Construction of Smart Grid Load Forecast Model by Edge Computing

Xudong Pang [1], Xiangchen Lu [2], Hao Ding [1,*] and Josep M. Guerrero [3]

1 Electrical Engineering Department, Yanshan University, Qinhuangdao 066000, China; pangxdong@stumail.ysu.edu.cn
2 School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China; lxc2049442462@163.com
3 Department of Energy Technology, Aalborg University, 9220 Aalborg, Denmark; joz@et.aau.dk
* Correspondence: dinghao@ysu.edu.cn

Abstract: This research aims to minimize the unnecessary resource consumption by intelligent Power Grid Systems (PGSs). Edge Computing (EC) technology is used to forecast PGS load and optimize the PGS load forecasting model. Following a literature review of EC and Internet of Things (IoT)-native edge devices, an intelligent PGS-oriented Resource Management Scheme (RMS) and PGS load forecasting model are proposed based on task offloading. Simultaneously, an online delay-aware power Resource Allocation Algorithm (RAA) is developed for EC architecture. Finally, comparing three algorithms corroborate that the system overhead decreases significantly with the model iteration. From the 40th iteration, the system overhead stabilizes. Moreover, given no more than 50 users, the average user delay of the proposed delay-aware power RAA is less than 13 s. The average delay of the proposed algorithm is better than that of the other two algorithms. This research contributes to optimizing intelligent PGS in smart cities and improving power transmission efficiency.

Keywords: edge computing; intelligent Power Grid System (PGS); PGS load; resource management

1. Introduction

It is reasonable to assume that power facilities empower modern social and urban development. In particular, an intelligent Power Grid System (PGS) has risen as a powerful technological means for supporting urban facilities [1–3] given the defects of the traditional PGS, such as high transmission cost and excessive on-the-line energy loss. These deficiencies affect the economic benefits of the PGS by increasing residential costs and decreasing grid transmission utilization.

With the further progress of network architecture technology, the Cloud Computing network has gradually broken through the existing limitations. Furthermore, Edge Computing (EC) has become a new way of collecting and processing network resources [4–6]. IoT architecture, based on enhanced data analysis and processing capabilities of IoT devices, minimizes resource allocation time [7] and improves system performance. Therefore, digital and intelligent upgrade of the comprehensive performance of the EC-empowered intelligent PGS can improve the system's grid load performance and power transmission capacity [8–10].

This work uses EC technology as the technical framework. It constructs the PGS wireless communication model based on a Mobile Edge Computing (MEC) network by analyzing the PGS-oriented Resource Management Scheme (RMS). As a result, a distributed PGS-oriented RMS is proposed based on matching theory and a delay-aware online power Resource Allocation Algorithm (RAA). The experimental results verify the effectiveness of the proposed algorithms. Section 1 briefly introduces the background of EC and intelligent PGSs. Section 2 organizes the deficiencies of previous research on the intelligent PGS load estimation through literature research and designs the distributed PGS-oriented RMS based

on matching theory. Section 3 proposes the MEC network and PGS model based on research related to task-offloading RMS. Then, Section 4 conducts the experiments to analyze the variation range of the system delay and overhead parameters. Section 5 summarizes the experimental results. The research reported here has practical reference value for digital and intelligent development of intelligent PGSs.

2. Recent Related Work

2.1. EC and IoT-Native Edge Devices

Many scholars have researched edge computing technology. For example, ref. [11] integrated EC and Deep Learning (DL) and researched the services and development model of Artificial Intelligence (AI) applications. The research results showed that integrating DL with the EC framework could provide an intelligent edge and realize dynamic and adaptive edge maintenance and management. Furthermore, unleashing deep learning services with resources at the network edge close to data sources has become an ideal research solution. Deng et al. [12] studied fusing EC and AI to optimize AI models under DL and improved hardware architecture. The results offered technical support for edge intelligence exploration. Deng et al. [13] researched the architecture and technical application of the MEC-based Mobile Augmented Reality (MAR). They found that MAR enhanced human perception of the world by combining natural environments with virtual spaces. Further, with the explosion of powerful, inexpensive mobile devices and the advent of complex communication infrastructure, mobile augmented reality applications could become popularized. Siriwardhana et al. [14] investigated the security and privacy of multiaccess EC systems to improve mobile network data processing efficiency. Furthermore, they analyzed the vulnerabilities leading to identified threat vectors and proposed potential security solutions to overcome these vulnerabilities.

The continuous development of EC technology places greater requirements on IoT edge-device security architecture and performance. Ranaweera et al. [15] researched AI security architecture for the IoT edge layer. They established an architecture on top of an AI security module at the edge layer to protect IoT infrastructure. The proposed module outperformed the Cyber Kill Chain model in intrusion detections. HaddadPajouh et al. [16] modeled the behavior of IoT–EC environments and designed a simulation framework. The findings demonstrated that the IoT–EC fabric could monitor the computing behavior of the infrastructure and allow users to test their infrastructure and frameworks in a configurable and straightforward manner. The collaborative MEC-oriented resource allocation and optimization was explored in [17]. They proposed two-edge server deployment schemes based on software-defined network technology: the enumeration-based optimal edge server deployment scheme and ranking-based near-optimal edge server placement algorithm. The proposed algorithm optimized the collaborative edge server deployment The total service configuration cost of the clustering-enhanced heuristic service algorithm was about 0.89 under eight service types. Lv and Liang [18] studied smart-cities-oriented intelligent EC based on machine learning. Focusing on the mobile edge server, they used the available resources around the mobile edge server for collaborative computing to improve the computing performance of the MEC system. AI-driven IoT–EC and cloud computing technologies were researched by [19]. Through mixed discussion of all important aspects of basic technology, they provided comprehensive research and accelerated knowledge acquisition opportunities and furnished the EC paradigm with comprehensive technical support. Firouzi et al. [20] discussed the opportunities and challenges of federated learning in IoT. They employed the Internet to bring more mobile network endpoints, thus reducing costs. Additionally, centralized cloud-based learning could help users solve user data security privacy issues.

2.2. Intelligent PGS Load and Forecast Models

The research on the PGS load prediction relies on the information exchange technology. Zhang et al. [21] predicted PGS load based on the heuristic algorithm by proposing a new

technology for feature selection based on improved mutual information. In addition, they constructed a novel hybrid short-term PGS load forecasting model. The results showed that the proposed algorithms improved the PGS load prediction accuracy. Hafeez et al. [22] utilized blockchain technology to balance the load in the energy trading market. They proposed a model to implement an efficient hybrid energy trading market while reducing costs and power peak-to-average ratios. Then, they checked the performance of the proposed system through simulation experiments. The results indicated that the research system effectively reduced the target cost. Khalid et al. [23] investigated renewable energy and PGS load forecasting methods for innovative grid operation. For PGS load forecasting, they utilized an improved stacked-gated Recurrent Neural Network (RNN). The findings proved that the proposed method outperformed state-of-the-art methods in machine learning or DL by achieving accurate energy predictions for intelligent grid operation. Xia et al. [24] examined power load clusters in smart grids and analyzed consumers' energy consumption patterns and preferences through data mining. They also summarized the basic concepts and general processes of power load clusters. Then, they concluded several similarity measures and five main categories in power load clusters and their advantages and disadvantages. Data analysis found that the PGS-load-clustering method could help detect different PGS load patterns and provide theoretical support for smart grid research and applications.

