

## Article

# Environment Optimization Scheme Based on Edge Computing Using PSO for Efficient Thermal Comfort Control in Resident Space

Rongxu Xu <sup>1</sup>, Wenquan Jin <sup>2</sup> and Dohyeun Kim <sup>3,\*</sup><sup>1</sup> Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; rongxu@jejunu.ac.kr<sup>2</sup> Big Data Research Center, Jeju National University, Jeju 63243, Korea; wenquan.jin@jejunu.ac.kr<sup>3</sup> Department of Computer Engineering, Advanced Technology Research Institute, Jeju National University, Jeju 63243, Korea

\* Correspondence: kimdh@jejunu.ac.kr

**Abstract:** With the fast development of infrastructure and communication technology, the Internet of Things (IoT) has become a promising field. Ongoing research is looking at the smart home environment as the most promising sector that adopts IoT and cloud computing to improve resident live experiences. The IoT and cloud-dependent smart home services related to recent researches have security, bandwidth issues, and a lack of concerning thermal comfort of residents. In this paper, we propose an environment optimization scheme based on edge computing using Particle Swarm Optimization (PSO) for efficient thermal comfort control in resident space to overcome the aforementioned limitations of researches on smart homes. The comfort level of a resident in a smart home is evaluated by Predicted Mean Vote (PMV) that represents the thermal response of occupants. The PSO algorithm combined with PMV to improve the accuracy of the optimization results for efficient thermal comfort control in a smart home environment. We integrate IoT with edge computing to upgrade the capabilities of IoT nodes in computing power, storage space, and reliable connectivity. We use EdgeX as an edge computing platform to develop a thermal comfort considering PMV-based optimization engine with a PSO algorithm to generate the resident's friendly environment parameters and rules engine to detect the environmental change of the smart home in real-time to maintain the indoor environment thermal comfortable. For evaluating our proposed system that maintains resident environment with thermal comfort index based on PSO optimization scheme in smart homes, we conduct the comparison between the real data with optimized data, and measure the execution times of optimization function. From the experimental results, when our proposed system is applied, it satisfies thermal comfort and consumes energy more stably.



**Citation:** Xu, R.; Jin, W.; Kim, D. Environment Optimization Scheme Based on Edge Computing Using PSO for Efficient Thermal Comfort Control in Resident Space. *Actuators* **2021**, *10*, 241. <https://doi.org/10.3390/act10090241>

Academic Editor: Paolo Mercorelli

Received: 30 June 2021

Accepted: 9 September 2021

Published: 17 September 2021

**Keywords:** edge computing; EdgeX; microservice; optimization; smart home; thermal comfort

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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

With the fast development of infrastructure and information technology, the IoT has become a promising field in the domain of global information and communication technology. Several industries have integrated the concept of the IoT into their traditional products and services to facilitate their use by consumers. In addition, the practice and scientific fields of the ICT domain have also changed. Previous studies have shown that the IoT has significantly influenced the overall structure, business philosophy, and future direction of the industry [1]. Current research is also looking at the smart home environment, one of the most promising environments for the adoption and employing of the IoT [2].

A smart home is one of the applications of the IoT, which is the connectivity of physical objects that allows a digital device, sensor, application, and network communication inside a home [3]. Many definitions have been used to design and define smart homes. Definitions

vary depending on the point of view. The author [4] describes a smart home as a user armed with a technology of computing and information, which inference and reacts to the requirement of the residents, trying to improve their living experiences such as convenience, comfort, and security by the control system within the home environment and connections to the world beyond. The author [5] introduces a smart home as a combination of every service within a home through adopting a common internet protocol. The system provides an intelligent ability and flexibility to assures the users with a secure, economic, and comfortable life at home. The author [6] explain smart homes are a building that armed with sensors and actuators. The control devices could be ventilation, lighting, heating, air conditioning. There is a control system called “gateways”, which provides communication ability to connect switches and sensors. The control system develops a user interface that allows consumers to interact with the system through a smartphone, a tablet, or a laptop.

With the advancement of technology, more and more people are working at home and wherever they feel comfortable place [7]. Since the mood of the residents and productivity of the employees are influenced by the quality of the place where they working and living, thermal comfort management is an important task [8]. There are studies to implement a smart home control system using the IoT and microcontrollers to manage the attributes of the environment. It detects and controls the home environment through IoT devices, and monitors manage environmental changes through mobile devices or web pages. The study [9] develops remote and microcontrollers with Android mobile phones to deploy a control system on Arduino Uno and provide connectivity through Wifi to create a smart home control system. The system provides capabilities such as turning lights on and off, opening and closing the door of the home, turning faucets on and off, and monitoring the condition of the house via video surveillance. The test results of home prototypes with Android mobile phones show us that the manage distance of household electronic devices is up to 150 m, and media can directly monitor the condition of the house through CCTV at a distance of monitor up to 100 m. Research [10] is working to make their homes environmentally friendly. The system enables users to manage energy usage and decrease cost by controlling light, watering, and usage monitoring. Mobility and smartphone technology provides users the ability to manage their home electronic devices through smartphones. The remote devices control enables users to complete manages before going home. The proposed home control system prototype allows users to remotely turn on or off any device of the IoT via an improved solar charger. However, the IoT is typically characterized by widely distributed devices with limited processing capabilities and storage. These IoT devices encounter issues regarding performance, reliability, privacy, and security [11]. In addition, it lacks an automation function that monitors the indoor environment of a smart home in real-time to maintain an optimized environment for the occupant.

