completed the task in 3030 s, while InfluxDB took 83,100 s, as depicted in Figure 5. However, MySQL encountered an error and could not function as a database. The elapsed time difference between MyRocks and InfluxDB exceeded 27 times. This indicates that under heavier loads,

the data collection system may pose a bottleneck in the operation of both AI and data collection systems.

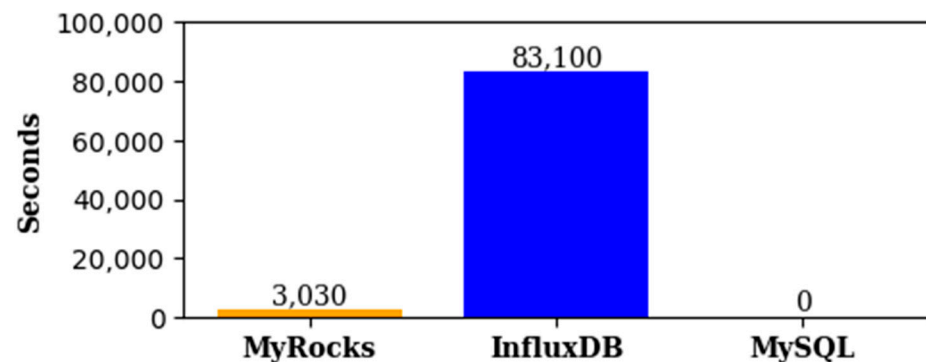


Figure 5. Evaluation of elapsed time for data insertion.

InfluxDB utilizes approximately 33% less memory than MyRocks, and its CPU utilization is approximately 6% lower than that of MyRocks, as illustrated in Figure 6. However, this difference in resource utilization is the reason why InfluxDB took approximately 27 times longer than MyRocks. In this scenario, InfluxDB's lower memory usage and CPU utilization are attributed to an input/output (I/O) bottleneck. Consequently, it is evident that MyRocks utilized resources more efficiently over time.

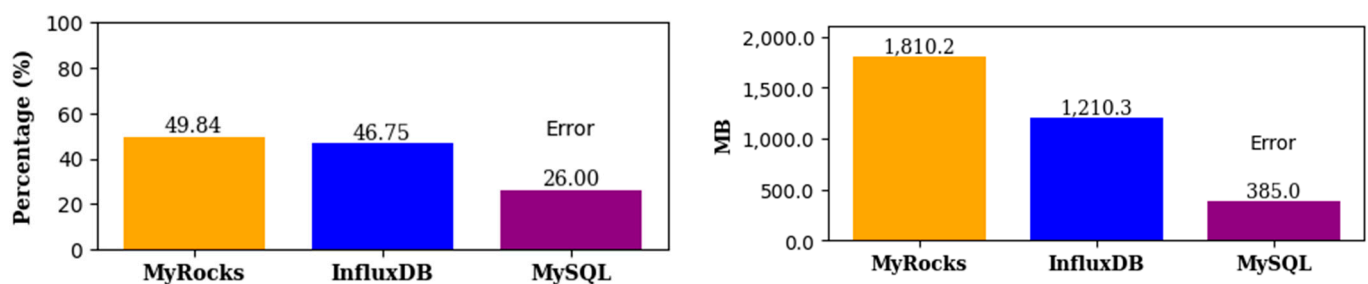


Figure 6. Evaluation of CPU utilization and memory usage for data insertion.

It is apparent that InfluxDB encounters issues with resource utilization due to its high I/O latency, which may compromise data processing reliability under heavier loads. Conversely, MyRocks demonstrated shorter elapsed time and higher resource utilization, indicating superior reliability as a data collection system compared to InfluxDB. Therefore, MyRocks was chosen as the optimal database for edge devices, and subsequent experiments were conducted using it.

## 5. Evaluation and Results

### 5.1. Evaluation Methodology

For this study, the Tinker Board 2, a widely utilized edge device, was employed for the evaluation. The specifications of this edge device are outlined in Table 2.

Table 2. Edge device specifications.

Components	TinkerBoard 2 (ASUS, Taipei, Taiwan)
SoC	Rockchip RK3399 (Rockchip, Fuzhou, China)
Memory	LPDDR4 2 GB
Storage	Samsung mSD 256 GB (Samsung, Yongin-si, Republic of Korea)
OS	Debian 11

MyRocks v5.6 was used as the database for the data collection system. All settings of RocksDB 6.8.0 were set to their defaults.

The experiment data consisted of power data obtained from 10 companies every minute. The first evaluation involved real-time data reception, while simultaneously conducting offline learning of the integration-training AI modeling technique to reconstruct the AI model using 100 days of data. This method aimed to evaluate the elapsed time for offline learning required to reconstruct the AI model. Subsequently, the second evaluation measured the elapsed time for online learning while making predictions based on real-time data collection.