It is also essential to replan the PGS power prediction model in intelligent PGSs. Si et al. [25] applied a hybrid model by studying Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks for short-term individual household load prediction. They used the CNN layer to extract features from input data and the LSTM layer for sequence learning. The research results showed that the intelligent PGS was transformed into a flexible and collaborative system. Additionally, residents' short-term PGS load forecasting would weigh heavier in future PGS operations and planning. Alhussein et al. [26] scrutinized forecasting models for renewable energy and electricity demand, reviewed high-quality research articles, and evaluated the model's forecast accuracy using five metrics. They summarized the forecasting objectives in power generation schedules, unit investment, and economic dispatching problems. The research offered practical references for critical and systematic research on renewable energy and power forecasting models. Ahmad et al. [27] delved into the short-term PGS load and price forecasting algorithms by introducing a new feature selection method based on entropy and mutual information. Then, candidate inputs were ranked to eliminate redundant input. The research results showed that the LSTM-based DL model could improve the model's predictive accuracy, significantly practical for intelligent development of PGS. Memarzadeh and Keynia [28] surveyed the PGS load prediction method based on DL and the Attention Mechanism (AM) by proposing an AM-based Bayesian optimized encoder–decoder network. The proposed model was used to forecast short-term PGS load against the existing model's poor stability and low prediction accuracy in processing time-series data. The experimental verification results indicated that the model was superior to other models regarding prediction accuracy and algorithm stability. The results provided an effective migration time series-oriented PGS load prediction method. Jin et al. [29] applied a Deep Belief Network (DBN) to predict short-term PGS loads. They also employed the Markov-based switching topology to address uncertain network attacks during neighbor communication. The findings showed that the load dataset was separated and trained the model locally without a central regulator. Meanwhile, it was updated through communication with random neighbors under the designed consensus program, significantly reducing the model training time.

To summarize, many PGS load forecasting-oriented research models have achieved high forecasting accuracy and excellent prediction effects. However, research is scarce regarding intelligent or digital PGS load forecasting Therefore, based on previous research, this paper designs a distributed PGS-oriented RMS using matching theory and MEC architecture to predict the intelligent PGS load accurately.

3. Distributed PGS-Oriented RMS by Matching Theory

3.1. RMS by Task Offloading

The development of MEC technology has challenged network resource management technology and faces many challenges [30–32]. MEC networks, mainly based on power resources and spectrum resources, place strict requirements on the edge device connection delay and rate. Moreover, mobile users often generate real-time, resource-intensive, complex, heterogeneous, and diversified service requests. Thus, resource management technologies in MEC networks are essential for improving User Experience (UX) and network resource utilization.

User-end mobile devices have minimal capacity to process resource-intensive tasks [33]. The intelligent PGS will offload those overloaded tasks to MEC devices to improve virtual resource utilization. Figure 1 displays RMS by task offloading.
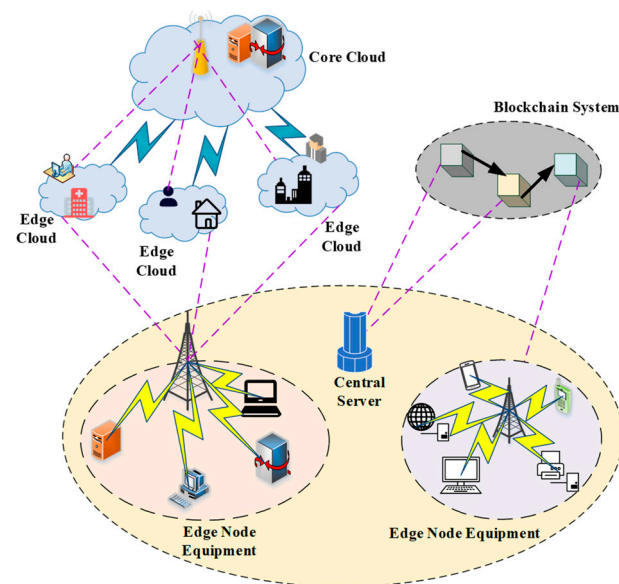
Figure 1. Structure of RMS by task offloading.

3.2. MEC Network Architecture and System Power Communication Model

Constructing a wireless communication model is essential to MEC network architecture. Specifically, all mobile devices run under a computing mode with different decision variables. Once enough mobile devices are connected to the network, user-end computing tasks can be offloaded through subchannels. The data transfer rate $r_{i,n}^m$ for the task is calculated according to (1).

$$r_{i,n}^m = W \log_2 \left(1 + \frac{p_{i,n} g_{i,n}^m}{\sigma^2 + K_{i,n}^m} \right) \tag{1}$$

In (1), W represents the channel bandwidth; $p_{i,n}$ denotes the information transmission power of the mobile device i; $g_{i,n}^m$ indicates the signal transmission gain between the mobile device i and the channel m; $K_{i,n}^m$ signifies the interference caused by other mobile devices to channel m; and σ^2 represents the Gaussian noise. In addition, computing local tasks using mobile devices must consider the local delay t_i^l and EC E_i^l of computing tasks. The local cost Z_i^l is defined by the local delay and EC, as given by (2).

$$Z_i^l = \lambda_1 t_i^l + \lambda_2 E_i^l \tag{2}$$

In (2), λ_1 and λ_2 denote the weight of the execution delay and EC of the mobile device i, respectively; and t_i^l and E_i^l represent the local delay and EC of the computing task, respectively. When a mobile device loads computing tasks on a remote server for

execution, the cost of edge computing mainly includes execution delay and EC. The total execution delay $t_{i,n}^{m,e}$ and total EC $E_{i,n}^{m,e}$ of mobile devices in the edge computing process can be expressed as:

$$t_{i,n}^{m,e} = t_{i,n}^{m,s} + t_{i,n}^{x} \tag{3}$$

$$E_{i,n}^{m,e} = E_{i,n}^{m,s} + E_{i,n}^{x} \tag{4}$$

where $t_{i,n}^{m,s}$ signifies the total delay of the data transmission process; $t_{i,n}^{x}$ represents the total delay of the system computing process; $E_{i,n}^{m,e}$ denotes the total EC of the system computing process; $E_{i,n}^{m,s}$ refers to the transmission energy consumption; and $E_{i,n}^{x}$ stands for the EC of the computing process. The sum of total delay and EC is the total cost Z_i^e of EC system, as written in (5).