There is a paradigm that encompasses a huge network with unlimited repository capacity and computing power, which is called cloud computing. In addition, it provides flexibility to the data managed by cloud computing through dynamic data combinations from different resources of data [12]. The IoT has challenges with reliability, security, privacy, and performance that are raised by limited possibilities such as computing power and storage. The combination of IoT with cloud computing is a solution to deal with most of the issues [13]. In addition, the cloud is user-friendly and reduces costs for end-users who use applications and services. With the cloud, the process of IoT data collection and processing is simplified, and it provides fast and low-cost setup and integration for comprehensive data treating and provisioning [14]. The research [15] suggests using several IoT devices with some digital assistants provided by Amazon, Google, Apple, or Microsoft to receive voice commands of people who have physical disabilities to manage the smart home devices. The research [16] introduces a small house IoT convection device with the implemented system which provides remote control. To provide a comfortable environment to the resident, the control module collects data from sensors and residents to calculate optimized environment parameters, and adopt the results to the IoT convection

system. Cloud computing is used to store all the data collected from IoT sensors and produce optimal parameters for thermal comfort inside the home. The system enables users to adjust the temperature of a smart home not only from a smartphone or web page but also from the local home networking system. However, security is a problem in the integration of IoT and cloud computing. Since the IoT devices and cloud-connected by the internet, there are many possibilities of attacks on both the IoT devices and cloud side [17]. In the context of IoT, encryption can be used to maintain data confidentiality and integrity. However, with the limited capabilities, insider attacks are hard to resolve by the IoT [18]. With constrained bandwidth for transfer data, cloud computing with centralized architectures has a problem in computing and analyzing the huge data gathered from IoT sensors [19].

In this paper, we propose a thermal comfort considering optimization scheme based on edge computing using the PSO algorithm for efficient thermal comfort control in a smart home environment. We integrate IoT with edge computing to extend the resources of the IoT devices. Edge computing assists computing and storing resources with reliable connectivity close to the IoT devices. We use EdgeX as an edge computing platform to develop a PMV-based optimization engine with a PSO algorithm to generate the resident's friendly environment parameter and rules engine to detects the environmental change of the smart home in real-time to maintain the indoor environment thermal comfortable.

The following is our contributions:

We developed an autonomous thermal comfort maintenance system through the integration of the PSO algorithm and the PMV model for a smart home.

We overcome the shortcoming of IoT devices by using edge computing.

The rest of this paper is structured as follows. In Section 2, we explored research related to edge computing and research related to optimization. In Section 3, the design of the smart home space and environment optimization scheme for efficient thermal comfort control is presented. In Section 4, introduces the development environment and experimental results. The evaluation results are presented in Section 5. Concludes our work in Section 6.

## 2. Related Works

Smart homes communicate with a lot of electronic home appliances based on IoT to support residents effectively. All the IoT devices in smart homes would interoperate with one another in the ideal future. The IoT-based smart home paradigm has significantly influenced the quality of human life by enables communications to everybody regardless of space and time [4]. The connected smart home appliances support an opportunity for the occupant of a smart home to effectively control their energy consumption while improving experience and comfort [20]. The automated control system has the possibility to enhance the experience of life and facilitate independent living [21]. Users can benefit from maintaining a convenient living environment or reducing energy consumption through a smart home, but at the same time, there are shortcomings. Various standards of IoT devices and their dependence on the cloud affect the performance of smart homes [22].

Edge computing is a promising technology that provides sufficient connectivity and stable computing power in terms of smart home applications [23]. The advent of the edge computing paradigm is the achievement of low latency, a bandwidth-efficient goal that can not only implement processes but also process large amounts of data at the network edge [24]. Edge computing supports compute and storage resources by providing reliable connections near IoT devices. Edge computing has several advantages over cloud computing. First, edge computing supports lower connection costs, by reducing data transmission and storing data locally [25]. Second, edge computing provides better security because less sensitive information is sent to cloud servers. In addition, it provides a reliable and uninterrupted network connection, more specifically, the local data processing and storage of IoT applications consume less bandwidth [26]. The study [27] introduces an edge computing platform-dependent smart home power demand inferencing system. The

system uses an electric meter as a sensor to collect data to forecast the next day's power demand. The system integrates with edge computing to provide capabilities such as collecting power usage data in real-time, improving data privacy, fast processing, low latency, and sufficient computing resource. Besides, using edge computing instead of cloud to offloading tasks to reduce bandwidth and response time. To satisfy the smart home application and power control, the study [28] proposes a sustainable edge computing-based power controlling framework. the purpose of this framework is to reduce the electric cost while making full use of renewable energy. The framework was developed with tiny and affordable computers. The study [29] using multiple microcontrollers to design an edge gateway for providing enhanced computing power to the IoT devices. With the enhanced computing power of the gateway, the industrial IoT system reduces the delay time of transferring the information to the cloud and decreases the response time of communication with IoT nodes. The edge gateway selects low-cost microcontrollers to conduct distributed computing that further shortens the power usage.