The prediction model was configured to forecast power data for the next unit time interval based on data received every minute. In this study, the unit time for the AI model's predictions was set to 15 min.

Scikit-learn was utilized as the library for AI modeling. For evaluation comparison, a server system with specifications outlined in Table 3, representative of a typical server system used in conventional server systems, served as the comparison target. The evaluation comparison was conducted using this specified server.

**Table 3.** Server system specifications.

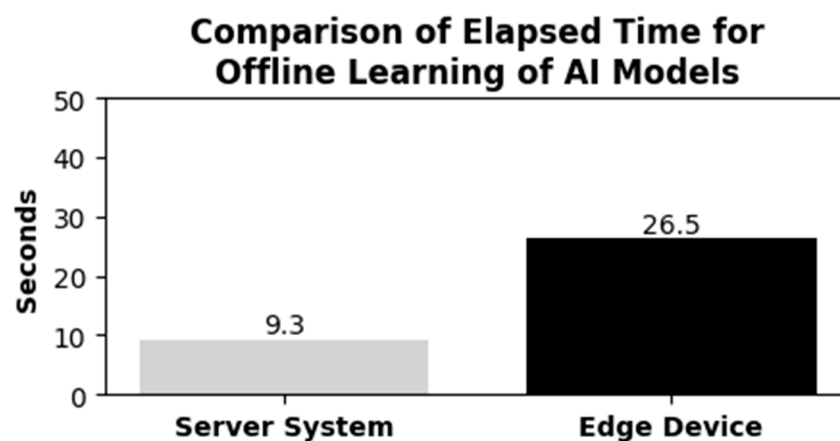
Components	Dell R630 (Dell, Round Rock, TX, USA)
CPU	E5-2620 v3
Memory	128 GB
Storage	Samsung 870 evo 250 GB (Samsung, Republic of Korea)
OS	Ubuntu 22.04

In this study, real-time power data were obtained from 10 factories, and the AI model was trained and utilized for predictions on the edge device. This approach ensured the reliability of the integrated platform for both data collection and AI systems on the edge device, thereby contributing to research on low-power computing.

### 5.2. Integration-Learning AI Modeling Offline Learning Evaluation

This study was conducted with the aim of making comparisons with the existing system.

As illustrated in Figure 7, there was approximately a 2.8 times performance difference in offline learning between the two models. Data were collected at 1-minute intervals in this study. Since predicting and controlling factory energy usage at intervals of seconds is inefficient, predictions were made with a unit time of over 1 minute. Therefore, having a model training time of less than 1 minute is not a concern. Moreover, offline learning was conducted during factory downtime, ensuring it did not directly impact predictions.



**Figure 7.** Comparison of elapsed time for offline learning of the AI models.

Hence, although offline learning while operating the data collection system on edge devices may be slower compared to conventional systems, it was verified that both the data collection system and offline learning operate with high reliability even on edge devices.

### 5.3. Integration-Learning AI Modeling Online Learning Evaluation

In the performance evaluation of online learning, the edge devices were approximately four times slower than the conventional server system, as depicted in Figure 8. However, with a time requirement of only 0.04 s, this discrepancy does not significantly impact predictions.

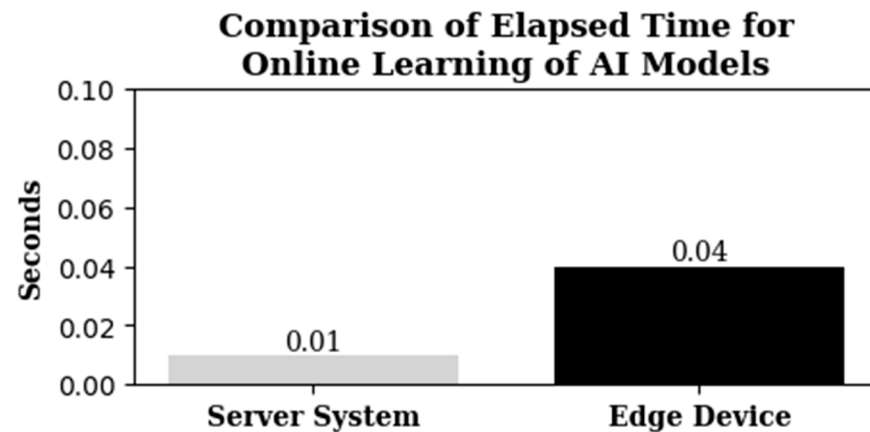


Figure 8. Comparison of elapsed time for online learning of the AI models.