$$Z_i^e = \lambda_1 \sum_n \sum_m a_{i,n}^m t_{i,n}^{m,e} + \lambda_2 \sum_n \sum_m a_{i,n}^m E_{i,n}^{m,e} \tag{5}$$

In (5), $\sum_n \sum_m a_{i,n}^m t_{i,n}^{m,e}$ represents the data transmission overhead; $\sum_n \sum_m a_{i,n}^m E_{i,n}^{m,e}$ refers to the system resource overhead; and λ_1 and λ_2 stand for the weight coefficients of the execution overhead in the two processes. Figure 2 shows the architecture of the MEC network.
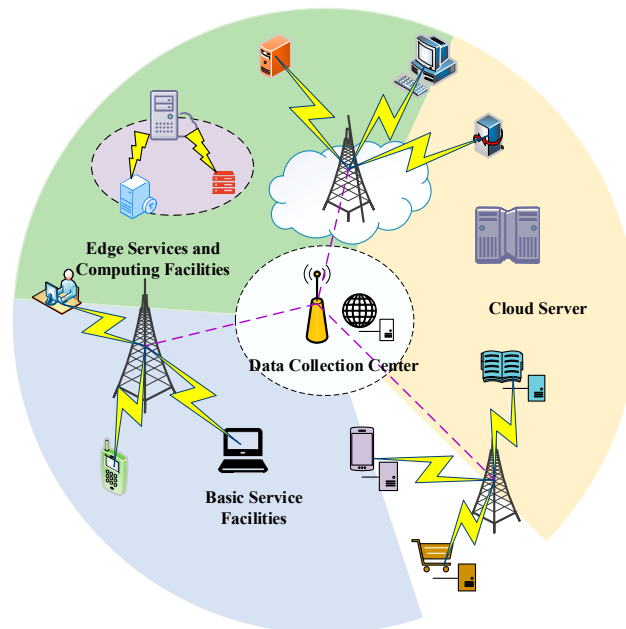

Figure 2. Schematic architecture of the MEC network.

In addition, decision variables are introduced in mobile devices to describe the system model. The cost minimization model is designed as (6).

$$\min_{a_i \in \{0,1\}} Z_i(a_i, a_{-i}), \forall i \in \mathbb{I} \tag{6}$$

Equation (7) describes the detailed objective function.

$$Z_i(a_i, a_{-i}) = \begin{cases} Z_i^l(1-a), & \text{if } a_i = 0 \\ Z_i^e(a), & \text{if } a_i = 1 \end{cases} \tag{7}$$

In (7), $Z_i(a_i, a_{-i})$ refers to the cost function, where a_i represents the decision variable value of the mobile device. Different load constraints need to be considered to solve the cost. The delay constraints and energy constraints of the server are expressed as:

$$Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i}) \tag{8}$$

$$P_i\left(a'_i, a_{-i}\right) < P_i\left(a_i, a_{-i}\right) \tag{9}$$

where $Z_i\left(a'_i, a_{-i}\right)$ indicates the actual system delay; $Z_i(a_i, a_{-i})$ denotes the estimated delay parameter value; $P_i\left(a'_i, a_{-i}\right)$ represents the actual system energy consumption; and $P_i(a_i, a_{-i})$ signifies the system energy consumption threshold. Furthermore, the threshold function $C^m_{i,n}$ for local computation offloading tasks is defined as (10).

$$C^m_{i,n} = \beta_1 \left(\frac{t^{m,e}_{i,n}}{t^l_i} \right) + \beta_2 \left(\frac{\sigma^2 + \sum_{l=1,l\neq n}^{N} \sum_{j=1,j\neq i}^{K} a^m_{j,l} p_{j,l} g^m_{j,n}}{p_{i,n} g^m_{i,n}} \right) \tag{10}$$

In (10), β_1 and β_2 are the weight of delay and interference, respectively; $\left(\frac{t^{m,e}_{i,n}}{t^l_i} \right)$ represent the influence factor of time delay on the system; $\left(\frac{\sigma^2 + \sum_{l=1,l\neq n}^{N} \sum_{j=1,j\neq i}^{K} a^m_{j,l} p_{j,l} g^m_{j,n}}{p_{i,n} g^m_{i,n}} \right)$ means the influence factor of data interference on the system. (11) indicates the threshold policy of the task offloading model.

$$P(a) = \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{I} C^m_{i,n} C^m_{j,l} I_{\{a_i = a_j : m\}} I_{\{a_i = 1\}} \tag{11}$$

In (11), $I_{\{a_i=1\}}$ represents the indicator function and $C^m_{i,n} C^m_{j,l} I_{\{a_i = a_j : m\}}$ refers to the task offloading process in the subchannel. In the edge computing system, the mobile device i needs to update its unloading decision in the EC system continuously. In the decision time slot τ, according to the interference and uplink data rate, the mobile device needs to update the server information according to (12).

$$\Theta_i(\tau) = \left\{ a'_i : Z_i\left(a'_i, a_{-i}\right) < Z_i(a_i, a_{-i}) \right\} \tag{12}$$

In (12), $\Theta_i(\tau)$ represents the updated parameter information and $Z_i\left(a'_i, a_{-i}\right)$ signifies the cost decision set in the unloading process. In the process of updating the decision, the mobile device needs to compete for computing resources to minimize the resource overhead Then, the competition for resources reaches equilibrium after finite iterations, which can be described as:

$$a^*_i = \arg \min_{a^*_i} Z_i\left(a'_i, a_{-i}\right), a^*_i \in \Theta_i(\tau) \tag{13}$$

where $\Theta_i(\tau)$ represents the set of optimal response decisions and $\Theta_i(\tau) = 0$ demonstrates that the algorithm is implemented within finite iterations. Then, after G iterations, the difference between the actual and the predicted decisions follows (14).

$$0 < \beta_1 \Delta \kappa_1 + \beta_2 \Delta \kappa_2 \leq 2 \tag{14}$$

In (14), $\Delta \kappa_1$ and $\Delta \kappa_2$ refer to the difference between the actual and the predicted decisions, and β_1 and β_2 represent the variation coefficient of the two thresholds, respectively. Suppose that the equilibrium state policy set of all mobile devices is a^*_i. Then, the approximate Efficiency Ratio *ER* of the optimal solution can be written as (15).

$$ER = \frac{\min \sum_{i=1}^{I} Z_i\left(a^*_i\right)}{\min \sum_{i=1}^{I} P_i\left(a^*_i\right)} \tag{15}$$

The closer the value of *ER* is to 1, the better the state of the mobile device; a^*_i in (15) stands for the optimal solution using minimal overhead. The proposed intelligent PGS-oriented delay cost model releases independent delay-sensitive tasks with an iden-

tical distribution. Meanwhile, the computing task *i* is offloaded through the self-carrier. Equation (16) describes the link rate of data transmission.