There are the following frameworks to implement edge computing. Microsoft Azure introduce Azure IoT Edge to develop edge computing. To simplify the implementation of services of image recognition and machine learning and other services, there are many functionalities provided such as Azure ML, Azure stream analytics, and Azure functions. The edge nodes could be any devices supporting computing capabilities [30]. AT&T initiate an ONF project CORD [31] to use the resource of network infrastructures to construct edge data centers. Network equipment providers offer closed proprietary integrated network infrastructure systems. Due to the reliance on network equipment suppliers, networking opportunities with network operators are difficult to manage. Therefore, computer and network resources are wasted. CORD utilizes the wasted resource of network infrastructures to construct data centers through network function virtualization and software-defined networks functionalities. It uses reconstructed data centers to provide clouds capabilities at the edge of the network. EdgeX edge computing platform is a vendor-neutral open source for implementing an edge computing paradigm. Since EdgeX is a vendor-neutral platform, it could be installed on any device that provides computing resources including routers, gateways, and other nodes. EdgeX is implemented by dividing functions into several layers. In particular, in the connection layer, there are microservices implemented with various protocols, and by providing a framework, users can easily implement connection services using the desired connection protocol. In the development language, not only Java but also C and Go lang languages can be used to implement services [32]. When comparing the investigated platforms, EdgeX was selected as the development platform, which can satisfy the requirements of heterogeneous devices of the IoT devices without depending on the development language as well as on the device.

The optimization problems consist of finding the best solution from all possible solutions that can be solved by optimization functions. There are exact and approximate optimization methods [33]. The small-scale optimization problems are suitable with exact methods. Instead, the high-dimensional optimization problems are suitable with approximate methods [34]. Due to their performance in investigating high dimensional search spaces, approximation methods are more effective than precision methods in solving the problem of power scheduling in a smart home. The approximation functions are comprising approximation algorithms and heuristic/metaheuristic algorithms [35]. Some studies implement a smart home energy management system using a metaheuristic algorithm to optimize the power consumption of home appliances. The author [36] introduces Binary PSO (BPSO) and Genetic Algorithm (GA) to optimize Peak-to-Average Ratio (PAR) and Electricity Bill (EB). The total EB is calculated with the time of use pricing mechanism in the scheduling tasks for the executions of smart home appliances for a day. The author [37] presents a home energy management system to deficiently control home appliances. The objective of the system reduces EB while peak periods. To generate an appropriate scheduling scheme for home appliances the PSO and GA metaheuristic optimization algorithms are adopted. The author [38] uses the BPSO optimization algorithm to balance the working

capacity in a smart home. The power is supported by traditional and green energy. The optimization problem is aimed at minimizing the EB and maximizing the power consumption of green energy. The researchers are considering user comfort level as one of the objectives of optimization problems. The delay time rate of the executing for actuators is used to evaluate the user comfort level [39]. However, the authors ignore the user's thermal comfort by focusing on minimizing energy use and maximizing user satisfaction, i.e., the running time of home appliances.

The PSO is a promising optimization algorithm. Due to its ease of implementation, high convergence speed, parallel operation, it has been employed by a variety of applications [38]. To satisfy resident thermal comfort in smart homes, we select the PSO algorithm to develop our proposed PMV based optimization scheme for efficient thermal comfort control in a smart home environment.

### 3. Design of Smart Home Space and Environment Optimization Scheme for Efficient Thermal Comfort Control

Due to the continuous development of the Internet of Things, people's living environments are greatly affected. In particular, the study of the smart home is transforming the house into a more intelligent and smarter object rather than a simple living environment. Through smart homes, people are benefiting from the customized housing environment in the popular housing environment. In order to provide a residential environment customized for resident thermal comfort, we propose an environment optimization scheme based on edge computing using PSO for efficient thermal comfort control in resident space. The proposed housing environment was designed as follows.

As shown in the following Figure 1, the proposed system consists of edge computing and a smart home environment component. The smart home environment is a space where the occupant lives. The environment is equipped with temperature, humidity sensors, and heating actuators. Temperature sensors are installed both inside and outside the house to collect temperature information around the house. Similarly, humidity sensors are used to collect information about the humidity around the house. The edge computing components connect with sensors and actuators via the internet. The smart home-related software services are located on the edge computing component. The service consists of three layers. The devices connectivity layer provides a connection to IoT devices, the core services layer conducts supporting services, and the environment optimization services are located on the top of the layers.

