IoT

MDPI

Article

Industrial IoT-Based Energy Monitoring System: Using Data Processing at Edge

Akseer Ali Mirani [1,2,*], Anshul Awasthi [1,2,*], Niall O'Mahony [1,2] and Joseph Walsh [1,2]

[1] IMaR Research Centre, Munster Technological University, V92 CX88 Tralee, Ireland
[2] CONFIRM Research Centre, Unit 2 Park Point, Dublin Road, Castletroy, V94 C928 Limerick, Ireland
* Correspondence: akseer.ali.mirani@research.ittralee.ie (A.A.M.); anshul.awasthi@mtu.ie (A.A.)

Abstract: Edge-assisted IoT technologies combined with conventional industrial processes help evolve diverse applications under the Industrial IoT (IIoT) and Industry 4.0 era by bringing cloud computing technologies near the hardware. The resulting innovations offer intelligent management of the industrial ecosystems, focusing on increasing productivity and reducing running costs by processing massive data locally. In this research, we design, develop, and implement an IIoT and edge-based system to monitor the energy consumption of a factory floor's stationary and mobile assets using wireless and wired energy meters. Once the edge receives the meter's data, it stores the information in the database server, followed by the data processing method to find nine additional analytical parameters. The edge also provides a master user interface (UI) for comparative analysis and individual UI for in-depth energy usage insights, followed by activity and inactivity alarms and daily reporting features via email. Moreover, the edge uses a data-filtering technique to send a single wireless meter's data to the cloud for remote energy and alarm monitoring per project scope. Based on the evaluation, the edge server efficiently processes the data with an average CPU utilization of up to 5.58% while avoiding measurement errors due to random power failures throughout the day.

Keywords: energy monitoring; industrial internet of things; data processing; data visualization; edge; Industry 4.0

check for updates

Citation: Mirani, A.A.; Awasthi, A.; O'Mahony, N.; Walsh, J. Industrial IoT-Based Energy Monitoring System: Using Data Processing at Edge. *IoT* **2024**, *5*, 608–633. https://doi.org/10.3390/iot5040027

Academic Editors: Antonio Cano-Ortega and Francisco Sánchez-Sutil

Received: 22 July 2024
Revised: 20 September 2024
Accepted: 26 September 2024
Published: 28 September 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

Electricity is the carrier of electric charge, also known as a secondary energy source, as it is converted from primary sources such as coal, fossil fuels, solar, and wind energy [1]. Over the last few centuries, electrical energy has transformed the world ecosystem, from replacing the use of candles and oil lamps to unprecedented use cases in practical life, laying the foundation for a new digital world [2] and acting as a crucial causal effect on the development work, supply quality, infrastructure, and growth of our economy [3]. According to the US Energy Information Administration's report in 2023, global electricity generation is expected to rise by between 30% and 76% by 2050 compared to 2022, while industrial energy consumption alone is forecasted to increase from 9% to 62% [4]. Moreover, in another report, worldwide electricity demand is projected to grow from 62% to 185% by 2050 compared to 2021 [5]. Based on the electricity consumption forecast from both reports, there is a need to take immediate steps to avoid any electricity shortage issues by developing intelligent energy monitoring systems to help moderate and optimize electricity consumption, especially in industrial applications.

Conversely, industrial process monitoring is undergoing significant technological advancements due to the integration of Information Technology (IT) with Operational Technology (OT) under Industry 4.0 and Industrial Internet of Things (IIoT) paradigm shifts [6]. This evolution has created significant opportunities for monitoring and managing energy from distribution to its usage in various applications, such as intelligent transportation, smart grid, green IoT, and smart buildings [7]. Meanwhile, as the rapid implementation of IIoT technology enables the robust monitoring of industrial field devices via the cloud [8], its fusion with edge computing further paves the way for developing scalable and intelligent systems by bringing cloud computing capabilities near the hardware, resulting in improved latency [9]. The edge-assisted intelligent systems facilitate the collection and storage of ample data from field devices and support process monitoring with high customizations and reliability [10]. Moreover, the edge platforms also support processing time-series data using the rule engine to aid the user in decision-making based on data, efficient utilization of resources, and reduced operational costs [11].

This research presents an edge-assisted IIoT-based system architecture for monitoring the energy consumption of stationary and mobile industrial processes on a manufacturing industry's factory floor with the help of a data processing feature to compute additional energy consumption metrics. The proposed solution uses industrial three-phase wired and wireless meters to monitor the energy usage of various industrial processes and send this information to the edge machine, which stores the data, calculates new energy usage parameters using a JavaScript-based data processing technique in the edge rule engine, provides a local user interface, and then filters and sends the data to the cloud for remote monitoring of a single wireless meter as per project scope. The proposed solution presents an approach to monitoring the individual analytics from each energy meter and collective energy consumption insights of all the energy meters to facilitate the end-user in monitoring the overall system based on the energy parameters calculated using the data processing method. Moreover, it also provides the activity and inactivity alarm events when an energy meter is not accessible for a specific time and sends the dashboard reports to the user via email. The following are the main contributions of this research:

- Development of IIoT system architecture for local and remote energy monitoring of stationary and mobile industrial processes on a factory floor.
- Implementation of a JavaScript-based data processing technique at the edge instance to calculate new energy consumption parameters from the collected energy meter data.
- Development of master and individual energy meter dashboards for comparative and individual energy usage information.
- Implementation of device activity/inactivity alarms and email-based reporting feature for smart energy monitoring.

The remaining article is organized as follows: Section 2 presents the literature review of related articles from the research and some leading industrial energy monitoring solutions. In Section 3, we discuss in detail this project's system architecture, including the hardware and software development of components, followed by the results and discussion from our research in Section 4. Finally, in Section 5, we conclude our research by presenting the possible directions for future work.

2. Literature Review

This section presents a detailed review of the proposed IoT-based solutions from the research side and some leading industrial systems, including their features for monitoring energy consumption.

2.1. Related Articles

The IoT-based monitoring solutions for any application in the literature are based on a multilayer architecture comprising sensors layer to cloud layer [12]. In this research, as the main objective is to monitor energy consumption, to achieve this, we conducted the

literature review by obtaining the related articles by searching the keywords ("IoT" or "IIoT" or "data processing in energy meters" and "energy monitoring") from different research databases from 2019 to 2024. Based on our review, the proposed solutions in the literature are based on three main categories: simulation, prototype, and implementation. Of the 32 articles, 2 present the solution based on simulation [13,14], 19 on prototypes [15–33], and 12 on implementation [34–45]. This section discusses each category of related articles in detail and then compares their features and limitations.

2.1.1. Simulation

Conventional energy monitoring systems require a manual meter reading process to record the electrical energy units consumed in household applications. In this regard, the authors in [13,14] propose intelligent energy monitoring systems which track energy usage using an Arduino controller. The solutions used a simulation model using Proteus software, where the controller monitored the current and voltage signals from the virtual sensors and forwarded this information to the Blynk IoT platform to monitor the system remotely. In addition to monitoring, the authors in [13] include a load tempering approach to detect electricity theft, while the researchers in [14] further provide an enhanced idea to detect electricity theft, control the output load, and send notifications based on threshold level using Node-RED graphical programming.

2.1.2. Prototype

A low-cost energy monitoring system enables monitoring of the energy consumption of various applications on a large scale. In this regard, the authors in [24,27] present prototype-based solutions using breadboards to monitor energy consumption with the help of microcontrollers and current-voltage sensors. Conversely, a few similar solutions are proposed by the researchers in [16–22] to monitor the energy consumption of solar panels using different prototypes with the help of microcontrollers to identify defective solar panels and wiring and ensure the optimal electricity output. However, the proposed solutions need to evaluate the accuracy of their energy monitoring systems.

The monitoring of energy consumption for geographically distributed assets takes a significant process. Concerning this, the researchers in [29] have presented a LoRa-based long-range wireless energy monitoring system using Arduino controllers and current–voltage sensors for the energy metering nodes and Raspberry for the primary node, which collects the energy consumption data and forwards this information to the cloud. Nonetheless, the authors of this solution need to provide the monitoring accuracy of the proposed solution. A similar solution is found in [30], where the authors performed experiments to calculate the accuracy of their LoRa-based energy monitoring system, which was found to vary from 1.54% to 5.04%. Nevertheless, the hardware used in these solutions is not suitable for industrial applications.

While the IoT-based insights on energy consumption pave the way towards the efficient utilization of resources, the intelligent controlling of load can further reduce unnecessary costs. In this regard, the authors in [31] propose a system to control the output load of electrical applications based on the consumer configuration. In a more advanced approach, the authors in [32] propose a solution to control the output load based on detecting over- and under-voltage fluctuations towards the preventive measures to protect the equipment. Moreover, the researchers in [26] present a solution to control the load by detecting power theft based on the total power flowing through the control panel, electricity service drop, and energy meter. The proposed solution in this research is a breadboard prototype that lacks the reliability to sustain the harsh industrial environment.

Aside from monitoring the energy consumption and controlling the output load with good efficiency, another key feature is to generate threshold-based alarms as provided by the authors in [23,28], which can be critical to managing the payment of energy usage and controlling the meter output if the bill is not paid, as proposed in [25]; however, the identified proposed systems are prototypes and are not suitable for industrial applications.

The researchers propose another non-industrial solution in [15] to monitor the energy consumption of residential houses using a low-cost PZEM004T energy monitoring circuit and ESP8266 controller to monitor the energy consumption and along with digital payment of electricity bill based on the RFID-based tag identification approach.

2.1.3. Implementation

The exponential growth in electricity consumption for consumers and industrial applications is creating challenges to provide continuous distribution of electrical energy. In this regard, innovative solutions, such as the hybrid renewable power plant, which provides clean energy, are being developed. The authors in [36] have introduced a customized IoT-based solution featuring an ESP32 controller and a low-cost energy circuit to monitor the electrical parameters of the power plant. Similarly, in [41,42], the authors propose a low-cost customized solution using a microcontroller and current–voltage sensors to monitor and conserve energy usage in distributed Zigbee-based wireless communication applications. However, it is important to note that the hardware of these proposed solutions requires further evaluation for operation in an industrial environment, highlighting potential areas for future research and development.

To address the ruggedness of hardware in industrial applications, the authors in [43,44] propose IoT-based architectures to monitor energy consumption. While the authors in [43] present an FPGA-embedded controller for monitoring power substations using a local monitoring approach along with the support of a remote connection instead of a scalable cloud platform, the researchers in [44] implement an industrial energy meter for real-time visualization of parameters available from the industrial meters. However, these solutions have not considered the crucial aspect of data processing at the edge to discover new energy parameters.

On the other hand, while the researchers in [40] propose a solution that provides daily and hourly energy consumption to manage the Demand Side Management (DSM) of consumer applications towards the Smart Grid application, ref. [35] implements the alarms feature to notify the users in case of abnormalities detected in the system for environmental sustainability. It underscores the importance of monitoring and controlling energy consumption in consumer applications and the role of IoT technology in achieving this. Conversely, Munoz et al. in [45] present a customized solution to develop an intelligent IoT-based energy monitoring system that provides various energy quality parameters with high accuracy using an ADE9000 electricity measurement circuit and ESP32 controller; however, this solution does not utilize an edge server to offer big data support along with intuitive visualization of data.