$$r_{i,k,m}(t) = \alpha_k \log_2 \left(1 + \frac{p_i g_{i,m}(t)}{I_{i,k,m}(t) + \sigma^2} \right) \tag{16}$$

In (16), α_k represents the bandwidth of the subcarrier at time t; p_i stands for the bandwidth allocation vector; $g_{i,m}(t)$ denotes the information transfer gain between the user and the channel; $I_{i,k,m}(t)$ refers to the unified channel unloaded resources by the remaining base stations; and σ^2 signifies the signal interference value. Figure 3 displays the PGS communication model based on unified channel resource scheduling by the base station.
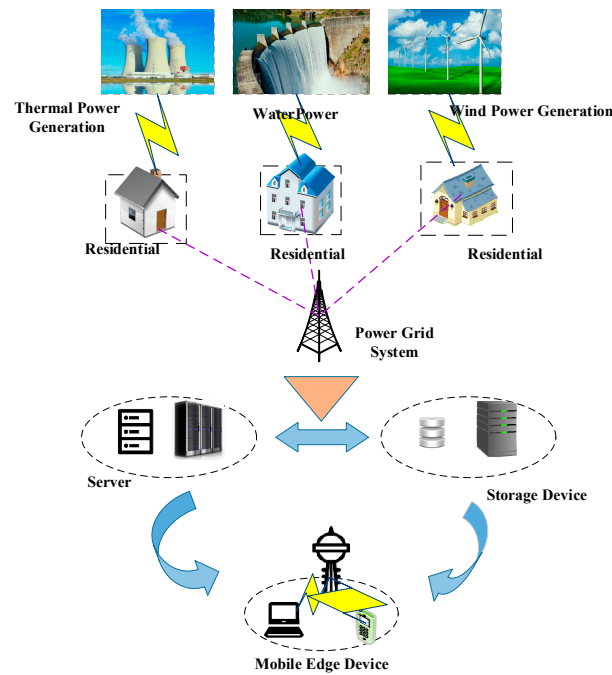

Figure 3. PGS communication model.

3.3. Online Delay-Awareness Power RAA

Reasonable computing resources allocation can improve the system's computing power. However, improvement of computing power has certain limitations, as shown in (17).

$$\sum_{k \in \underline{K}} a_{i,k,m}(t) r_{i,k,m}(t) \leq \theta_m(t-1) \frac{F_m^{max}}{\eta^m} \tag{17}$$

In (17), $a_{i,k,m}(t)$ denotes the bandwidth resource of the signal base station; $r_{i,k,m}(t)$ refers to the signal gain power of user i between channels; θ_m represents the path loss parameter; and $\frac{F_m^{max}}{\eta^m}$ signifies the task offloading decision parameter. The total delay cost T_i and average delay cost $\overline{T}$ of the user performing the computing task can be written as:

$$T_i = \sum_{t=1}^{T} E_{\{L_i(t) \neq 0\}} \tag{18}$$

$$\overline{T} = \lim_{T \to \infty} \frac{1}{I} \sum_{i=1}^{I} \sum_{t=1}^{T} T_i \tag{19}$$

where $E_{\{L_i(t) \neq 0\}}$ measures whether or not the event occurs. If the event occurs, $E_{\{L_i(t) \neq 0\}} = 1$; otherwise, the parameter value $E_{\{L_i(t) \neq 0\}} = 0$. According to different offloading strategy

functions, tasks are injected into different classification sets. The three different offloading strategy functions can be expressed as:

$$Q_i^l(t) = \frac{L_i(t)C_i}{f_i} \tag{20}$$

$$Q_i^m(t) = \frac{L_i(t)\eta^m}{f_i} \tag{21}$$

$$Q_i^n(t) = \frac{L_i(t)\eta^n}{f_i} \tag{22}$$

where $Q_i^l(t)$, $Q_i^m(t)$, and $Q_i^n(t)$ represent the user online classification strategy, the allocation strategy based on the macro base station, and the allocation strategy based on the small base station, respectively. $L_i(t)$ stands for the task uploading resource consumption to the remote server. C_i, η^m, and η^n denote the task unloading efficiency of different unloading strategies, and f_i represents the task execution rate. Then, the delay cost on multiple tasks is evaluated by the competition rate $o(ORAM)$, as calculated by (23).

$$o(ORAM) = \max_{\underline{I}} \left\{ \frac{e(ORAM(\underline{I}))}{e(OHA(\underline{I}))} \right\} \tag{23}$$

In (23), $e(ORAM(\underline{I}))$ represents the delay cost of executing the task event through the system mechanism and $e(OHA(\underline{I}))$ denotes the delay cost of the offline allocation algorithm to complete the task. In addition, the offline objective and the penalty function need to jointly form a fitness function to solve the optimization problem. The offline fitness function is defined as (24).

$$\text{Fitness} = \overline{T} + \text{penalty}\,(X) \tag{24}$$

In (24), $\overline{T}$ represents the offline objective function and (X) stands for the offline penalty function. The blockchain model calculates the probability that mobile devices conduct transactions at network edge nodes according to (25).

$$P_m^{mine} = \frac{w_m}{\sum_{m \in M} w_m} \tag{25}$$

In (25), P_m^{mine} represents the probability of mining a block during the consensus process of the MEC network, and w_m denotes the computing power of the edge node m. Then, a block in edge nodes is successfully loaded into the blockchain, and the edge node mining the block will be rewarded. Figure 4 illustrates the framework of the intelligent PGS-oriented PPS.

In addition, the mining rewards R_m and mining accuracy P_m^{win} are calculated by:

$$R_m = \left(R + \sum_n x_{nm} \rho_n \frac{w_m}{\sum_{m \in M} w_m} e^{-\lambda \alpha_m} \right) \tag{26}$$

$$P_m^{win} = P_m^{mine} \left(1 - P_m^{orphan} \right) \tag{27}$$

where P_m^{win} denotes the mining accuracy, P_m^{mine} represents the overall mining rate, and P_m^{orphan} refers to mining inaccuracy (28) counts the mining inaccuracy.

$$P_m^{orphan} = 1 - e^{-\lambda \alpha_m} \tag{28}$$

In (28), λ denotes the Poisson distribution parameter; m is the edge node; and R_m stands for the mining reward. In addition, the calculation cost CC_m of the edge node m can be presented as (29).

$$CC_m = PP_m + MP_m \tag{29}$$

In (29), PP_m represents the power of the edge node processor and MP_m signifies the memory power. Given the system resource utilization, the computing resource scheduling is set as (30).

$$\sum_{m \in N} x_{nm} d_n \leq v_m \tag{30}$$

In (30), d_n represents the system Central Processing Unit (CPU) cycle and x_{nm} denotes the computing resources of edge nodes. Equation (31) demonstrates the service cost of edge nodes.