To provide the best living environment to users, we propose the following system structure. It consists of three tiers: smart home, IoT device, and edge computing platform.

Since it is not possible to build a smart home physical environment, we design a smart home environment emulator as follows. The smart home environment emulator consists of an indoor environment inference model, a power inference model, an outdoor environment generator, and a smart home environment data. The smart home emulator trained an indoor environment inference model based on the smart home environment data to generate the parameters of the indoor home environment. In response to the data transmitted from the actuator, it outputs the indoor home environment data for the next period. Also, the power inference model is trained with the LSTM deep learning algorithm based on smart home environment data. Based on the external input data, the energy used in the next period is output. The outdoor environment data is extracted based on the smart home environment data. All data is input to the edge computing platform through the sensors of the IoT devices layer.

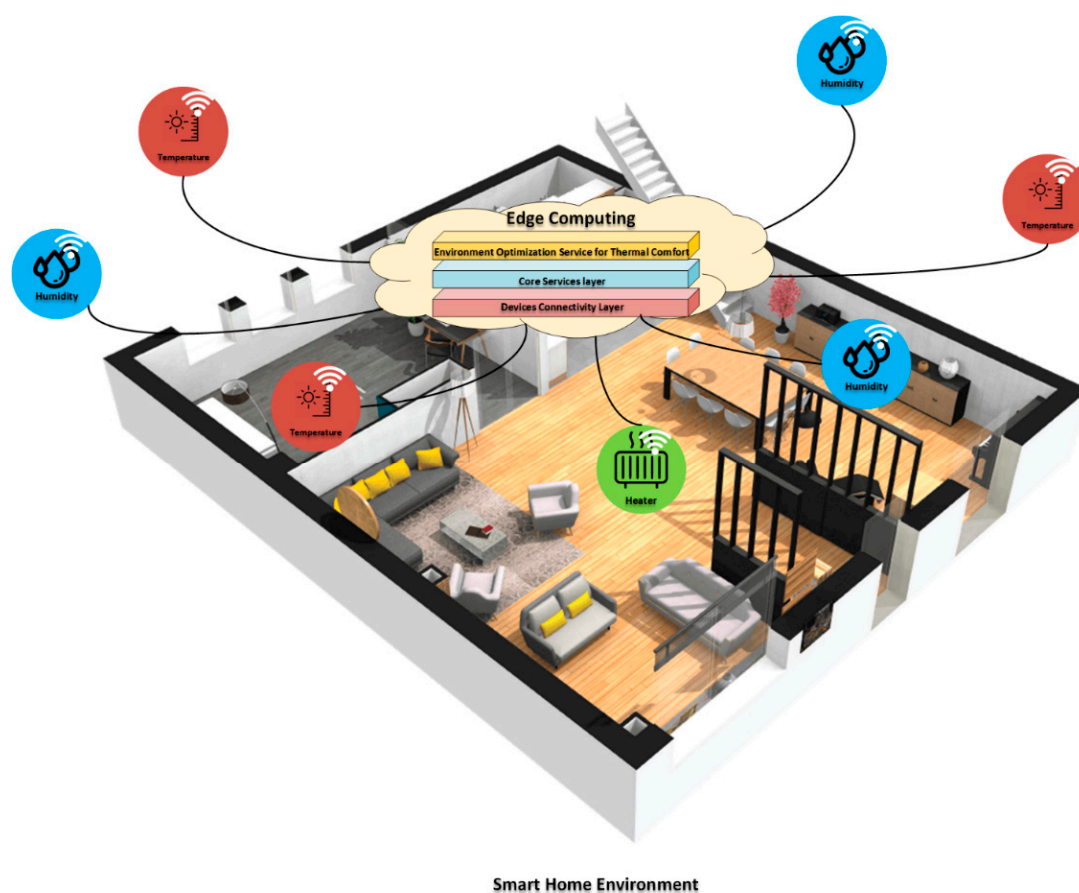
IoT devices are directly connected to the smart home and serve to transfer data between the edge computing platform and the smart home. The edge computing platform is running services to provide a suitable living environment for residents. It mainly includes the EdgeX framework and optimization engine. Core data stores data input from IoT devices and transmits it to the rule engine at the same time. The rule engine compares it with a predefined rule and, if matched, proceeds with a corresponding reaction. EdgeX framework detects environmental changes through the rules engine. Core commands are