Due to the increasing global energy demands, current research practices address conservative energy usage methods using various scientific innovations. In this regard, the researchers in [39] propose a Smart Compact Energy Meters (SCEMs)-based system for IoT-based monitoring the energy patterns and controlling its usage to improve the power quality issues of a commercial building using a low-cost ESP8266 controller, which collects the energy information from the sensors and forward this information to the Blynk IoT cloud platform. Moreover, this research also implements the control of secondary load to manage power consumption during peak times. A few similar related works are available in [33,34,37,38] to monitor the trend in energy consumption, control the output load towards conservative energy usage during the peak time throughout the day, and control the output load based on the utility bill payment from the customer.

2.1.4. Comparison of Related Articles

The proposed solutions in the literature are innovative, utilizing different layered architectures. They primarily differ based on various critical parameters such as communication interface, type of hardware, edge and cloud user interface, and hardware reliability. Furthermore, the reviewed papers are also, in contrast, based on the accuracy of monitoring using industrial energy meters or custom-developed hardware using low-cost sensors.

A customized solution is a set of sensors and controllers where the controller continuously monitors the current and voltage signals of the sensors, calculates the energy consumption, and then transmits the processed data to the cloud. Moreover, an industrial solution uses an energy meter to measure different parameters accurately while withstanding the harsh industrial environment. The industrial energy meter also supports digital communication protocols from which the monitoring system collects data, processes them, and then sends them to the cloud platform.

Based on the review, the simulation-based papers [13,14] are found to be based on low-cost hardware components, while in prototype-based solutions [15–33], the researchers have developed temporary testbeds using customized solutions comprising microcontrollers, current–voltage sensors, and various IoT platforms for the remote visualization of data. However, most of the customized solutions in the literature primarily lack two parameters: the efficiency of the monitoring solution and the rugged hardware to work in the harsh industrial environment. On the other hand, implementation-based solutions [34–45] are found to use two types of energy metering systems: customized hardware and industrial meters. While customized hardware is low-cost, industrial meters can efficiently perform in an industrial environment.

The energy monitoring solutions available in the literature have various uses depending on the applications; however, based on the review, there is still a gap in utilizing edge platforms' computational abilities to develop smart energy consumption monitoring solutions. Although the researchers in [33,43,44] provide a local graphical interface to monitor the energy data, the reviewed articles in the literature need the ability to utilize the edge-based data processing feature to calculate new energy consumption parameters and scalably monitor the energy parameters locally. Table 1 compares the related work in the literature and their key limitations for IoT/IIoT-based energy monitoring systems.

Table 1. Comparison of IoT-based energy monitoring solutions in the literature.

Ref.	Edge GUI	Cloud GUI	Simulation/ Prototype Implementation	Wired/ Wireless	Customized Hardware/ Industrial Meter	Key Limitations
[13,14]			Simulation	Wireless	Customized hardware	This is only conceptual model
[15–32]		✓	Prototype	Wireless	Customized hardware	Hardware is not suitable for industrial applications
[33]	✓	✓	Prototype	Wireless	Customized hardware	Need evaluation of energy monitoring accuracy
[34]		✓	Implementation	Wireless	Industrial meter	Not suitable for local monitoring and processing of energy data
[35]		✓	Implementation	Wired	Industrial meter	Limited processing power of controller to calculate new energy parameters
[36–38,45]		✓	Implementation	Wireless	Customized hardware	Limited processing power of controller to calculate new energy parameters
[39–41]		✓	Implementation	Wireless	Customized hardware	Need evaluation of energy monitoring accuracy
[42]		✓	Implementation	Wireless	Customized hardware	Hardware is not suitable for industrial applications
[43]	✓		Implementation	Wired	Customized hardware	The system can further improve by adding data processing and cloud-based monitoring
[44]	✓	✓	Implementation	Wired	Industrial meter	Limited processing power for scalability and to calculate new energy parameters

2.2. Industrial Solutions

An energy management system is the process of collecting the measured energy consumption data and evaluating their usage to optimize electrical energy consumption. In this context, the industrial products [46,47] provide a cloud-based customized energy monitoring solution based on the customers' needs; however, these solutions do not specify the range of features energy monitoring features offered at the user interface.

In [48–50], the industrial solutions provide analytics platforms to visualize the geographically dispersed industrial assets and monitor their energy consumption using cloud platforms and take suitable actions to optimize the energy consumption based on the trend in data; nonetheless, these solutions do not provide a software customization framework for the developers to process the collected data using rule engine and design the system as required.

In [51], the industrial solution stands out with its comprehensive cloud-based energy monitoring system. It offers a wide range of customized reports and a user interface that allows users to select different parameters using mathematical formulas. This solution provides in-depth visual insights from the energy meter data through time-series graphs, heatmaps, and tables, making it a valuable tool for energy management. It also supports the integration of third-party energy meters; however, it does not support the feature to process the raw energy meter data, and users rely on the available range of functions.

The industrial solution in [52] offers a cloud-based data processing feature to visualize the energy parameters for comparative energy usage of meters and a daily energy reporting feature. However, it is important to note that this solution has a significant limitation as it only supports the integration of energy meters manufactured by the same organization. It could be a crucial factor for organizations to manage the interoperability between the diverse energy meters and other field devices, making it challenging to develop a unified system.

To avoid the complete reliance of industrial applications on closed energy monitoring systems, the industrial solution in [53] presented by ABB company provides an interoperable energy data visualization software that provides the cost for the hourly, daily, weekly, and yearly historical data. The system also compares current energy consumption data against the past. Moreover, it also allows for the setting up of targets for the efficiency of energy usage. However, this solution is suitable for cloud-based monitoring where the users only require an easy-to-use interface to process the energy consumption data and provide comparative insights.

To overcome the limitations of local data processing for energy meters, the industrial solution in [54] provides an energy suite comprising two software: Eco Explorer Go, which provides the visualization of collected energy consumption information, and the ResMa energy management software, which also provides the integration support for different field devices with the system. This solution supports fewer customization features to design the user interface for the comparative energy insights using the rule engine.

The industrial energy monitoring solutions in the literature mainly provide the graphical visualization of collected energy meter data, alarms based on threshold events, system reports consisting of dashboards, and features to provide comparative insights. Since the reviewed energy monitoring solutions are developed for the dedicated purpose of monitoring energy consumption, the primary limits for these systems are standalone operation and the extensive support of features. For example, Industrial IoT platforms such as ThingsBoard [55], AWS IoT [56], and Azure IoT [57] provide a vast range of software components, including the edge server and the processing of data using a rule engine. Due to the limitations of identified features, industrial energy monitoring software solutions have limitations related to interoperability and customization. Table 2 compares some of the leading energy monitoring solutions from different industries.

Table 2. Comparison of various industrial energy monitoring solutions.

Ref.	Industrial Solution	Features	Limitations
[53]	ABB Ability™ Energy Management System	• Historical timeseries data. • Hourly, Daily, Weekly, and Yearly consumption record. • Benchmark of energy present energy consumption data against the historical data. • Customized energy monitoring report.	• The cloud offers storage and data processing features with less emphasis on the edge. • Predefined functions and less flexibility for customizations. • High cost due to cloud-oriented solution.
[46]	BONNER Monitoring Systems	• Fully customized energy consumption solutions provided by the industry.	• Less opportunities for modifications due to a closed system. • Less documentation of features offered.
[49]	DIAEnergie Industrial EMS	• Energy visualization. • Data analysis. • Alarms and reports. • Different levels of user authorities.	• There are fewer opportunities for modifications due to a closed system. • Less documentation of features offered.
[52]	Eniscope Analytics	• Provides energy consumption insights, different variables and cost of usage. • Daily energy monitoring report. • Provides energy insights, such as per-minute consumption, daily consumption, and average consumption.	• It is a cloud-based solution dedicated to monitoring the energy consumption of company-owned meters. • Not compatible with local monitoring. • No big data support.
[51]	EnergyCAP SmartAnalytics	• Customized energy reports. • Integration support for different energy metering systems. • ROI calculations.	• Limited to cloud-based support. • Not compatible for local monitoring. • Costly for large historical data.
[50]	FourJaw Energy Monitoring Software	• Easy to integrate energy meters. • Detect the operational times of machines.	• Lack customization features • Only supports cloud-based UI.
[48]	Optii Data Analytics Platform	• Monitoring geographically dispersed industrial assets. • Historical timeseries insights from data. • Customized user interface as per customers' need.	• Prime focus on visualization and less emphasis on data processing. • High cost due to cloud-oriented solution.
[47]	SIMATIC Energy Manager	• Fully customized energy consumption solutions provided by the industry.	• Less opportunities for modifications due to a closed system. • Less documentation of features offered.
[54]	Weidmüller Energy Suite	• Local energy monitoring suite. • Integration support for different energy metering systems. • Support of local data processing feature.	• Less customization features to design the user interface. • Support cloud-based deployment in a virtual environment, but it is costlier.

3. System Architecture

This research presents an IIoT-based energy monitoring architecture part of a SCADA-based system to monitor and control various industrial applications on the factory floor. The energy monitoring strand that is also the focus of this research comprises three architectural layers: an energy metering system, an edge gateway and a cloud layer. At the device layer, the architecture consists of wireless and wired energy metering systems installed to monitor various industrial assets on different factory floor parts. These metering systems monitor different energy consumption parameters and use wireless and wired communication methods to integrate devices with the local edge server and forward the data in JSON format, including the variables data and timestamp. Once the edge receives data, it preprocesses this payload data to save information in the PostgreSQL database, finds additional parameters, provides a local user interface, and transfers the filtered data to the cloud for remote monitoring. Figure 1 shows the proposed system architecture for an IIoT-based energy monitoring solution for industrial machinery on an industrial factory floor.

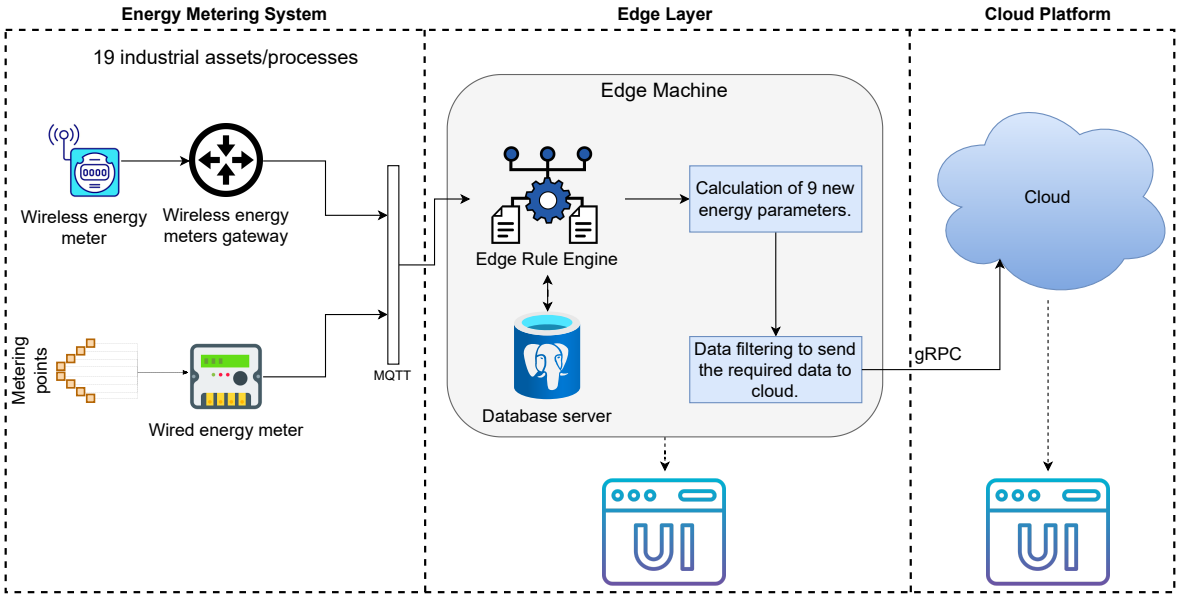


Figure 1. Proposed architecture for Industrial IoT-based energy monitoring system.