$$C_m = \zeta * TC_m + \vartheta * CC_m \tag{31}$$

In (31), $\zeta * TC_m$ represents the system data transmission cost and $\vartheta * CC_m$ denotes the system computing cost. The preference file $\phi_n(m)$ of the edge node of the matching system with unilateral preference is determined according to (32).

$$\phi_n(m) = \left\{ \rho_n \frac{w_m}{\sum_{m \in M} w_m} e^{-\lambda f s_n} \right\} - \left\{ \frac{\zeta}{B_n} + \vartheta f s_n \right\} \tag{32}$$
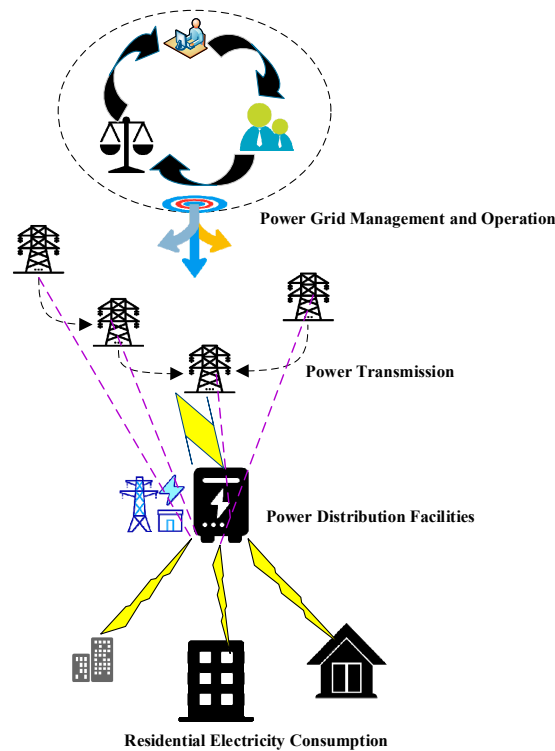
Figure 4. The architecture of the intelligent PGS-oriented PPS.

In (32), B_n represents the bandwidth allocated by edge node m to transaction n; $\phi_n(m)$ denotes the preference coefficient of edge nodes; ζ refers to the network resource utilization; ϑ stands for the data transmission efficiency; f represents the transmission resistance coefficient; and s_n indicates the transmission benefit cost. Figure 5 provides the structure of the online power RAA.

The data collection equipment collects the PGS loads systematically to analyze and compare the performance of the EC-based intelligent PGS load forecasting system. Algorithm 1 presents online delay-aware RAA.
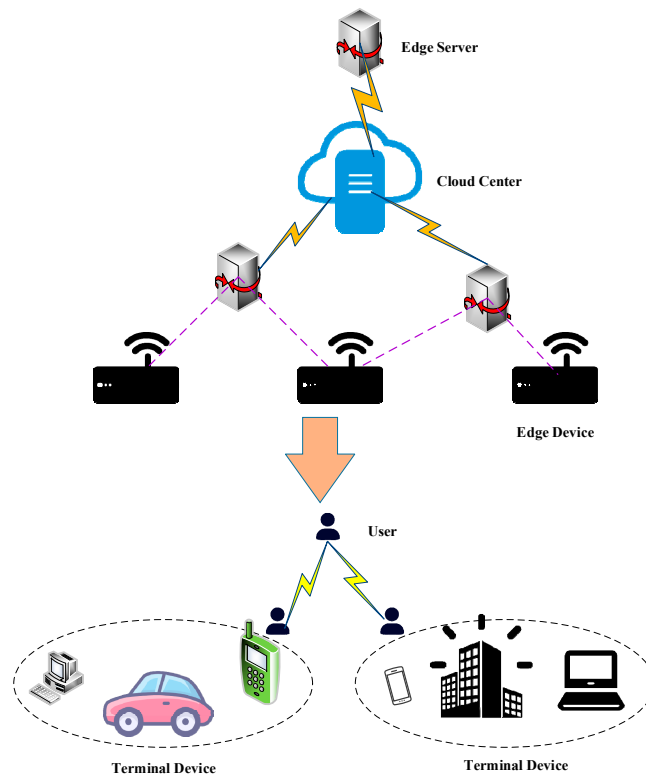
Figure 5. Structure of online power RAA.

Algorithm 1: Distributed matching algorithm.

1	Input : $\pi = (N, M, \phi_N(M)), \gamma_k \sigma_{mv} \zeta_m$
2	Output : μ
3	Initialization: $flag(m) \leftarrow Failure, \forall m \in M$
4	Repeat
5	While Transaction was not successfully matched do
6	The maximum number of transactions randomly scheduled by the main edge node D_n
7	If $s_n \geq \phi_m$ or $\rho_n \geq \zeta_m$ then
8	update ζ_m;
9	$flag(m) \leftarrow Success$;
10	End if
11	If $\phi_n(m) > \phi'_n(m)n, n' \in \zeta_m$ then
12	If $|\sum_n x_{nm} s_n| \leq \zeta_m \gamma_m$ or, $|\sum_n x_{nm} d_n| \leq v_m$ then
13	$\zeta_m = \zeta_m / n'$;
14	End if
15	End if
16	End while
17	Until $\phi_n(m) = \varnothing, \forall m \in M, n \in N$;
18	If Each edge node is successfully matched then
19	output μ
20	Else
21	Return "Failed"
22	End if
23	Output: μ is a stable match.

3.4. Experimental Parameter Settings

Simulation experiments verify the effect of the proposed intelligent PGS load forecasting model, and the mechanical properties are theoretically analyzed. The experimental data used for the simulations are from Ampds, a public dataset for load decomposition and ecological feedback studies. This dataset contains detailed energy consumption data

for 19 household internal grid devices during a year. This experiment assumes that the resource release time A of the PGS is uniformly distributed between [1,T], and the released resources change continuously over time. At the same time, the bandwidth between the user and the MEC system is set to 9×10^6 MHz. The frequency of the local CPU obeys a uniform distribution in the range [0.65, 0.75] GHz. The actual signal transmission power of the system is 5 W. Additionally, the proposed delay-aware online power RAA is compared with the Remote Upload Algorithm (RUA) and the Local Computing Algorithm (LCA), factoring in system overhead and system delay. The experimental data of each group of algorithms are divided into the experimental group (Group 1) and control group (Group 2) to reduce the experimental error. The six groups are RUA-1, RUA-2, LCA-1, LCA-2, OAP-1, and OAP-2. The statistical results are analyzed in the next section.