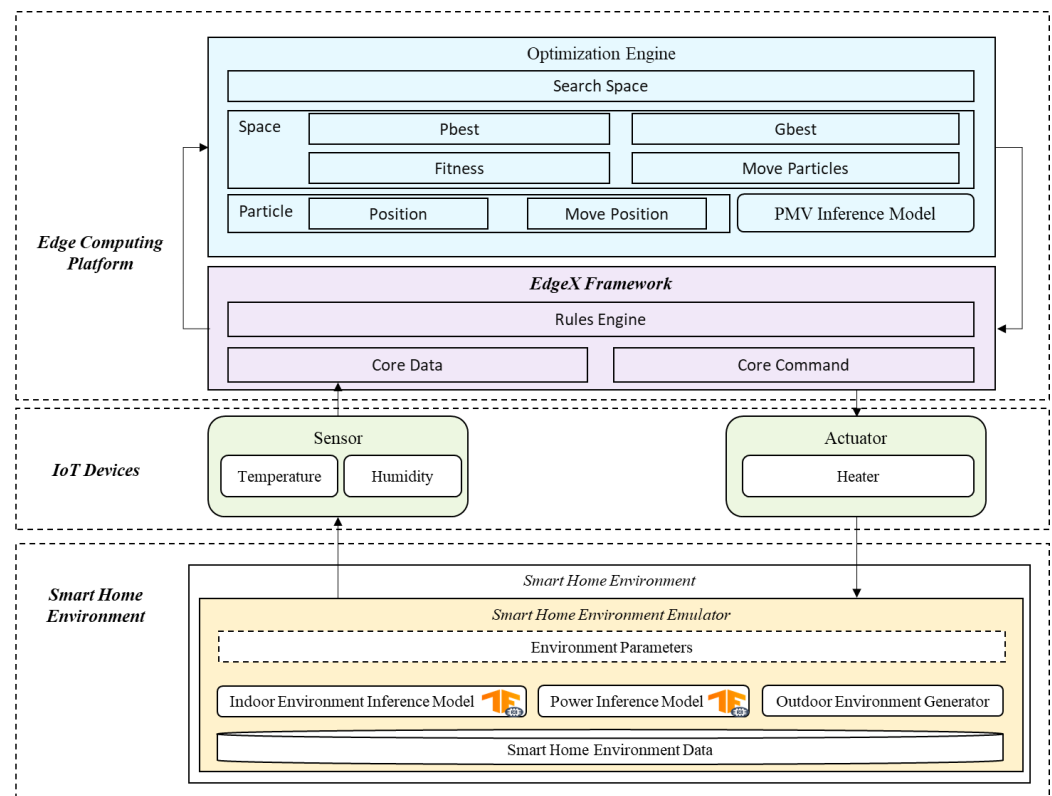
responsible for sending commands to IoT devices. To optimize the thermal comfortable indoor environment parameters, the optimization engine initializes a population of candidate solutions based on data that feed on the rules engine and updating these candidates around in the search space according to the best PMV level over the particle's attributes such as velocity and position. With each iteration, each particle is updated by its local best position, but it is also guided to the best position in the search space and these positions are updated as other particles find better positions. Finally, it prints out the optimal temperature and humidity values to the rules engine. The detailed information please refer to Figure 2.

The well-known thermal comfort model PMV [40] is developed in 1970 and employed by Standard 55 of ASHRAE [41]. To design a ventilation system and evaluate acceptable indoor air quality, the standard is adopted. The thermal comfort level is ranging from  $-3$  to  $3$  to represent the thermal response of occupants such as cold, cool, slightly cool, neutral, slightly warm, warm, and hot. The PMV level is influenced by environmental factors and vital factors such as air temperature, mean radiant temperature, relative humidity, air velocity, metabolic rate, and clothing insulation. The corresponding comfort level takes the form of the regression function.

In this paper, we use the following features to training the PMV inference linear regression model. There are environmental features such as indoor air temperature (Top), relative humidity (Rh), outdoor temperature (Tout) as well as personal features like clothing insulation (Clo) and metabolic rates of activity (MET). The factors of the PMV inference model are presented in Table 1.



**Figure 1.** Conceptual Smart Home Environment of Edge Computing for Efficient Thermal Comfort Control.



**Figure 2.** Proposed Configuration of Smart Homes based on PSO Optimization Scheme.

**Table 1.** The Factors of PMV Inference Model.

Definition	Description
Top	Indoor air temperature
Rh	Indoor relative humidity
Met	Metabolic rates of activity
Tout	Outdoor temperature
Clo	Clothing insulation

For the convergence of the PMV inference linear regression model, we use the stochastic gradient descent scheme (SGD). The SGD is an iterative function to optimize an objective function with suitable smoothness properties. Linear regression is machine learning which trying to minimizing a loss function that has the form of a sum:

$$L(w) = \frac{1}{n} \sum_{i=1}^n l_i(w) \quad (1)$$

In the formula, for minimizing  $L(w)$ , the parameter  $w$  is to be estimated. The  $i$ th observation from the dataset (for training) is associated with every summand  $l_i$ . The following formula is adopted to perform a standard gradient descent method:

$$w = w - \alpha \nabla L(w) = w - \frac{\alpha}{n} \sum_{i=1}^n \nabla l_i(w) \quad (2)$$

The  $\alpha$  is called the learning rate and affects the convergence rate of machine learning. In general, the evaluation of the summation gradient may require an expensive evaluation of the gradients of all added functions. If the training set is large and there is no simple for-

mula, the sum of the evaluation gradients becomes very expensive because the evaluation gradient has to evaluate the gradients of all added functions. To save the computational cost of each iteration, the stochastic gradient descent samples a subset of the added function at each step. This is very effective for large-scale machine learning problems.