3.1. Energy Metering Systems

The research's scope for monitoring three-phase energy consumption contains two main categories of assets: movable and stationary. The movable type contains one mobile welding machine, while the stationary category comprises several machines or industrial units. Due to the nature of machine mobility, the proposed system contains an industrial wireless energy metering system consisting of wireless and gateway nodes. At the same time, the fixed assets or processes are monitored using wired energy meters.

3.1.1. Wired Energy Meter

An industrial wired energy meter monitors the energy consumption of assets within the industrial ecosystem, and it provides analytical information using wired communication protocol to another processing device or a platform to monitor the system efficiently [58]. In this research, we monitor the three-phase energy consumption of eighteen machines and processes using three Eniscope Hybrid energy meters [59]. Each Eniscope Hybrid meter contains eight channels, each combining three current transformers (CT), three voltage terminals, and a neutral terminal to monitor the three-phase energy consumption; however, we can also use each input channel to monitor the energy consumption of three different single-phase machines or processes. These energy meters provide a diverse range of energy consumption parameters, including but not limited to voltage (V), current (A), active power (W), frequency (Hz), and total energy (kWh). This device is suitable for harsh industrial environments and provides an accuracy of $\pm 1\%$ per the IEC 60044-1 standard [60].

The wired energy meters used in research are installed in the industrial factory floor's main power control room, which manages all the distributions for electrical connections. The electrical room contains main power management cabinets comprising industrial-wired energy meters, breakers and panel LCDs for local monitoring. In terms of installation setup, each eight-channel wired meter uses a three-phase and four-line (3LN) connection method [61] for monitoring the energy consumption of machines. Figure 2 shows the schematic diagram of the connection of one of the metering points of the eight-channel wired energy meters.

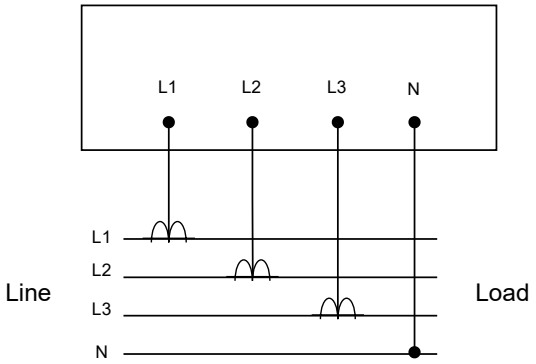
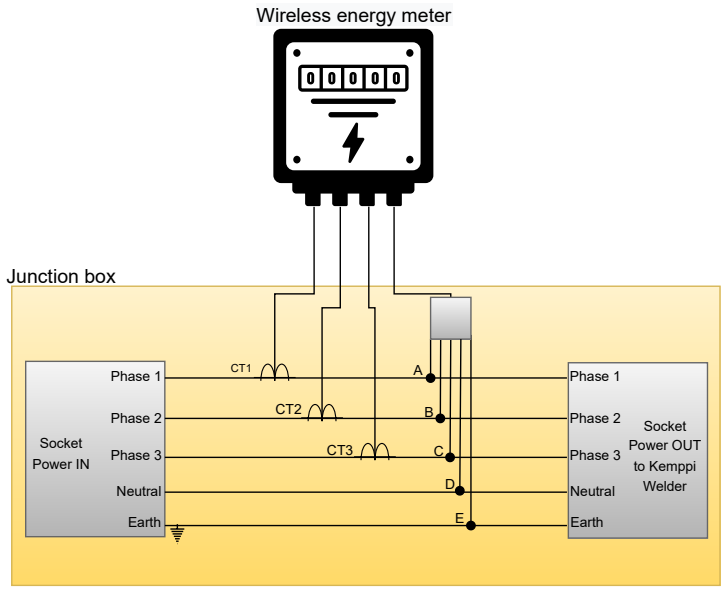


Figure 2. Schematic diagram of electrical connections for a single channel of wired energy meter.

3.1.2. Wireless Energy Meter

A wireless energy meter is a device that uses the wireless medium to transfer information containing energy consumption parameters either to a gateway controller or directly to the cloud [58]. In this work, we are monitoring the energy consumption of a single mobile welding machine on the factory floor using an Episensor wireless metering system consisting of a wireless node (ZEM-63) [62] for the welding machine and a gateway (NGR-30-5) [63], which collects the data using Zigbee-based wireless communication and transfers it to the local edge server. The Episensor wireless meter, similar to the Eniscope Hybrid meter, provides three-phase energy consumption with various parameters at the accuracy of class 0.5s according to IEC 62053-22 standard [64]. Moreover, the energy gateway is designed to operate in the industrial environment, support the wireless transmission range of 100 m, and add up to 100 wireless meters using a mesh network.

On the other hand, the wireless energy meter is installed at the welding machine using a junction box that ensures a reliable power connection between the electrical socket at the wall and the welding machine's input power connection. The electrical connections are also made using a 9S/16S four-wire Wye configuration, following the installation guidelines of the industrial meter, where each CT and voltage wire is connected to the appropriate input power phases with an additional ground wire. Figure 3 shows the schematic diagram of the junction box for installing a wireless energy meter.



- CT1, CT2, & CT3 is connected at Live wires to measure current.
- Wires A, B, & C is connected at Live wires to measure voltage
- Wire D is connected to Neutral wire
- Wire E is connected to earthing connection

Figure 3. Schematic diagram and junction box for wireless energy meter.

3.2. Edge Layer

An IoT edge platform brings the cloud computing services closer to the monitoring devices to process the data with low latency and visualize the information with the support of a big database server [65]. According to Chougule et al. [66], executing algorithms directly on the edge devices without relying on the cloud services improves the system efficiency in real time decision-making based on the vast amounts of data with low latency. Moreover, using an edge instance addresses security and privacy concerns along with an offloading feature to adjust the operational load between the local devices and the cloud [67].

3.2.1. Edge Setup

In this research, we use a ThingsBoard IoT edge software platform in a docker containerized environment [68], deployed on a local Dell Optiplex 5060 machine [69] in the project's first phase to develop project architecture with the plan to use the more rugged system in the next phase. The edging machine uses the Ubuntu operating system (OS), having a Core i5-8600 processor and 16 GB RAM. Furthermore, to provide extensive storage service, the edge machine contains two drives, one with 512 GB SSD to run the OS and the second with 1 TB SSHD dedicated to the PostgreSQL database server for storing and retrieving device data. The secondary storage was added to the system to configure it to keep historical data for up to five years, which is compatible with the number of devices used in our architecture.

3.2.2. Implementation Overview

Following the installation of the edge server and integrating energy meters with the edge using the MQTT protocol, the meters forward the energy consumption data at a sampling rate of one minute. Once the data reach the edge, they are stored on the database server after identifying the source device. Then, the edge instance processes the energy meter data received at the edge instance to find analytical parameters from the variable kWh, which always show the absolute incremental energy consumption value. Although the energy meters also provide other parameters, including but not limited to current and voltage, we focus on kWh as it is relevant to our user interface requirements and crucial for the overall comparative analysis of energy consumption from different industrial machines.

On the data processing side, we implement the JavaScript-based approach at the edge rule engine to find nine new analytical parameters. Based on our review, the industrial product [52] provides a daily energy reporting service consisting of five similar parameters we find through data processing. Notably, our research focuses on measuring these existing variables and four additional parameters using a data processing method at the edge with an interactive and intuitive monitoring interface for monitoring information using edge and cloud platforms. Table 3 provides an overview of parameters calculated using this project's edge data processing method, including their variable name used in UI and the sampling rate using the edge rule engine.

Table 3. Parameter calculation using data processing method at the edge.

Abbreviation	Variable Used in UI	Energy Parameter	Sampling Rate	E/A
$^{\dagger}E_{\text{min}}$	per_minute	Per minute energy consumption	1 min	E
$^{*}E_{\text{today}}$	Today (kWh)	Today consumption	1 min	E
$^{\dagger}E_{\text{24h}}$	Last 24 h (kWh)	Last 24 h energy consumption	1 min	E
$^{*}E_{\text{weekly avg}}$	Average Day (kWh)	Weekly average energy consumption	10 min	E
$^{*}E_{\%}$	Usage (%)	Percentage of energy consumption increased or decreased as compared to weekly average energy consumption	10 min	E

Table 3. *Cont.*

Abbreviation	Variable Used in UI	Energy Parameter	Sampling Rate	E/A
$^{*}E_{\text{total-weekly avg}}$	Last week's average	Total weekly average energy consumption for last week	1 h	A
$^{*}E_{\text{total-today}}$	Today Energy	Total energy consumption for today in real time	10 min	A
$^{\dagger}E_{\text{total-rt-weekly avg}}$	Average Last Week (kWh)	Total weekly average energy consumption with reference to the same timestamp in database	10 min	A
$^{\dagger}E_{\text{total}}$	Total Energy	Total energy usage since the installation of devices	10 min	A

* Indicates the similar parameters found in [52]. † Indicates the new analytical parameters presented in our research. E indicates the calculation of the parameter for each energy meter while A indicates the parameter for overall energy usage.

In the next step, the edge provides the local user interface, and the required analytics are filtered and forwarded to the cloud service for remote system monitoring. In this research, we design an edge user interface that consists of a master dashboard for all the devices and an individual dashboard state for each energy meter to provide in-depth insights for the end-user. At the same time, the cloud monitors a single energy meter as per the scope of work. The edge is also responsible for raising the activity and inactivity alarms when the energy meters do not send the data to the edge. Figure 4 shows the software features implemented at the edge of this project.

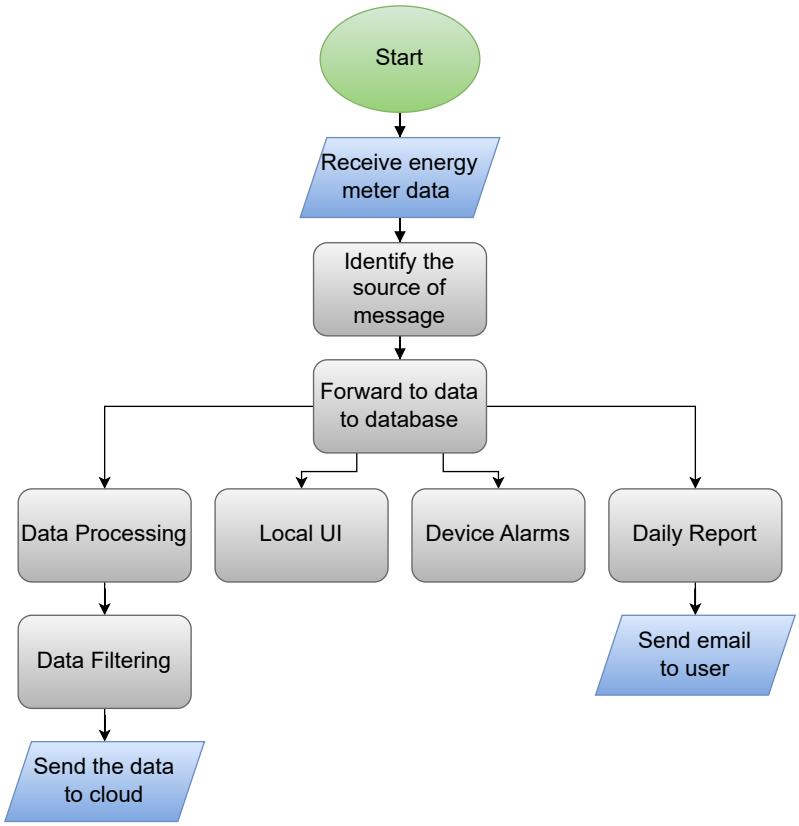


Figure 4. Overview of software implementation at edge.