4. Results and Discussion

4.1. An RMS by Task Offloading

The proposed online delay-aware power RAA is compared with RUA and LCA on overhead and system delay. As shown in Figure 6, the system overhead decreases gradually with iteration. Figure 7 reveals the impact of mobile device numbers and data volume on the system overhead. Figure 8 indicates the influence of the mobile device number on the system delay. The proposed online delay-aware power RAA is denoted as Optimization Algorithm Proposed (OAP) in the following graphs.
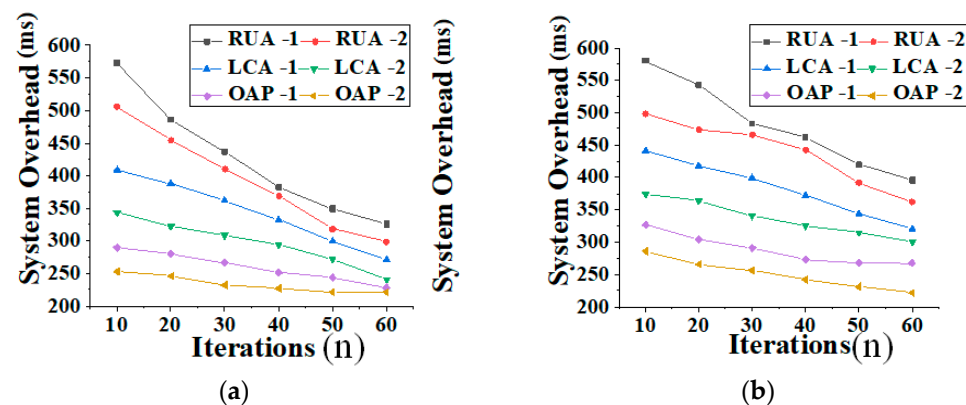

Figure 6. Variation curves of the system overhead with iteration: (**a**) variation curve under threshold = 0.1; (**b**) variation curve under threshold = 0.5.
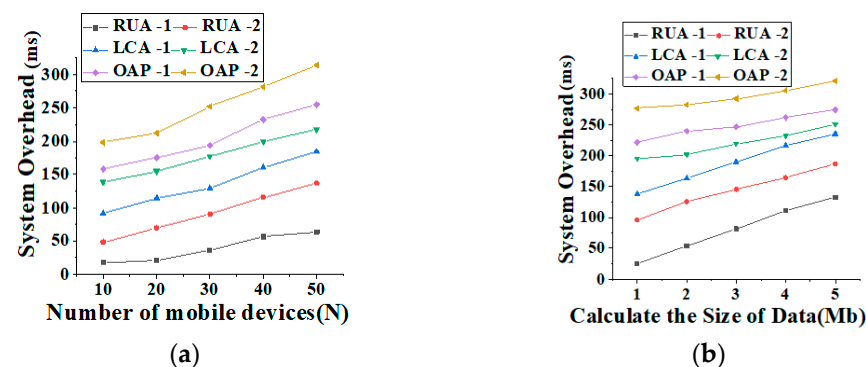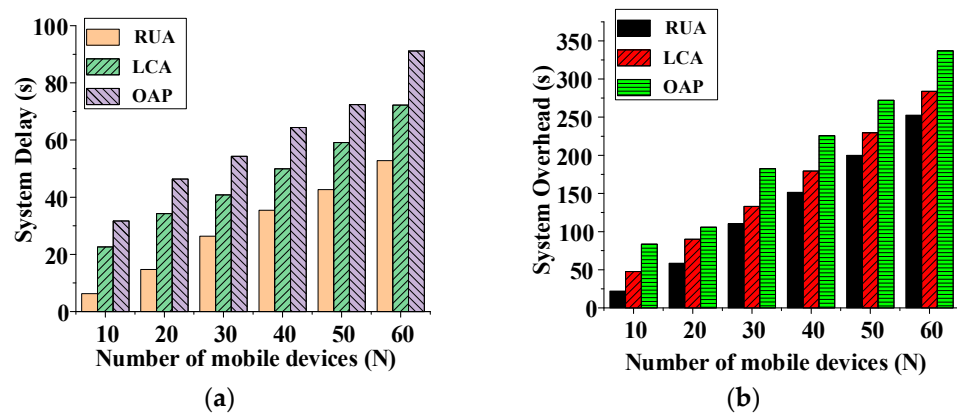

Figure 7. Variation curves of the influence of the mobile device number and the data volume on the system overhead: (**a**) mobile device number on the system overhead; (**b**) data volume on system overhead.

Figure 8. Influence of the mobile device number on the system delay and system overhead: (**a**) the changing trend of the system delay with the increase in mobile devices; (**b**) the changing trend of the system overhead with the increase in mobile devices.

Figure 6 shows that the system overhead of the three algorithms decreases significantly with model iterations. From the 40th iteration, the overall system overhead stabilizes. Apparently, the threshold also impacts the system overhead: the larger the threshold, the greater the system overhead. Meanwhile, the system overhead of the three algorithms also tends to be stable from the 40th iteration. Comparing the system overhead under different thresholds reveals that the system overhead increases significantly, as shown from Figure 6a to Figure 6b. Thus, the model parameters have a specific impact on the system overhead. Figure 8 presents the variation curves of the influence of the mobile device number and the data volume on the system overhead.

Evidently, the mobile device number is positively proportional to system overhead; the system overhead increases gradually with the increase in mobile devices. Then, when mobile devices are 40 N, the system overhead begins to stabilize. Remarkably, OAP has the largest system overhead, mainly used to allocate PGS resources. Meanwhile, computing data volume is also positively proportional to the system overhead; the system overhead increases with the computing data volume. However, when data volume = 4 MB, the system overhead stabilizes. RUA takes up the lowest proportion of system overhead among the three algorithms. Hence, its data calculation is insufficient, and the results are less reliable. Therefore, the system model with a calculation dataset that is too large can be optimized to improve the utilization of the algorithm.

Apparently, with the growth in mobile device numbers, the system delay and system overhead show a significant increasing trend. The system delay and overhead of RUA are the smallest among the three algorithms. In contrast, OAP presents the largest proportion of system delay and system overhead. This result proves that OAP needs larger system overhead to allocate power resources with the increase in the mobile device number. Although the accuracy of the delay-aware power RAA is improved, prolonged system delay will affect the data transmission efficiency, which is worthy of improvement.