We integrate the PSO mechanism with the PMV model to find out the comfortable temperature and humidity of the indoor environment for the occupant. PSO iteratively improves a candidate solution with a given measure of quality to optimize a problem. To optimize a problem, it initializes a population of candidate solutions, here dubbed particles, and updating these candidates around in the search space according to a given formula over the particle's attributes such as velocity and position. The movement of each particle is influenced by its local best position, but it is also guided to the best position in the search space and these positions are updated as other particles find better positions.

We instance a PMV model as pmv and initialize target thermal comfort level and target error as target and target-error accordingly. We initialize the  $S$  as the number of particles in the swarm, each having a position  $x_i$  in the search space and a velocity  $v_i$ . The  $x_i$  is Initialized with a random vector from current temperature and humidity that is feed from a smart home. The  $p_i$  are the best-known position of particle  $i$  and the  $g$  be the best-known position of the entire swarm. The particles represent the candidate solution which is the comfortable temperature and humidity. With the iteration, each particle would have a thermal comfort level (TCL) through the PMV model. To find the best environmental variable under a termination criterion,  $p_i$  or  $g$  compares the distance from the target value in each iteration and substitutes a closer value. The break criterion would be a number of iteration or a condition that the PSO finds the best environment parameter. The particles and velocity are updated at the end of an iteration through a simple formula. Figure 3, introduces the PSO algorithm of PMV based optimization scheme.

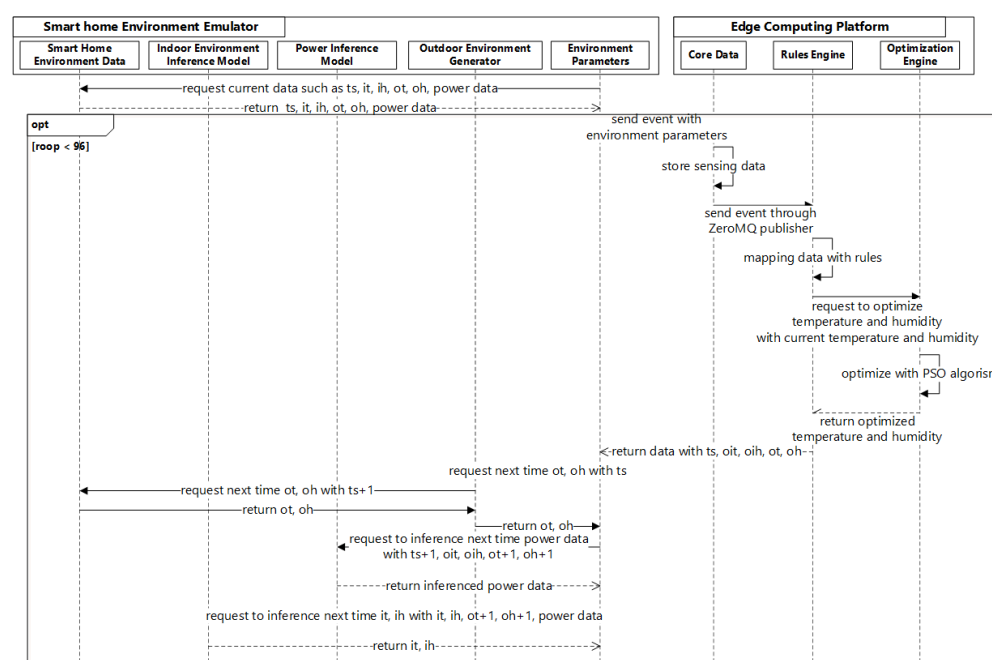
1. **Input:** current temperature and humidity.
2. **Output:** optimized temperature and humidity.
3. Instantiation PMV model as pmv
4. Initialize target thermal comfort level and target error as target and target-error
5. **for** each particle  $i = 1, \dots, S$  do
6.     Initialize the particle's position with random vector from input:  $x_i$
7.     Initialize the particle's best known position to its initial position:  $p_i \leftarrow x_i$
8.     Initialize the particle's velocity:  $v_i$
9. **end for**
10. **while** a termination criterion is not met do:
11.   **for** each particle  $i = 1, \dots, S$  do
12.     if  $\text{abs}(\text{pmv}(x_i) - \text{target}) < \text{abs}(\text{pmv}(p_i) - \text{target})$  then
13.       Update the particle's best known position:  $p_i \leftarrow x_i$
14.     if  $\text{abs}(\text{pmv}(p_i) - \text{target}) < \text{abs}(\text{pmv}(g) - \text{target})$  then
15.       Update the swarm's best known position:  $g \leftarrow p_i$
16.     if  $\text{abs}(\text{pmv}(g) - \text{target}) \leq \text{target-error}$  then
17.       break
18.     Update the particle's velocity.
19.   **end for**
20.   Update the particle.
21. **end while**
22. Output set optimized temperature and humidity from  $g$ .
23. **return** output.