3.2.3. Software Integration of Energy Meters with Edge

The three-phase industrial wired and wireless energy metering systems installed at the factory floor are integrated into the edge server by utilizing the local area network (LAN) based on network switches.

1. Wired meters: Each wired monitoring device used in this strand consists of 8 channels, where each channel has a set of current transform (CT) and voltage measuring terminals for three-phase energy monitoring. Furthermore, each channel is configurable to independently send the monitoring data to the edge server using MQTT client and broker-based communication [70]. Figure 5 shows the integration method of connecting the wired meter with an edge instance.

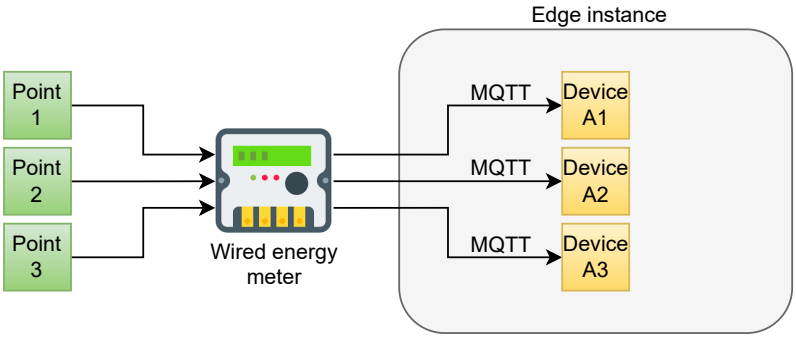


Figure 5. Overview of wired energy meter integration with an edge instance.

2. Wireless gateway: In this strand, once the wireless meter node successfully sends the measurements to the gateway via Zigbee protocol, the gateway forwards this information to the edge instance using a single MQTT client connection with an edge. Regardless of the single wireless meter node used in this research, the system is designed to identify the nodes based on the source address. The system architecture can also scalably add more wireless nodes, forming a wireless mesh network for future energy monitoring of mobile welding machines. Figure 6 shows the integration method of a wireless energy metering system with an edge instance.

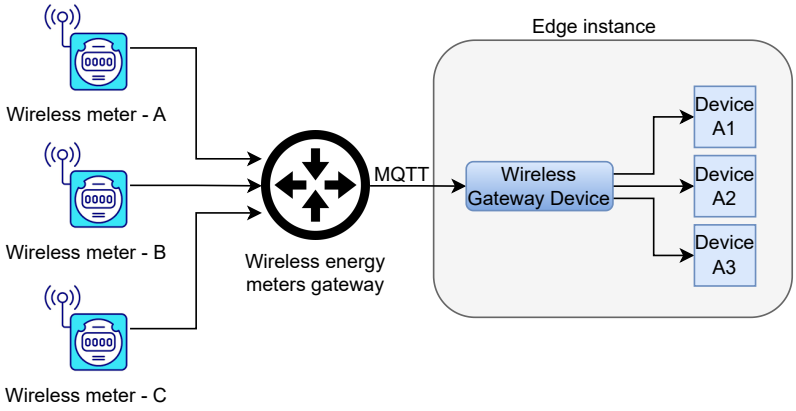


Figure 6. Overview of wireless energy meter integration with an edge instance.

3.2.4. Data Processing

Data processing is the systematic procedure of combining various data collection and preparation frameworks that transform raw data into useful information in numerical, textual, and graphical formats [71]. In this research, data processing is an essential method implemented at the edge instance of the system architecture to process the energy meter data. A mathematical approach is used to find additional energy parameters based on user feedback for data visualization at the edge and cloud. After the energy meter data are received at the edge, the rule engine identifies the source of telemetry data, forwards them to the respective virtual devices created at the edge instance, and stores them in the database. The data received contain different variables, including the energy consumption value in kWh obtained from both three-phase energy meter types in ascendant value format, which means that with time, the value of the kWh parameter always increases. In the

next step, the rule engine feature of the edge calculates energy consumption parameters as identified in Table 3. This section discusses the mathematical approach to finding the energy parameters, including the interval-based data fetching approach to minimize measurement errors due to power or network connectivity events.

1. E_{min}: It is an essential parameter for monitoring machines as it shows the energy consumed by the meter every 1 min. The edge instance finds the E_{min} value by fetching the kWh value of a digital meter from the database within the last 1–2 min, denoted by $last_{\text{e}}$, and then subtracting this new value from the latest kWh received at the edge using Equation (1).

$$E_{\text{min}} = kWh - last_{\text{e}} \tag{1}$$

   where kWh is the variable for an energy meter's latest total energy usage and $last_{\text{e}}$ is the kWh value within the previous 2 min timeframe.
2. E_{today}: The daily energy consumption (E_{today}) is calculated by fetching a meter's total energy consumption till midnight from the database and subtracting it from the latest kWh value as shown in Equation (2). Suppose the last midnight value is unavailable in the database due to power failures; the system fetches the last available value using the 12 h interval-based approach with a timestamp from the previous day to the last midnight. This approach helps to avoid measurement errors due to power or connectivity issues.

$$E_{\text{today}} = kWh - E_{\text{last-midnight}} \tag{2}$$

   where $E_{\text{last-midnight}}$ is the total energy consumption of an energy meter till the last midnight and kWh is the latest total energy usage value of that energy meter.
3. $E_{24\text{h}}$: The parameter $E_{24\text{h}}$ provides information on energy consumption from a three-phase metering point over the last 24 h. It is calculated by fetching the time-series kWh value from the database 24 h ago and subtracting it from the latest energy consumption data. In the event the power consumption value is unavailable for the last 24 h timestamp in the database as compared to the latest timestamp, the edge rule engine fetches the last available value within 24 h interval-based approach similar to 12 h interval used for variable E_{today} in Equation (2) to address the incorrect calculations due to power failure. Equation (3) shows the formula used to find $E_{24\text{h}}$ at the edge.

$$E_{24\text{h}} = kWh - E_{\text{last-24h}} \tag{3}$$

   where $E_{\text{last-24h}}$ is the kWh energy usage 24 h ago relative to the latest timestamp and kWh is the total energy usage at the time of parameter calculation.
4. $E_{\text{weekly avg}}$: This parameter provides an energy meter's average energy consumption measurement for the last seven days. To calculate the average energy, we first fetch the maximum time-series value of "today's energy consumption (E_{today})" for the last seven days, with the use of 24 h timestamp intervals to address the power failure concerns similar to Equations (2) and (3). Then, use the mathematical conversion to find a specific energy meter's average weekly energy consumption, as shown in Equation (4).

$$E_{\text{weekly avg}} = \frac{\sum_{k=1}^{7} E_{\text{mdk}}}{7} \tag{4}$$

   where E_{md} is the maximum energy value of an energy meter for the last 'kth' day.
5. $E_{\%}$: This variable provides the percentage of energy consumption that increased or decreased in real time compared to last week's average energy usage, $E_{\text{weekly avg}}$. To find the $E_{\%}$ parameter, we first subtract the latest time series value of E_{today} from $E_{\text{weekly avg}}$, by fetching the last available value within the last 2 h time frame and then

use the percentage formula to find the increase or decrease in energy consumed today. Equation (5) shows the formula to calculate the $E_{\%}$ value.

$$E_{\%} = \frac{(E_{\text{weekly avg}} - E_{\text{today}}) \times 100}{E_{\text{weekly avg}}} \tag{5}$$

where $E_{\text{weekly avg}}$ is the average energy consumption of a digital meter for the last seven days while E_{today} is the energy usage since midnight.

6. $E_{\text{total-weekly avg}}$: In this parameter, the rule engine finds the average weekly energy consumption of all the energy meters connected to the edge, except the meter that monitors the input solar energy, by fetching the last $E_{\text{weekly avg}}$ value of each three-phase metering point within last 6 h interval from the database and then using the formula presented in (6).

$$E_{\text{total-weekly avg}} = \sum_{k=1}^{n} E_{\text{weekly avg-k}} \tag{6}$$

where 'n' is the number of energy meter devices configured at the edge, and 'k' is used to sum the $E_{\text{weekly avg}}$ values of different devices.

7. $E_{\text{total-today}}$: The edge instance provides total daily energy consumption information from all the wired and wireless meters, except the meter that monitors the input solar energy. This information is calculated at the edge by fetching all the meters' last E_{today} energy consumption values within the interval of the last 6 h and then adding them together to obtain the total value at the sampling rate of 10 min. Equation (7) shows the formula for calculating the daily real-time energy usage of three-phase energy meters.

$$E_{\text{total-today}} = \sum_{k=1}^{n} E_{\text{today-k}} \tag{7}$$

where 'k' represents the kth energy meter value of E_{today} parameter to find the total energy consumption for today of 'n' energy meters.

8. $E_{\text{total-rt-weekly avg}}$: This parameter provides the average weekly consumption of all the energy meters, except the meter that monitors the input solar energy, like Equation (6), but in real time. For the calculation of this parameter, the edge first fetches the $E_{\text{total-today}}$ values for the last seven days around the same time throughout the day from the database at the rate of ten minutes and intervals of the last 12 h to address power failures. Then, the edge rule engine calculates the weekly average energy consumption by adding the fetched data and dividing it by the number of values as shown in Equation (8).

$$E_{\text{rt-total-weekly avg}} = \frac{\sum_{k=1}^{7} E_{\text{total-today-k}}}{7} \tag{8}$$

where $E_{\text{total-today}}$ presents the daily total energy consumption of the 'kth' energy meter device configured at the edge.

9. E_{total}: The E_{total} parameter provides the energy consumption information of all the energy meters since their installation, except the meter that monitors the input solar energy. Its value is measured by fetching the latest 'E' values of all the meters within a time frame of 6 h to avoid power failure issues and then summing them, as shown in Equation (9).

$$E_{\text{total}} = \sum_{k=1}^{n} E_{\text{k}} \tag{9}$$

3.2.5. Data Filtering

As IoT systems generate and send enormous amounts of data to the cloud, the cost of cloud-backed storage services rises exponentially. In this research, to reduce the number of devices from the cloud, we use a data processing feature at the edge to filter the required data before them it to the cloud by identifying the desired wireless energy metering device and selecting only the essential key-value time series data pairs. Although the edge receives more than 30 energy variables for each meter and then saves them in the local database server, it only sends the wireless meter data with variables mentioned in Table 3. This approach enables the user to monitor the desired data remotely by saving the cost of cloud services. Figure 7 shows the software approach for filtering data at the edge.

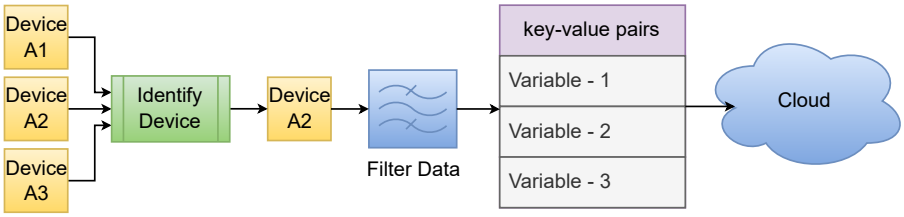


Figure 7. Overview of data filtering at the edge for cloud data.