### 4.2. Changing Curves of Average Time Delay and System Fixed Control Parameters

Further, user numbers and system running time are factored into the experiment to evaluate the three algorithms' variation range of system fixed parameters. The longer the system runs, the greater the requirements for the algorithm, the longer time the algorithm needs, and the higher the fitness. Therefore, it is essential to evaluate the algorithm efficiency through running time. Figure 9 plots the impact of the user number and system running time on the average delay of users. Figure 10 gives the parameter changes of system calculation resource surplus and average system delay. Figure 11 describes how the trend of fixed control parameters and system average delay change with the increase in users.
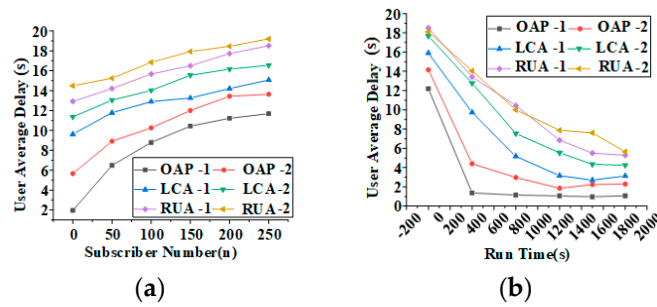
Figure 9. Influence of the number of users and the system running time on the average user delay (**a**) curves of the average user delay with user number; (**b**) curve of the average user delay with the system running time.
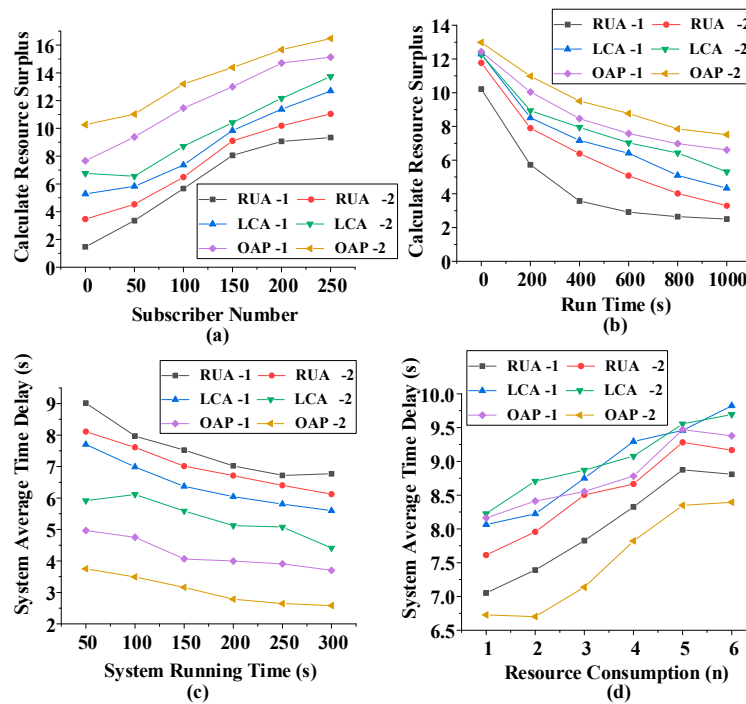
Figure 10. Parameter changing curves of the resource remaining and the average system delay: (**a**) the relationship between the user number and resource remaining; (**b**) the relationship between the system running time and resource remaining; (**c**) variation curves of average system delay with the increase in system running time; (**d**) variation curves of average system delay with resource consumption rate.
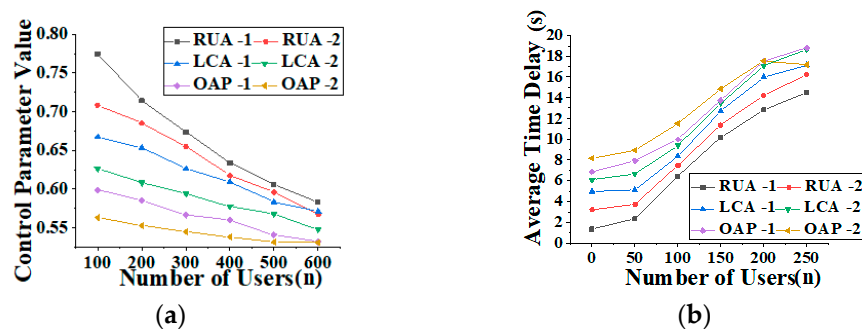
Figure 11. Variation trends of the system fixed control parameters and average delay with the increase in users: (**a**) changing curves of the system fixed control parameters with the increase in users; (**b**) changing curves of the average system delay with the increase in users.

Obviously, with the increase in users, the average user delay increases significantly. Among the three algorithms, the OAP has the smallest average user delay, less than 13 s, when the number of users is less than 50. Furthermore, the average user delay shows a downward trend with the increase in the system running time. Noticeably, the trend of RAU and LCA is not mild. In contrast, OAP decreased sharply. Therefore, when the system calculation time exceeds a threshold, the OAP delay greatly reduces and stabilizes at about 4 s when the system running time reaches 400 s. Thus, the operation efficiency of the OAP system is obviously better than the other two algorithms.

According to Figure 10, the number of users is positively proportional to the resource remaining. The system running time is inversely proportional to the resource remaining. With the increase in users, more system resources are generated. OAP performs best in calculating the resource remaining. By comparison, the average system delay decreases slightly with the system running time. With the increase in resource consumption rate, the average system delay also increases. Moreover, as shown in Figure 10d, the gap between OAP-1 and OAP-2 is large, indicating that the system stability is not high. However, the average delay between the two systems is still very low. Thus, the performance of the OAP algorithm is still outstanding, and the average delay of OAP is about 7.5 s. Compared with the other two algorithms, OAP significantly improves the efficiency of system resource allocation.

According to Figure 11, with the increase in the user numbers, the system fixed control parameter shows a downward trend, while the average system delay of several algorithm models increases. When the user number reaches 100, the fixed control parameter of OAP is about 0.75, while that of RUA is only 0.6; the higher the fixed-parameter, the better the data processing effect. Therefore, OAP is superior to other algorithms in data processing. Meanwhile, the average system delay of OAP is higher than that of the other two algorithms at the beginning but decreases significantly when the number of users increases to a threshold. Moreover, the OAP curve is lower than the other two models. Therefore, the performance of OAP in system delay is better than the other two algorithms, significantly reducing system resource consumption.

5. Conclusions

This paper proposes a distributed PGS-oriented RMS based on matching theory and an online delay-aware power RAA. Initially, it discusses the algorithm theory, then designs the algorithm model, and finally evaluates the algorithm performance by comparing three different algorithms. The results show that the system overhead of the three algorithms decreases significantly with the model iteration. From the 40th iteration, the system overhead stabilizes. Meanwhile, the average user delay of OAP is the smallest, less than 13 s, given no more than 50 users. Thus, OAP outperforms the other two algorithms in the average user delay. This research has practical value for the digitization and intellectualization of PGS. However, some defects were noted. First, given the limited communication and computing resources, the delay and resource consumption are simplified by calculating system overhead through the overhead minimization model. Second, this research focuses on maximizing the system benefits of the MEC network while ignoring the impact of blockchain technology on network resource allocation. Future research will combine the online RAA with the blockchain network, build a new network RMS, and improve the utilization efficiency of the system on the edge computing resources.