Figure 3. PSO Algorithm of PMV based Optimization Scheme.

To evaluate the performance of our proposed system, an experiment is conducted according to the following procedure. First, in the smart home environment emulator, envi-



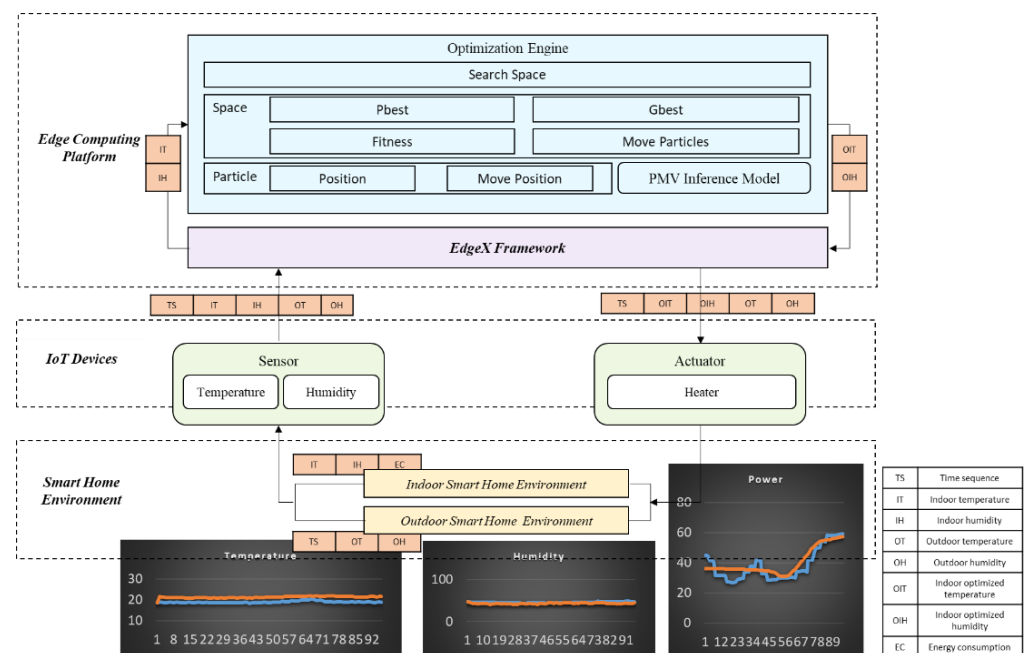
ronment parameters request current data from smart home environment data. The smart home environment data includes ts indicating the order, it and ih indicating the indoor temperature and humidity, and ot and oh indicating the external environment temperature and humidity. Finally, power data used to maintain the environment is included. The requested data is delivered as an event to the core data of the edge computing platform. Core data stores the received data and then publishes it as an event for the rules engine to process. The rule engine detects events and maps them to the registered rules. After mapping, processing proceeds according to the rules. Here, the optimal environmental data is requested based on the currently collected data through the optimization engine. The optimized data is transmitted back to the smart home environment emulator. Based on the optimization data, the smart home environment uses the power inference model, the indoor environment inference model, and the outdoor environment generator to generate the indoor temperature and humidity at the next time, the energy used, and the outdoor temperature. Finally, it passes back to the edge computing platform to get back the optimal parameters. This cycle goes 96 times. Figure 4, illustrates the sequence diagram of the proposed system operation for efficient thermal comfort in the smart home.



**Figure 4.** Sequence Diagram of Proposed System Operation for Efficient Thermal Comfort.

## 4. Implementation and Result

The structure of our proposed system is implemented as shown in Figure 5. Indoor or outdoor data is generated in a virtual smart home environment, and the generated data is transmitted to the edge computing platform by sensors, which are IoT devices. The data collected from sensors would feed to the optimization engine as input. The optimization engine returns the optimum temperature and humidity for the current environment state based on the temperature and humidity which is input from sensors. Finally, the optimal temperature and humidity are transferred to the actuator, which is an IoT device, and applied to the virtual smart home environment. By repeating the above procedure, it is always executed to keep the virtual smart home environment in an optimal state.



**Figure 5.** Developed Structure of Smart Home Environment Optimization Scheme based on Edge Computing Platform.

Our proposed system was implemented in the following development environment. To better understand and reflect the patterns of given environmental data, we developed PMV inference models, Power inference models, and Indoor environment inference models using the TensorFlow platform and Python language. The PSO algorithm based on the PMV inference model that optimizes the user's appropriate environment temperature and humidity was also developed based on the Python language. The rule engine is implemented based on Java to detect changes in the environment of the smart home in real-time and respond according to appropriate rules. The overall development was carried out using a Windows-based desktop. Refer to Table 2 for specific development environment settings.