3.2.6. Device Alarms

In this step, the edge instance monitors the activity and inactivity status of each three-phase metering point by monitoring the latest telemetry data at the edge. When device data are not received at the edge continuously for ten (configurable) minutes, the system raises an alarm event at the graphical interface to alert the user, which is automatically cleared when the device restarts sending the data. Figure 8 shows an overview of the configuration at the backend for managing the activity and inactivity alarms of energy meters at the edge.

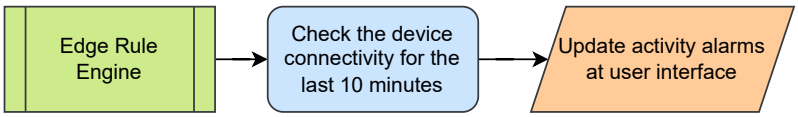


Figure 8. Approach to monitoring activity and inactivity events of devices at the edge.

3.2.7. Edge User Interface Hierarchy

The proposed interface for energy monitoring at the edge provides a master dashboard for overall energy consumption information and a sub-dashboard for each energy metering point for in-depth historical insights from data. Figure 9 provides an overview of the user interface hierarchy at the edge.

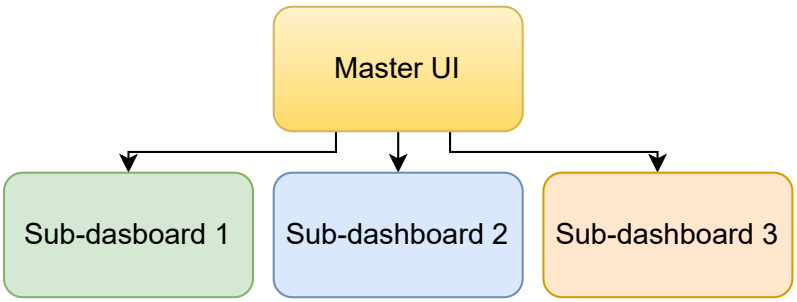


Figure 9. Hierarchy of user interface for energy monitoring at the edge.

3.2.8. Daily Report

Apart from providing an intuitive user interface and energy analytics in real time, the edge instance also generates a daily energy report at 4 p.m. (configurable) consisting of a master dashboard, forwards it to the local dashboard, and sends a copy to the user via an email service implemented at the edge. The daily report enables the admin user to monitor the energy consumption of the factory floor to manage and plan operational activities of industrial machinery.

3.3. Cloud Layer

3.3.1. Overview

An IoT cloud platform offers data visualization and computational services to provide analytical information on devices, including their management via remote internet connectivity [72]. In this study, we utilize ThingsBoard cloud service [55] in combination with the ThingsBoard Edge platform [68] to monitor the energy consumption of a single wireless energy meter per the research scope after applying a data filtering technique at the edge, which provides both the master and individual energy meter dashboards locally. At the cloud, once the telemetry data are received, the rule engine verifies the source of the message, checks if the incoming message is a telemetry key-value pair or a device attribute and finally forwards the data to the respected virtual energy meter device created at the cloud to visualize the data on the dashboard. The next step in the cloud is monitoring the energy meter's activity alarms, including an optional feature to generate the daily report of the user interface. The configured inactivity time for the energy meter device in the cloud is ten minutes, just like on the edge. However, the email feature for the inactivity alarms is only added at the edge level to avoid duplication of email notifications. Figure 10 provides an overview of some of the features implemented in the cloud.

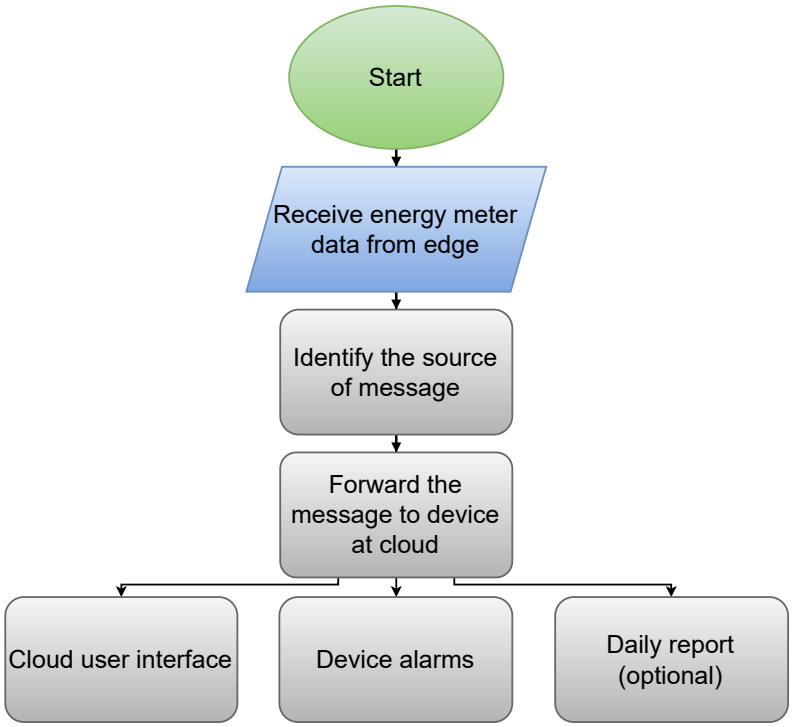


Figure 10. Overview of cloud features for energy monitoring.

3.3.2. Edge vs. Cloud Features

The edge-assisted IoT-based solution in the proposed architecture provides various features, such as visualization of energy analytics via an intuitive user interface, activity and inactivity alarms, daily report scheduling, email service, and a secured user authentication

method. Furthermore, the unlimited API usage characteristics of the edge allow data processing and data filtering services for scalable energy monitoring devices. Another crucial requirement added to the system is the extensive data storage facility at the edge, which, unlike the cloud, ensures the monitoring of substantial historical data and provides a systematic framework to use the data as required for further research. Conversely, even though the current setup only includes one wireless meter in the cloud, the cloud and edge can add more three-phase energy meters if required. Table 4 shows the features of this project used at the edge and cloud layers.

Table 4. Features used at edge and cloud layers in this research.

Feature	Edge	Cloud
Configuration of devices	✓	✓
Device inactivity alarms	✓	✓
Scheduled reporting of user interface	✓	✓
Secured system access	✓	✓
Energy meters sub-dashboard	✓	✓
Email service	✓	Possible
Energy meters master dashboard	✓	Possible, but costly.
Data processing and filtering	✓	Possible, but costly.
Big data storage	✓	Possible, but costly.

4. Results and Discussion

4.1. Hardware

Following the installation of industrial three-phase energy meters and their integration with the edge machine, the system was thoroughly evaluated regarding connectivity and reliability. From the energy meters side, a wireless node at the mobile welding machine is installed to monitor energy consumption and forward the information to the gateway without wireless connectivity issues in the indoor environment, as the wireless gateway was always within 100 m of the range of wireless communication. At the same time, the wired meters also provided continuous energy consumption data via ethernet-based network infrastructure on the factory floor.

4.2. Software

The edge-assisted IoT-based energy monitoring approach has enabled the offloading of processes from the cloud by bringing the computational power closer to the local system, which helps collect and process the data with high scalability and low latency. Once the industrial meter's data are received at the edge via MQTT protocol, the system stores the information in the local database and finds nine additional energy consumption metrics using a single variable, kWh, which has become possible due to unlimited API limitations to process the data using rule engine executions with the help of a database server, unlike the cloud platform, which requires a costly subscription. In the next step, the edge provides local UI and then filters the essential data to send to the cloud for remote monitoring of a single energy meter device. Moreover, during the testing and evaluation of the project, we observed a few compatibility issues between the edge and the cloud, which primarily resulted from the software updates of the cloud and were resolved after updating the version of the edge instance to match compatibility with the cloud.

For the data processing method, the edge rule engine employs various nodes consisting of scripts, data fetching, and data storage functions to handle the energy data of 19 devices. Once an energy meter's message arrives at the edge, it has to go through 25 different nodes in the rule engine at the sampling rate of 1 min, which amounts to 475 rule engine executions per minute. Meanwhile, the weekly average and percentage of increase/decrease in

energy consumption for each meter is calculated using a rule chain consisting of 27 nodes at the sampling rate of 10 min, resulting in 51.3 executions per minute. Moreover, the cumulative analytical parameters are calculated once every 10 min using 33 nodes, except the $E_{\text{total-weekly avg}}$ variable, calculated using 6 nodes every 1 h, resulting in 3.4 rule engine executions per minute. Based on the calculations, the engine executions for each minute are 529.7 per minute or approximately 23.20086 million per month, almost equal to the Rule Engine Execution limits of the ThingsBoard Cloud's minimum subscription plan.

4.3. Edge Performance Evaluation

To evaluate the performance of collecting energy meter data and processing the information at the edge, we used a shell script to monitor the average CPU and memory usage of the edge container running in the Ubuntu-based machine. The experiment was performed when no other monitoring user accessed the edge instance to avoid fluctuations in experimental results. When the energy meters were connected, the script continuously collected the CPU usage (CU) and memory usage (MU) for 10 and 20 min at the sampling rate of 1 s to calculate the average operational usage of the container on the machine. Based on the results, the average calculation for 10 min was 5.58% CU and 16.356% MU, while the average calculation for 20 min was 5.49% CU and 16.356% MU, respectively.

In the next step, the same process was repeated by disconnecting the energy meters and rule chains from the edge, followed by a 15-min delay of no activity, and then executing the script sequentially with the same intervals and sampling rate. This time, while the average calculation for 10 min provided 5.03% CU and 16.364% MU, the average calculation for 20 min was found to be 5.07% CU and 16.361% MU. Based on the experiments, there is a slight decrease in the average CPU usage when the energy meters are disconnected and the data processing feature is disabled; however, no difference was observed for the memory usage between the two states. The low percentage of CU utilization in both states reflects the efficiency of the proposed energy monitoring solution based on edge technology.

Conversely, on the dashboard side, we noticed that when a dashboard contains multiple widgets, and the user selects one month of historical data from the edge, the dashboard sometimes takes longer to load the graphs compared to the cloud-based UI. This delay might be due to the specifications of the edge machine, as it is less powerful than the cloud platform. From the edge computation, although this research utilizes a single-edge instance for processing information and offloading services from the cloud for scalable management of energy meters locally, the proposed solution can further corporate edge-fog computing technology to divide operational usage on a single-edge sever, making it suitable for use cases with distributed application requirements [73].

4.4. Impact of Power Failures

Implementing an interval-based approach in this research allows us to find different metrics when energy meters or edge machines face moderate interruption due to network or power connectivity issues. Based on the observations of the deployed system, it was found that when the edge turns off randomly for a few hours throughout a day, the interval-based technique to fetch the last telemetry data from the database copes with network and power failures, except when the edge is continuously off for one or more than one day which results into delay of up to 7 days in calculating the new energy parameters. Aside from the power failures for a long duration, there is no impact on calculations due to voltage, frequency, and power factor signal fluctuations or varying loads, as this research utilizes a single cumulative variable kWh to find energy parameters. Figure 11 provides an overview of parameter calculation at the edge when the machine is started for the first time or if it restarts again after one or more days.