It is worth noting that the performance evaluation results of online learning mentioned above do not include database access time. If the evaluation were to encompass database access time, it would lead to a high load, as illustrated in Figure 9, owing to the complexity of the queries sent to the database. Consequently, the majority of the elapsed time would be attributed to database processing. Thus, the performance difference is less pronounced than when the database access time is excluded. Even when the database access time is factored in, online learning performance registers a short elapsed time of less than 1 minute.

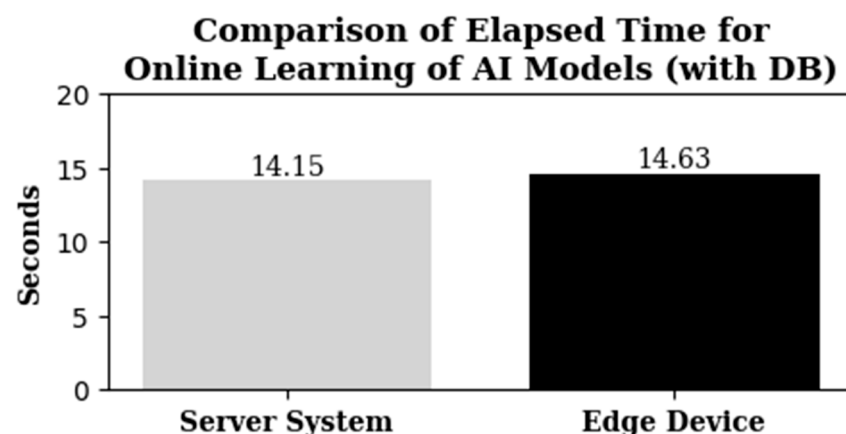


Figure 9. Comparison of elapsed time for online learning of the AI models (with database access).

In summary, utilizing edge devices as a lightweight platform for data collection and AI systems, integrating learning AI modeling suitable for various environments, yields lower performance than the conventional server system. However, offline learning typically takes less than 30 s on average. Furthermore, there is no significant disparity in online learning performance between this AI system and the conventional server system, indicating a high level of reliability in this AI system.

## 6. Conclusions

In this study, we investigated the feasibility of operating data collection and AI systems on edge devices as standalone lightweight platforms in each factory.

The reliability of operation on the standalone edge device was assured through integration-learning AI modeling. Furthermore, even when the edge device was used independently, the load remained low, allowing for improvements to the AI model.

Additionally, although there was a performance difference of 2.8 times in offline learning and up to four times in online learning compared to the conventional system, predictions could be made without any issues on the edge device for a unit time of over 1 minute. In the database evaluation under high load conditions, reliable data collection was achieved when MyRocks was used.

This evaluation demonstrates the potential for replacing conventional server systems, which encounter issues such as high power consumption, inefficient space utilization, and management load. Moreover, it highlights the possibility of ensuring reliability in building edge devices as standalone systems in factories.

This study introduces an integrated lightweight platform by developing both a data collection system and the entire AI system cycle on an edge device. To the best of our knowledge, while there have been numerous endeavors to address energy, environmental, and space constraints, this study represents a significant milestone as the first to overcome these challenges by constructing a comprehensive system entirely on an edge device.

Our investigation devised an AI system capable of learning and predicting data with significant fluctuations on existing low-performance edge devices that are not specialized for AI tasks. This breakthrough addresses the constraints encountered with existing techniques, notably the high computational load, thereby enabling the utilization of low-performance edge devices in handling data with significant fluctuations through the AI system.

This research shows promise for various applications, including smart factories and smart homes, as well as in scenarios where network connectivity and physical locations are constrained due to security concerns or environmental characteristics.

Based on the insights gained from this study, we are currently involved in real-time prediction activities, having deployed our integrated lightweight platform in a demonstration factory. Additionally, we are investigating the enhancement of our capabilities by evolving the data collection system into an IIoT (Industrial Internet of Things) platform to enable control functionalities. Moreover, we have devised plans for a visualization project in response to requests from the demonstration factory.

For future research, our objectives include enhancing the performance of online learning and the edge device AI model by clustering edge devices and enhancing the data collection system. We also intend to explore the feasibility of utilizing edge devices as a lightweight platform for the entire prediction and data collection system cycle.

Moreover, we aim to expand our research beyond AI system development and delve into improving AI algorithms. Our plans entail exploring advanced AI algorithms, refining hyperparameter tuning techniques specifically designed for edge devices, and investigating self-adaptive methodologies to enhance the AI model itself.

**Author Contributions:** Conceptualization, W.C. and J.-h.G.; methodology, W.C.; software, W.C.; writing—original draft preparation, W.C. and H.L.; writing—review and editing, W.C. and H.L.; visualization, W.C. and H.L.; supervision, J.-h.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (20202020900170).