Author Contributions: Conceptualization, X.P.; methodology, X.P.; software, X.P.; investigation, all authors; writing—original draft preparation, X.P.; writing—review and editing, X.P., X.L., H.D. and J.M.G. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ding, N.; Liu, J.; Kong, Z.; Yan, L.; Yang, J. Life cycle greenhouse gas emissions of Chinese urban household consumption based on process life cycle assessment: Exploring the critical influencing factors. *J. Clean. Prod.* **2019**, *210*, 898–906. [CrossRef]
2. Akbari Vakilabadi, M.; Afzalabadi, A.; Khoeini Poorfar, A.; Rahbari, A.; Bidi, M.; Ahmadi, M.H.; Ming, T. Technical and economical evaluation of grid-connected renewable power generation system for a residential urban area. *Int. J. Low-Carbon Technol.* **2019**, *14*, 10–22. [CrossRef]
3. Sun, K.; Li, K.; Zhang, Z.; Liang, Y.; Liu, Z.; Lee, W. An Integration Scheme of Renewable Energies, Hydrogen Plant, and Logistics Center in the Suburban Power Grid. *IEEE Trans. Ind. Appl.* **2022**, *58*, 2771–2779. [CrossRef]
4. 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]
5. Kumar, P.M.; Gandhi, U.D. A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases. *Comput. Electr. Eng.* **2018**, *65*, 222–235. [CrossRef]
6. Iqbal, J.; Khan, M.; Talha, M.; Farman, H.; Jan, B.; Muhammad, A.; Khattak, H.A. A generic internet of things architecture for controlling electrical energy consumption in smart homes. *Sustain. Cities Soc.* **2018**, *43*, 443–450. [CrossRef]
7. Lasmar, C.J.; Bishop, T.R.; Parr, C.L.; Queiroz, A.C.M.; Schmidt, F.A.; Ribas, C.R. Geographical variation in ant foraging activity and resource use is driven by climate and net primary productivity. *J. Biogeogr.* **2021**, *48*, 1448–1459. [CrossRef]
8. Azzolin, A.; Dueñas-Osorio, L.; Cadini, F.; Zio, E. Electrical and topological drivers of the cascading failure dynamics in power transmission networks. *Reliab. Eng. Syst. Saf.* **2018**, *175*, 196–206. [CrossRef]
9. Rivera, J.; Nasirifard, P.; Leimhofer, J.; Jacobsen, H.A. Automatic generation of real power transmission grid models from crowdsourced data. *IEEE Trans. Smart Grid* **2019**, *10*, 5436–5448. [CrossRef]
10. Gazafroudi, A.S.; Neumann, F.; Brown, T. Topology-based approximations for $N-1$ contingency constraints in power transmission networks. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107702. [CrossRef]
11. Wang, X.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]
12. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [CrossRef]
13. Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. A Survey on Mobile Augmented Reality with 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1160–1192. [CrossRef]
14. Ranaweera, P.; Jurcut, A.D.; Liyanage, M. Survey on Multi-Access Edge Computing Security and Privacy. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1078–1124. [CrossRef]
15. HaddadPajouh, H.; Khayami, R.; Dehghantanha, A.; Choo, K.K.R.; Parizi, R.M. AI4SAFE-IoT: An AI-powered secure architecture for edge layer of Internet of things. *Neural Comput. Appl.* **2020**, *32*, 16119–16133. [CrossRef]
16. Jha, D.N.; Alwasel, K.; Alshoshan, A.; Huang, X.; Naha, R.K.; Battula, S.K.; Ranjan, R. IoTSim—Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments. *Softw. Pract. Exp.* **2020**, *50*, 844–867. [CrossRef]
17. Lv, Z.; Liang, Q. Optimization of collaborative resource allocation for mobile edge computing. *Comput. Commun.* **2020**, *161*, 19–27. [CrossRef]
18. Lv, Z.; Chen, D.; Lou, R.; Wang, Q. Intelligent edge computing based on machine learning for smart city. *Future Gener. Comput. Syst.* **2021**, *115*, 90–99. [CrossRef]
19. Firouzi, F.; Farahani, B.; Marinšek, A. The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT). *Inf. Syst.* **2022**, *107*, 101840. [CrossRef]
20. Zhang, T.; Gao, L.; He, C.; Zhang, M.; Krishnamachari, B.; Avestimehr, S. Federated Learning for Internet of Things: Applications, Challenges, and Opportunities. *arXiv* **2021**, arXiv:2111.07494.
21. Hafeez, G.; Alimgeer, K.S.; Khan, I. Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. *Appl. Energy* **2020**, *269*, 114915. [CrossRef]
22. Khalid, R.; Javaid, N.; Almogren, A.; Javed, M.U.; Javaid, S.; Zuair, M. A Blockchain-Based Load Balancing in Decentralized Hybrid P2P Energy Trading Market in Smart Grid. *IEEE Access* **2020**, *8*, 47047–47062. [CrossRef]
23. Xia, M.; Shao, H.; Ma, X.; de Silva, C.W. A Stacked GRU-RNN-Based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7050–7059. [CrossRef]
24. Si, C.; Xu, S.; Wan, C.; Chen, D.; Cui, W.; Zhao, J. Electric load clustering in smart grid: Methodologies, applications, and future trends. *J. Mod. Power Syst. Clean Energy* **2021**, *9*, 237–252. [CrossRef]
25. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [CrossRef]
26. Ahmad, T.; Zhang, H.; Yan, B. A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. *Sustain. Cities Soc.* **2020**, *55*, 106995. [CrossRef]

27. Memarzadeh, G.; Keynia, F. Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm. *Electr. Power Syst. Res.* **2021**, *192*, 106995. [CrossRef]

28. Jin, X.; Zheng, W.Z.; Kong, J.; Wang, X.; Bai, Y.; Su, T.; Lin, S. Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization. *Energies* **2021**, *14*, 1596. [CrossRef]

29. Dong, Y.; Dong, Z.; Zhao, T.; Li, Z.; Ding, Z. Short term load forecasting with markovian switching distributed deep belief networks. *Int. J. Electr. Power Energy Syst.* **2021**, *130*, 106942. [CrossRef]

30. Lu, H.; Zhang, Y.; Li, Y.; Jiang, C.; Abbas, H. User-Oriented Virtual Mobile Network Resource Management for Vehicle Communications. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3521–3532. [CrossRef]

31. Liu, C.; Zheng, X. Exploring resource management for innovation power network based on deep learning algorithm. *Neural Comput. Appl.* **2021**, *33*, 4013–4025. [CrossRef]

32. Xu, H.; Klaine, P.V.; Onireti, O.; Cao, B.; Imran, M.; Zhang, L. Blockchain-enabled resource management and sharing for 6G communications. *Digit. Commun. Netw.* **2020**, *6*, 261–269. [CrossRef]

33. Yang, H.; Cai, W.; Xia, Y.; Ouyang, W.; Xie, X. Identity authentication system for mobile terminal equipment based on SDN network. *Int. J. Inf. Commun. Technol.* **2020**, *17*, 257–273. [CrossRef]