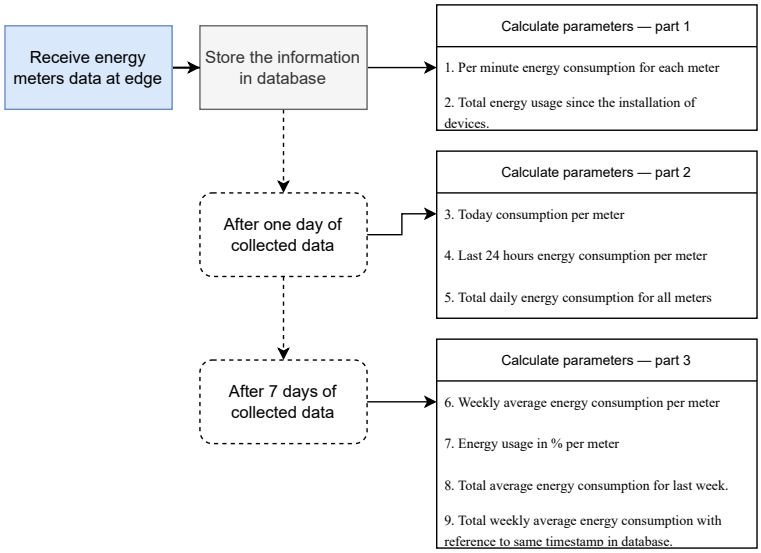


Figure 11. Overview of parameters calculation at edge.

4.5. Scalability

The proposed system architecture is designed to provide scalability features from the energy meters layer to the cloud layer. As the integration of the ThingsBoard Edge platform allows to add upto 1000 devices depending on the type of machine [74], the edge machine utilized in the first phase of this research is compatible with the number of monitoring devices, which can be easily increased by upgrading the local edge machine. From the communication side, the related IoT-based energy monitoring articles widely utilize MQTT [38,40,43], HTTP [16,17,41], NB-IoT [37], and Modbus TCP/IP [35,35] protocols for the transmission of energy meter's data. Using a particular communication protocol depends on the application; however, using a lightweight MQTT protocol helps achieve high interoperability and scalability with broker-based data exchange. Moreover, in comparison to the smart grid protocols, as defined by the IEC 61850 standard [75], which is mainly used in power substations for mission-critical applications and high reliability, the use of the MQTT protocol allows for the integration of a wide range of other IoT devices to the edge machine, making it compatible with the requirements for a multi-application-based IoT monitoring system.

4.6. Security

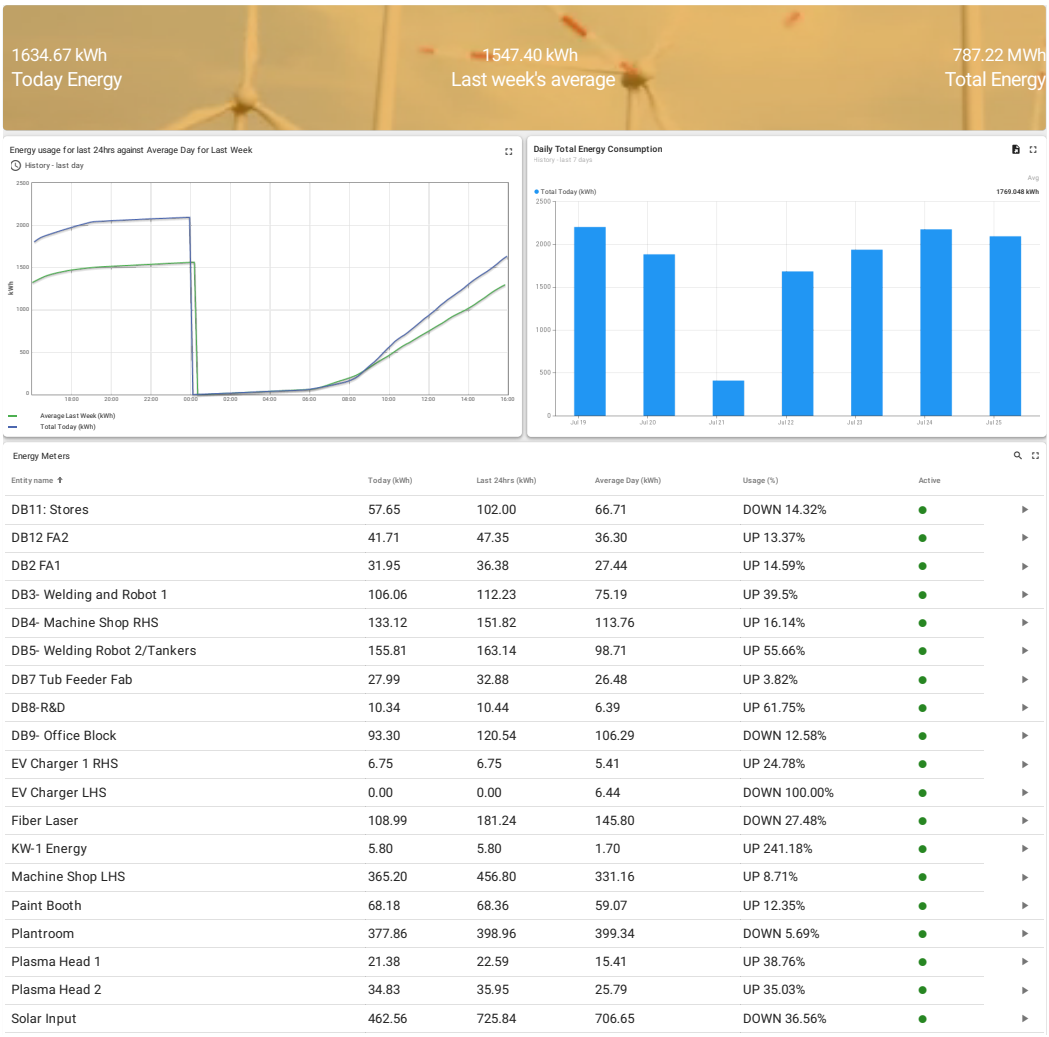
In this project, the security concerns are given moderate consideration as the key objective of the research was to provide a monitoring system for an industrial factory floor where the information technology (IT) devices connected to the LAN are deemed less of a threat to security. The connected energy meters send the data using port 1883 of the MQTT protocol, which uses a username and password without encryption; however, for stringent security requirements, we have proposed a network security plan for future work. Conversely, from the edge side, the docker container isolates the database server, ensuring security concerns in our application. Moreover, the end-to-end security between the edge and cloud communication is ensured using TLS/SSL-based encrypted communication [76].

4.7. User Interface

The proposed architecture provided the monitoring of UI on both the edge and cloud platforms and was designed based on the quality feedback from the monitoring user. The local UI can be accessed from the edge machine and any other computer within the LAN using ethernet-based communication. In contrast, the cloud-based UI can be accessed from anywhere and anytime via the internet. Based on this research's scope, the edge consists of a master dashboard for all the energy meters and an individual dashboard state for each energy meter to further provide in-depth insights, including alarm information. Conversely,

the edge is also designed to provide the reporting feature by sending the PDF copy of the master UI to the monitoring user via email; however, this feature still requires more work as sometimes the edge forwards the UI image with empty data for the time series widgets, which might be resulting from the timeout of loading data from the edge and cache support of the local browser.

From the parameters side, the master UI provides a dashboard containing the parameters calculated at the edge in a tabular form. The master dashboard provides the daily, last 24 h, average energy consumption information, and the percentage of energy increased/decreased over time. Moreover, it provides a comparative widget showing the daily total energy usage against last week's average consumption to monitor the trend in data in real time. The UI also provides a few other overall energy consumption parameters, such as total energy consumption, average energy usage for the last week, and the total energy usage since the energy meter installation. Based on the evaluation of the project, as the industrial factory floor is operational for five working days a week, the daily energy consumption of a single meter and the cumulative energy usage of all the meters usually show higher as compared to the weekly average energy usage that is calculated based on the last day's data, including the weekends. While the energy data of weekends can also be excluded from the calculation of the average energy consumption parameter, their comparative analysis also provides key insights about the low operational activity of a machine when the energy usage (%) shows a DOWN value. Figure 12 shows the master user interface developed for all the energy meters at the edge instance.



Entity name	Today (kWh)	Last 24hrs (kWh)	Average Day (kWh)	Usage (%)	Active
DB11: Stores	57.65	102.00	66.71	DOWN 14.32%	
DB12 FA2	41.71	47.35	36.30	UP 13.37%	
DB2 FA1	31.95	36.38	27.44	UP 14.59%	
DB3- Welding and Robot 1	106.06	112.23	75.19	UP 39.5%	
DB4- Machine Shop RHS	133.12	151.82	113.76	UP 16.14%	
DB5- Welding Robot 2/Tankers	155.81	163.14	98.71	UP 55.66%	
DB7 Tub Feeder Fab	27.99	32.88	26.48	UP 3.82%	
DB8-R&D	10.34	10.44	6.39	UP 61.75%	
DB9- Office Block	93.30	120.54	106.29	DOWN 12.58%	
EV Charger 1 RHS	6.75	6.75	5.41	UP 24.78%	
EV Charger LHS	0.00	0.00	6.44	DOWN 100.00%	
Fiber Laser	108.99	181.24	145.80	DOWN 27.48%	
KW-1 Energy	5.80	5.80	1.70	UP 241.18%	
Machine Shop LHS	365.20	456.80	331.16	UP 8.71%	
Paint Booth	68.18	68.36	59.07	UP 12.35%	
Plantroom	377.86	398.96	399.34	DOWN 5.69%	
Plasma Head 1	21.38	22.59	15.41	UP 38.76%	
Plasma Head 2	34.83	35.95	25.79	UP 35.03%	
Solar Input	462.56	725.84	706.65	DOWN 36.56%	

Figure 12. Master user interface for energy monitoring at the edge.

The interactive master dashboard allows users to visualize the insights for individual energy meters and easily access the individual energy monitoring dashboard by clicking the play sign on the right side of rows in the master dashboard. It includes the visualization of per minute, daily energy, and total energy consumption information forwarded by the edge instance, along with the activity status of energy meters. At the same time, the cloud only contains a single wireless energy meter as per the research scope and due to the limitations of the current cloud platform subscription plan. Figure 13 shows the user interface of a three-phase wireless energy meter at the edge and cloud.

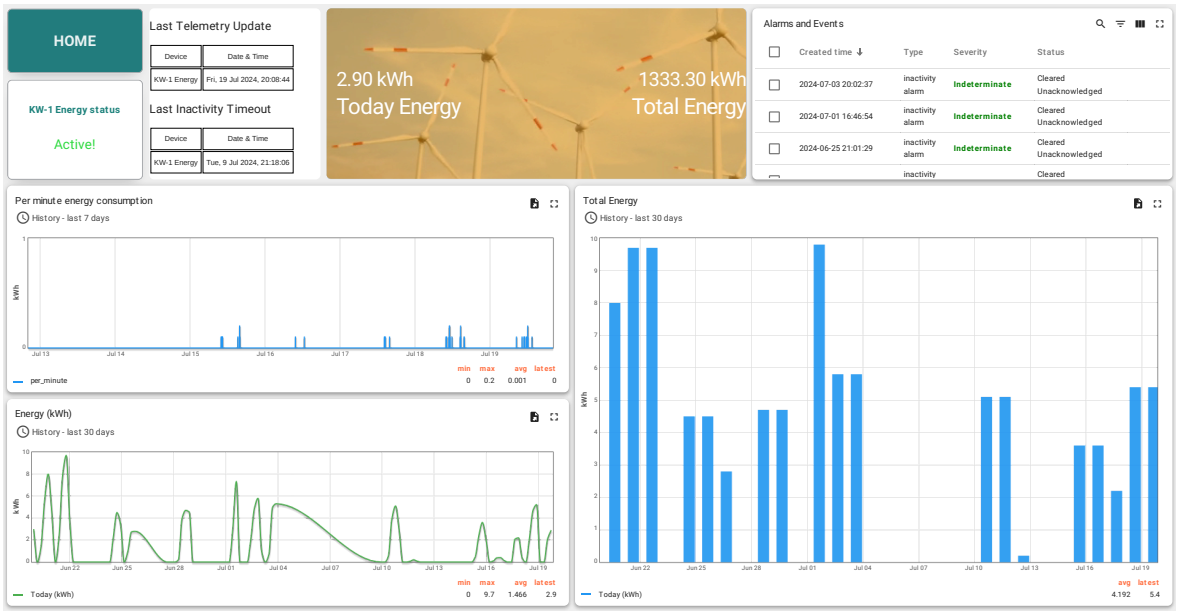


Figure 13. User interface for a wireless energy meter sub-dashboard at the edge and in the cloud.