**Data Availability Statement:** The data presented in this paper are available on request from the corresponding author. The data are not publicly available due to the funding institution's research security pledge.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Kim, C.W.; Kim, J.; Kim, S.M.; Hwang, H.T. Technological trends and case studies of factory energy management systems (FEMS) for energy saving in manufacturing industries. *Equip. J.* **2015**, *44*, 22–27.
- Why Industrial Complexes Are Centers of Carbon Neutrality. SK Ecoplant. Available online: <https://news.skecoplant.com/plant-tomorrow/3079/> (accessed on 20 February 2024).
- Korea Energy Agency. Collective Energy Project. Available online: <https://www.energy.or.kr/front/conts/105001003005000.do> (accessed on 20 February 2024).
- Chul, H.C. Energy sharing transaction network establishment of industrial complex and integrated design method. *J. Energy Eng.* **2023**, *32*, 11–22.
- Godse, M.; Mulik, S. An approach for selecting software-as-a-service (SaaS) Product. In Proceedings of the IEEE International Conference on Cloud Computing, Bangalore, India, 21–25 September 2009; pp. 155–158.
- Zeyu, H.; Geming, X.; Zhaohand, W.; Sen, Y. Survey on edge computing security. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 12–14 June 2020; pp. 96–105. [CrossRef]
- Chen, M.; Liu, A.; Xiong, N.N.; Song, H.; Leung, V.C.M. SGPL: An intelligent game-based secure collaborative communication scheme for metaverse over 5G and beyond networks. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 767–782. [CrossRef]
- Merenda, M.; Porcaro, C.; Iero, D. Edge machine learning for AI-enabled IoT devices: A review. *Sensors* **2020**, *20*, 2533. [CrossRef] [PubMed]
- Alajlan, N.N.; Ibrahim, D.M. TinyML: Enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications. *Micromachines* **2022**, *13*, 851. [CrossRef] [PubMed]
- Liu, R.; Xie, M.; Liu, A.; Song, H. Joint optimization risk factor and energy consumption in IoT networks with Tinymml-enabled internet of UAVs. *IEEE Internet Things J.* **2024**, *1*.
- Lapegna, M.; Balzano, W.; Meyer, N.; Romano, D. Clustering algorithms on low-power and high-performance devices for edge computing environments. *Sensors* **2021**, *21*, 5395. [CrossRef] [PubMed]
- He, X.; Kaiyong, Z.; Xiaowen, C. AutoML: A survey of the state-of-the-art. *Knowl. Based Syst.* **2021**, *212*, 106622. [CrossRef]
- Zhang, T.; Zhang, Y.; Katterbauer, K.; Al Shehri, A.; Sun, S.; Hoteit, I. Deep learning-assisted phase equilibrium analysis for producing natural hydrogen. *Int. J. Hydrogen Energy* **2024**, *50*, 473–486. [CrossRef]
- Huang, L.; Chao, Z.; Hongyang, Z. Self-adaptive training: Beyond empirical risk minimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19365–19376.
- Chen, Y.-S.; Cheng, K.-H.; Hsu, C.-S.; Zhang, H.-L. MiniDeep: A standalone ai-edge platform with a deep learning-based mini-pc and ai-qsr system. *Sensors* **2022**, *22*, 5975. [CrossRef] [PubMed]
- MyRocks. Facebook. Available online: <https://github.com/facebook/mysql-5.6> (accessed on 20 February 2024).
- Innopolis. Data Acquisition System Market. Available online: <https://www.innopolis.or.kr/fileDownload?titleId=177527&fileId=1&fileDownType=C&paramMenuId=MENU009992021.04> (accessed on 20 February 2024).
- Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
- Rumelhart, D.E.; Hinton, G.E.; McClelland, J.L. Parallel distributed processing. *Foundations* **1988**, *1*, 45–76.
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
- Runge, J.; Zmeureanu, R. Forecasting energy use in buildings using artificial neural networks: A review. *Energies* **2019**, *12*, 3254. [CrossRef]
- Lee, H.; Kim, D.; Gu, J.-H. Prediction of factory energy consumption using MLP and SVR algorithms. *Energies* **2023**, *16*, 1550. [CrossRef]
- Cho, W.; Lim, C.-Y.; Gu, J.-H. Comparison and evaluation of data collection system database for edge-based lightweight platform. *J. Platf. Technol.* **2023**, *11*, 49–58.
- MySQL. Available online: <https://www.mysql.com/> (accessed on 20 February 2024).
- InfluxDB. Available online: <https://www.influxdata.com/> (accessed on 20 February 2024).
- ASUS TinkerBoard 2. Available online: <https://tinker-board.asus.com/series/tinker-board-2.html> (accessed on 20 February 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.