5. Conclusions and Future Work

This research presented an IoT-based energy monitoring approach based on an edge data processing method using JavaScript-based rule engine execution and an edge database for an industrial factory floor. It uses industrial wireless and wired energy meters that monitor the energy consumption of mobile and stationary assets with high accuracy in the industrial environment and forward the collected data to a local edge instance. The edge server processes the collected information by first enriching the database and then transforming the data based on the latest and historical time series values to find nine new energy consumption parameters using a single variable, kWh, which provides cumulative energy usage value over time. For each energy meter, the edge utilizes the rule engine feature to calculate the energy usage for every minute, daily, last 24 h, average weekly, and percentage of consumption increased/decreased compared to average weekly usage for every meter. The edge instance also calculates total energy consumption parameters for all the combined energy meters, such as total daily energy usage, average energy usage in real time, average energy for the last week, and the total energy usage since the installation of the energy meters. Following the calculation of parameters, the edge instance provides a master dashboard and individual dashboard states of all the energy meters, including the alarms and reporting feature, then forwards the filtered data to the cloud that provides the remote monitoring of a single wireless energy meter along with its activity and inactivity alarms. While the master UI provides comparative insights from the meter nodes and overall energy consumption analytics in real time, the individual dashboard states for each meter assist the end user in obtaining in-depth visual insights into the energy consumption of different machines and sections that can assist in moderate energy consumption and planning the operational activities on the factory floor. Based on the assessment of the system, the proposed solution effectively provides the comparative energy monitoring of

industrial processes at the edge with an average CPU utilization of up to 5.58% along with interval-based data retrieval techniques in the data processing to avoid measurement errors due to random power and network connectivity issues throughout the day.

Based on the evaluation and discussion of this research, the proposed system lays the foundation for integrating more devices at the edge and cloud, including transmitting telemetry data from existing energy meter devices as per requirements by upgrading the cloud subscription. Moreover, the use of a robust industrial server would further allow the expansion of the number of IoT devices. As each energy meter provides more than 30 energy consumption variables per minute stored in the database, the edge instance with advanced processing power would create more opportunities to find new parameters using various data processing techniques and optimized algorithms. Another direction might be upgrading data processing features to address power failure events that remain for one or more than one day. From the security side, the proposed architecture aims to enhance the security between the energy meters and edge machine by using a combination of virtual private network (VPN) and network firewall. While the VPN would ensure the isolation of the connected devices and provide some privacy, using a firewall would further limit the accessibility of devices in the local network. An additional layer of security could also be added by employing MQTT with 8883 TLS-based encrypted communication for security and privacy-preserved communication at the device layer. Moreover, a key direction for future research would be to utilize the proposed architecture by extracting the energy data from the edge database and applying machine learning techniques to detect anomalies in the system. It can further help the user gain more insights and use the data for predictive maintenance of large industrial machinery.

Author Contributions: Conceptualization, A.A.M.; methodology, A.A.M.; software, A.A.M.; validation, A.A.M.; formal analysis, A.A.M.; investigation, A.A.M.; resources, A.A. and J.W.; data curation, A.A.M.; writing—original draft preparation, A.A.M.; writing—review and editing, A.A.M., N.O. and A.A.; visualization, A.A.M.; supervision, A.A.; project administration, A.A. and J.W.; funding acquisition, A.A. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science Foundation Ireland under Grant No. 16/RC/3918.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors acknowledge the support and guidance of the IMaR team at Munster Technological University.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
CT	Current Transformer
IA	Industrial Automation
IIoT	Industrial Internet of Things
IoT	Internet of Things
IT	Information Technology
RFID	Radio-Frequency Identification
DSM	Demand Side Management
FPGA	Field Programmable Gate Array
HTTP	Hypertext Transfer Protocol
NB-IOT	NarrowBand-Internet of Things
MQTT	Message Queuing Telemetry Transport
JSON	JavaScript Object Notation

kWh	Kilowatt-hour
LAN	Local Area Network
LCD	Liquid Crystal Display
OT	Operational Technology
PCB	Printed Circuit Board
SCADA	Supervisory Control And Data Acquisition
CU	Central Processing Unit Usage
MU	Memory Usage
VPN	Virtual Private Network

References

1. Stern, D.I.; Burke, P.J.; Bruns, S.B. The Impact of Electricity on Economic Development: A Macroeconomic Perspective; UC Berkeley: Center for Effective Global Action. 2019. Available online: https://escholarship.org/uc/item/7jb0015q (accessed on 26 September 2024).
2. Electricity Explained—U.S. Energy Information Administration (EIA). Available online: https://www.eia.gov/energyexplained/electricity/ (accessed on 26 September 2024).
3. Stern, D.; Burke, P.; Bruns, S. The Impact of Electricity on Economic Development: A Macroeconomic Perspective. *Int. Rev. Environ. Resour. Econ.* **2018**, *12*, 85–127. [CrossRef]
4. International Energy Outlook 2023—U.S. Energy Information Administration (EIA). Available online: https://www.eia.gov/outlooks/ieo/narrative/index.php (accessed on 26 September 2024).
5. Raimi, D.; Zhu, Y.; Newell, R.G.; Prest, B.C.; Bergman, A. Global Energy Outlook 2023: Sowing the Seeds of an Energy Transition. Technical Report 23-02, Resources for the Future. 2023. Available online: https://www.rff.org/publications/reports/global-energy-outlook-2023/ (accessed on 26 September 2024).
6. The Rise of IT/OT Convergence in Industry 4.0 | Engineering.com. Available online: https://www.engineering.com/story/the-rise-of-itot-convergence-in-industry-40 (accessed on 26 September 2024).
7. Hossein Motlagh, N.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of Things (IoT) and the Energy Sector. *Energies* **2020**, *13*, 494. [CrossRef]
8. Serror, M.; Hack, S.; Henze, M.; Schuba, M.; Wehrle, K. Challenges and Opportunities in Securing the Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2985–2996. [CrossRef]
9. Qiu, T.; Chi, J.; Zhou, X.; Ning, Z.; Atiquzzaman, M.; Wu, D.O. Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2462–2488. [CrossRef]
10. Nain, G.; Pattanaik, K.; Sharma, G. Towards edge computing in intelligent manufacturing: Past, present and future. *J. Manuf. Syst.* **2022**, *62*, 588–611. [CrossRef]
11. Kong, L.; Tan, J.; Huang, J.; Chen, G.; Wang, S.; Jin, X.; Zeng, P.; Khan, M.; Das, S.K. Edge-computing-driven Internet of Things: A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–41. [CrossRef]
12. Sobin, C.C. A Survey on Architecture, Protocols and Challenges in IoT. *Wirel. Pers. Commun.* **2020**, *112*, 1383–1429. [CrossRef]
13. Karpagam, M.; Sahana, S.S.; Sivadharini, S.; Soundhariyasri, S. Smart Energy Meter and Monitoring System using Internet of Things (IoT). In Proceedings of the IDCIoT 2023—International Conference on Intelligent Data Communication Technologies and Internet of Things, Proceedings, Bengaluru, India, 5–7 January 2023; pp. 75–80. [CrossRef]
14. Reddy, V.M.K.; Malavika, K.V.; Lokasree, B.S.; Kumar, K.N. IOT based Smart Meter Using Node-Red. In Proceedings of the 2023 International Conference on Artificial Intelligence and Smart Communication, AISC 2023, Greater Noida, India, 27–29 January 2023; pp. 931–934. [CrossRef]
15. Syafii; Luthfi, A.; Rozzi, Y.A. Design of Raspberry Pi Web-based Energy Monitoring System for Residential Electricity Consumption. In Proceedings of the 2020 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 19–23 October 2020; pp. 192–196. [CrossRef]
16. Cheddadi, Y.; Cheddadi, H.; Cheddadi, F.; Errahimi, F.; Es-sbai, N. Design and implementation of an intelligent low-cost IoT solution for energy monitoring of photovoltaic stations. *SN Appl. Sci.* **2020**, *2*, 1165. [CrossRef]
17. Rani, D.D.P.; Suresh, D.; Kapula, P.R.; Akram, C.H.M.; Hemalatha, N.; Soni, P.K. IoT based smart solar energy monitoring systems. *Mater. Today Proc.* **2023**, *80*, 3540–3545. [CrossRef]
18. Bhau, G.V.; Deshmukh, R.G.; kumar, T.R.; Chowdhury, S.; Sesharao, Y.; Abilmazhinov, Y. IoT based solar energy monitoring system. *Mater. Today Proc.* **2023**, *80*, 3697–3701. [CrossRef]
19. Ch, K.R.; Sahoo, S.K.; Yanine, F. An IoT-based intelligent smart energy monitoring system for solar PV power generation. *Energy Harvest. Syst.* **2023**, *11*, 20230015. [CrossRef]
20. Ziane, A.; Dabou, R.; Necaibia, A.; Rouabhia, A.; Bouchouicha, K.; Sahouane, N.; Lachtar, S.; Bouraiou, A.; Larbi, A.A. IoT Platform For Online Monitoring Of Renewable Energy Systems. In Proceedings of the 2022 3rd International Conference on Embedded & Distributed Systems (EDiS), Oran, Algeria, 2–3 November 2022; pp. 55–60. [CrossRef]
21. Hussain, S.S.; Khattak, K.S.; Khan, A.; Khan, Z.H. Cyber physical system for solar energy monitoring. In Proceedings of the 2019 International Conference on Frontiers of Information Technology, FIT 2019, Islamabad, Pakistan, 16–18 December 2019; pp. 185–190. [CrossRef]

22. Ali, M.; Paracha, M.K. An Iot Based Approach for Monitoring Solar Power Consumption with Adafruit Cloud. *Int. J. Eng. Appl. Sci. Technol.* **2020**, *4*, 335–341. [CrossRef]
23. Reddy, B.M.; Reddy, M.U.G.; Paruchuri, S.; Reddy, V.P.; Sunny, R. An Iot-Based Sustainable Energy Monitoring And Alert System With Billing Integration. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 6–8 July 2023; pp. 1–7. [CrossRef]
24. Anusha, M.; Kumar, P.B.; Akhil, V.; Gouthami, M.; Chinnaaiah, M.; Shaik, S. Internet of Things (IOT) based energy monitoring with ESP 32 and using Thingspeak. In Proceedings of the 2024 10th International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 12–14 April 2024; pp. 1383–1387. [CrossRef]
25. Allirani, S.; Karthik, M.; Krishiba, S.S.; Shubha, P.; Sujima, S. IoT Based Real Time Energy Billing and Monitoring. In Proceedings of the 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 10–11 August 2023; pp. 428–432. [CrossRef]
26. Joha, M.I.; Rahman, M.M.; Zubair, M.I. IoT-Based Smart Energy Monitoring, Management, and Protection System for a Smart MicroGrid. In Proceedings of the 2024 Third International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 18–20 January 2024; pp. 398–403. [CrossRef]
27. Tsyrulnyk, S.; Tromsyuk, V.; Tsyrulnyk, M.; Rymar, P. Energy Monitoring System based on IoT. *CEUR Workshop Proc.* **2021**, *3039*, 136–153.
28. Soh, Z.H.C.; Hamzah, I.H.; Abdullah, S.A.C.; Shafie, M.A.; Sulaiman, S.N.; Daud, K. Energy consumption monitoring and alert system via IoT. In Proceedings of the 2019 International Conference on Future Internet of Things and Cloud, FiCloud 2019, Istanbul, Turkey, 26–28 August 2019; pp. 265–269. [CrossRef]
29. Santos, D.; Ferreira, J.C. IoT Power Monitoring System for Smart Environments. *Sustainability* **2019**, *11*, 5355. [CrossRef]
30. Ramelan, A.; Adriyanto, F.; Hermanu, B.; Ibrahim, M.H.; Saputro, J.S.; Setiawan, O. IoT Based Building Energy Monitoring and Controlling System Using LoRa Modulation and MQTT Protocol. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1096*, 012069. [CrossRef]
31. Reddy, K.K.; Kumar, N.V.K.; Vani, T.; Khan, P.M.; Reddy, C.B.; Monika, P.; Sambhavi, T. Iot Based Smart Energy Meter Monitoring and Controlling System. *Int. J. Mech. Eng.* **2022**, *7*, 974–5823.
32. Muhammed, A.W.O.; Oisamoje, V.; Amhenrior, H.E.; Evbogbai, E.M.; Abanihi, V.K.; Bello, L.O.; Obasi, C.C. Design and Implementation of an IoT Based Home Energy Monitoring System. In Proceedings of the 5th International Conference on Information Technology for Education and Development: Changing the Narratives Through Building a Secure Society with Disruptive Technologies, ITED 2022, Abuja, Nigeria, 1–3 November 2022. [CrossRef]
33. Abbasi, O.U.R.; Bukhari, S.B.A.; Iqbal, S.; Abbasi, S.W.; Rehman, A.U.; AboRas, K.M.; Alshareef, M.J.; Ghadi, Y.Y. Energy management strategy based on renewables and battery energy storage system with IoT enabled energy monitoring. *Electr. Eng.* **2023**, *106*, 3031–3043. [CrossRef]
34. Nandish, B.M.; Pushparajesh, V. Economic operation of residential load using IOT-based renewable energy management system. *Electr. Eng.* **2024**. [CrossRef]
35. Pavithra, C.V.; Divya, R.; Suresh, B.; Deepak, K.; Sabaresh, B. IoT Based Smart Energy Monitoring in Multi Layer Car Parking. In Proceedings of the 2023 International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (iTech SECOM), Coimbatore, India, 18–19 December 2023; pp. 82–86. [CrossRef]
36. Perkasa, S.D.; Suwarno, I.; Megantoro, P.; Muzadi, A.R.; Sukmawan, I.H. Towards Sustainable Energy Distribution: IoT-Based Monitoring of Electrical Load Parameters in a Hybrid Renewable Power Plant. In Proceedings of the 2023 6th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 10 November 2023; pp. 29–34. [CrossRef]
37. Siva Balan, R.; Gouri, M.; Senthilnathan, T.; Gondkar, S.R.; Gondar, R.R.; Loveline Zeema, J.; Jothikumar, R. Development of smart energy monitoring using NB-IOT and cloud. *Meas. Sens.* **2023**, *29*, 100884. [CrossRef]
38. Anand, T.; Upare, S.; Jain, S.; Andhare, M.; Bhangale, K. Deployment of Real-Time Energy Monitoring System Using IoT. In Proceedings of the 2022 3rd International Conference for Emerging Technology, INCET 2022, Belgaum, India, 27–29 May 2022. [CrossRef]
39. Karthick, T.; Chandrasekaran, K. Design of IoT based smart compact energy meter for monitoring and controlling the usage of energy and power quality issues with demand side management for a commercial building. *Sustain. Energy Grids Netw.* **2021**, *26*, 100454. [CrossRef]
40. Saleem, M.U.; Usman, M.R.; Usman, M.A.; Politis, C. Design, Deployment and Performance Evaluation of an IoT Based Smart Energy Management System for Demand Side Management in Smart Grid. *IEEE Access* **2022**, *10*, 15261–15278. [CrossRef]
41. Saluja, S.; Prochnow, C.; Couper, G.; Zivko, F.; Yu, X.; Gromov, M.; Saniie, J. Smart Multi-Building Energy Monitoring System. In Proceedings of the 2024 IEEE International Conference on Electro Information Technology (eIT), Eau Claire, WI, USA, 30 May–1 June 2024; pp. 342–347. [CrossRef]
42. Govindarajan, R.; Meikandasivam, S.; Vijayakumar, D. Performance Analysis of Smart Energy Monitoring Systems in Real-time. *Eng. Technol. Appl. Sci. Res.* **2020**, *10*, 5808–5813. [CrossRef]
43. Zhao, L.; Matsuo, I.B.M.; Zhou, Y.; Lee, W.J. Design of an Industrial IoT-Based Monitoring System for Power Substations. *IEEE Trans. Ind. Appl.* **2019**, *55*, 5666–5674. [CrossRef]
44. Mudaliar, M.D.; Sivakumar, N. IoT based real time energy monitoring system using Raspberry Pi. *Internet Things* **2020**, *12*, 100292. [CrossRef]

45. Munoz, O.; Ruelas, A.; Rosales-Escobedo, P.F.; Acuña, A.; Suastegui, A.; Lara, F.; Reyes-Zamora, R.A.; Rocha, A. Development of an IoT smart energy meter with power quality features for a smart grid architecture. *Sustain. Comput. Inform. Syst.* **2024**, *43*, 100990. [CrossRef]
46. Bonner Monitoring Systems. Available online: https://bonner.ie/service/automation-platforms/monitoring-systems/ (accessed on 26 September 2024).
47. SIMATIC Energy Manager—Certified Energy Management. Available online: https://www.siemens.com/global/en/products/automation/industry-software/automation-software/energymanagement/simatic-energy-manager-pro.html (accessed on 26 September 2024).
48. Optii Data Analytics Platform. Available online: https://optima.energy/optii-data-analytics-platform/ (accessed on 26 September 2024).
49. DIAEnergie Industrial Energy Management System. Available online: https://deltathailand.com/en/products/Automation/Industrial-Automation/Software-System/DIAEnergie (accessed on 26 September 2024).
50. Energy Monitoring Software for Manufacturers. Available online: https://fourjaw.com/features/energy-monitoring (accessed on 26 September 2024).
51. EnergyCAP Smart Analytics. Available online: https://www.energycap.com/energy-monitoring-software/ (accessed on 26 September 2024).
52. Eniscope Analytics. Available online: https://analytics.eniscope.com/login?go=%2F (accessed on 26 September 2024).
53. Industrial Energy Monitoring and Reporting Software. Available online: https://new.abb.com/industrial-software/sustainability/energy-management-system-ems/energy-monitoring-and-reporting (accessed on 26 September 2024).
54. Data Processing for Energy Management and Energy Analytics. Available online: https://www.weidmueller.com/int/products/automation_software/measuring_and_monitoring_systems/software/index.jsp (accessed on 26 September 2024).
55. ThingsBoard Cloud. Available online: https://thingsboard.io/docs/paas/ (accessed on 26 September 2024).
56. AWS IoT. Available online: https://aws.amazon.com/iot/ (accessed on 26 September 2024).
57. Azure IoT. Available online: https://azure.microsoft.com/en-us/solutions/iot (accessed on 26 September 2024).
58. Garcia, D. How Do Smart Meters Communicate? Available online: https://www.emnify.com/blog/how-smart-meters-communicate (accessed on 26 September 2024).
59. Energy Monitoring. Available online: https://www.best.energy/solutions/energy-monitoring/ (accessed on 26 September 2024).
60. *IEC 60044-1:1996*; Instrument Transformers—Part 1: Current Transformers. Standard, International Electrotechnical Commission: Geneva, Switzerland, 1996.
61. 3 Phase 4 Wire System—Your Electrical Guide. Available online: https://www.yourelectricalguide.com/2018/12/3-phase-4-wire-system.html (accessed on 26 September 2024).
62. Wireless Electricity Monitor (ZEM). Available online: https://episensor.com/product/wireless-electricity-monitor/ (accessed on 26 September 2024).
63. Gateway (NGR-30-5) Datasheet. Available online: https://episensor.com/document/datasheet-ngr-30-5-gateway-epi-077-01/ (accessed on 26 September 2024).
64. *IEC 62053-22*; Electricity Metering Equipment—Particular Requirements—Part 22: Static Meters for AC Active Energy (Classes 0.1S, 0.2S and 0.5S). Standard, International Electrotechnical Commission: Geneva, Switzerland, 2020.
65. Bourechak, A.; Zedadra, O.; Kouahla, M.N.; Guerrieri, A.; Seridi, H.; Fortino, G. At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. *Sensors* **2023**, *23*, 1639. [CrossRef] [PubMed]
66. Chougule, S.B.; Chaudhari, B.S.; Ghorpade, S.N.; Zennaro, M. Exploring Computing Paradigms for Electric Vehicles: From Cloud to Edge Intelligence, Challenges and Future Directions. *World Electr. Veh. J.* **2024**, *15*, 39. [CrossRef]
67. Wang, Z.; Sun, Y.; Liu, D.; Hu, J.; Pang, X.; Hu, Y.; Ren, K. Location Privacy-Aware Task Offloading in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2024**, *23*, 2269–2283. [CrossRef]
68. ThingsBoard PE Edge. Available online: https://thingsboard.io/docs/pe/edge/ (accessed on 26 September 2024).
69. Dell OptiPlex 5060 Tower Setup and Specifications Guide. Available online: https://www.dell.com/support/manuals/en-ie/optiplex-5060-desktop/optiplex_5060_mt_setup_and_specs/processor?guid=guid-b6e22f5e-0a6e-434f-b097-e21f7ea9e9b4&lang=en-us (accessed on 26 September 2024).
70. Singh, M.; Rajan, M.; Shivraj, V.; Balamuralidhar, P. Secure MQTT for Internet of Things (IoT). In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 746–751. [CrossRef]
71. What Is Data Processing? Available online: https://www.talend.com/resources/what-is-data-processing/ (accessed on 26 September 2024).
72. Asemani, M.; Abdollahei, F.; Jabbari, F. Understanding IoT Platforms: Towards a comprehensive definition and main characteristic description. In Proceedings of the 2019 5th International Conference on Web Research, ICWR 2019, Tehran, Iran, 24–25 April 2019; pp. 172–177. [CrossRef]
73. Diamanti, M.; Charatsaris, P.; Tsiropoulou, E.E.; Papavassiliou, S. Incentive Mechanism and Resource Allocation for Edge-Fog Networks Driven by Multi-Dimensional Contract and Game Theories. *IEEE Open J. Commun. Soc.* **2022**, *3*, 435–452. [CrossRef]
74. FAQ | ThingsBoard Edge. Available online: https://thingsboard.io/docs/edge/faq/ (accessed on 26 September 2024).

75. IEC 61850 Compliance and Verification Services. Available online: https://www.dnv.com/services/iec-61850-compliance-and-verification-69301/ (accessed on 26 September 2024).
76. Edge over TLS/SSL. Available online: https://thingsboard.io/docs/edge/user-guide/grpc-over-ssl/ (accessed on 26 September 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.