**Table 2.** Specifications of Development Environment for Environment Optimization Scheme based on Edge Computing Platform.

Hardware			Software	
Desktop	OS	Windows 10	Library	Java
				A statically typed and compiled language
	CPU	Intel® core™ i5-8500		Python
				A dynamically typed and interpreted language
			Application	Eclipse
	Memory	64 GB		An integrated development environment for java
			Framework	Pycharm
	Hard Disk	500 GB		An integrated development environment for python
				Spring boot
			Platform	It easy to create stand-alone, production-grade Spring based Applications that you can “just run”
				Flask
				A microframework for Python web application
				TensorFlow
				An opensource platform for machine learning

Figure 6, presents the source code of implemented PMV based PSO algorithm. The PSO algorithm has been implemented in several classes. The Particle class at line 5 is a set of optimum temperature and humidity for our purposes. At the beginning of the program execution, values are generated randomly based on the current temperature and humidity mentioned in line 7. Depending on the value of the variable called velocity, the

value is repeatedly updated until the optimal value is found. The Space class at line 15 is responsible for finding the optimal value for each loop. In the fitness function at line 23, we find the values of temperature and humidity that generate the optimal thermal comfort index by finding the PMV for the parameters of the temperature/humidity set in the particle. Lastly, the PSOAlgorithm class at line 51 is a class where the main function to find the optimal value using space and particles is located. When the PSOAlgorithm class is created, the initialization function is executed. The number of iterations, the error range, the number of particles, the target value, and the current temperature and humidity is entered as parameters. Here, the number of iterations refers to the number of attempts to find the optimal environmental variable, the target value refers to the optimal environmental variable to be found, that is, the thermal comfort level and the error range expresses the allowable error value. When the optimization function in line 59 is called, the Space class and particles are initialized by the number of particles and the temperature and humidity values. And the while function is executed, the break criterion of the while function would be a number of iteration or a condition that the PSO finds the best environment parameter.

```

1  W = 0.5
2  c1 = 0.8
3  c2 = 0.9
4  pi = PMVInferenceEngine()
5  class Particle():
6      def __init__(self, temperature, humidity):
7          self.position = np.array([temperature + random.random() * (abs(25-temperature)), humidity + random.random() * (abs(45-humidity))])
8          self.pbest_position = self.position
9          self.pbest_value = float('inf')
10         self.velocity = np.array([0,0])
11     def __str__(self):
12         print("I am at ", self.position, " meu pbest is ", self.pbest_position)
13     def move(self):
14         self.position = self.position + self.velocity
15 class Space():
16     def __init__(self, target, target_error, n_particles, temperature, humidity):
17         self.target = target
18         self.target_error = target_error
19         self.n_particles = n_particles
20         self.particles = []
21         self.gbest_value = float('inf')
22         self.gbest_position = np.array([temperature + random.random() * (abs(25-temperature)), humidity + random.random() * (abs(45-humidity))])
23     def fitness(self, particle):
24         top = particle.position[0]
25         rh = particle.position[1]
26         met = 1.25
27         tout = 18.5
28         clo_tot = 0.985
29         data = [[top, met, rh, tout, clo_tot]]
30         pmv = pi.predictPMV(data)
31         return pmv
32     def set_pbest(self):
33         for particle in self.particles:
34             fitness_cadidate = self.fitness(particle)
35             if(abs(particle.pbest_value-self.target) > abs(fitness_cadidate-self.target)):
36                 particle.pbest_value = fitness_cadidate
37                 particle.pbest_position = particle.position
38     def set_gbest(self):
39         for particle in self.particles:
40             best_fitness_cadidate = self.fitness(particle)
41             if(abs(self.gbest_value-self.target) > abs(best_fitness_cadidate-self.target)):
42                 self.gbest_value = best_fitness_cadidate
43                 self.gbest_position = particle.position
44     def move_particles(self):
45         for particle in self.particles:
46             global W
47             new_velocity = (W*particle.velocity) + (c1*random.random()) * (particle.pbest_position - particle.position) + \
48                 (random.random()*c2) * (self.gbest_position - particle.position)
49             particle.velocity = new_velocity
50             particle.move()
51 class PSOAlgorithm():
52     def __init__(self, ni, te, np, target, temperature, humidity):
53         self.n_iterations = ni
54         self.target_error = te
55         self.n_particles = np
56         self.target = target
57         self.temperature = temperature
58         self.humidity = humidity
59     def optimize(self):
60         search_space = Space(self.target, self.target_error, self.n_particles, self.temperature, self.humidity)
61         particles_vector = [Particle(self.temperature, self.humidity) for _ in range(search_space.n_particles)]
62         search_space.particles = particles_vector
63         iteration = 0
64         while (iteration < self.n_iterations):
65             search_space.set_pbest()
66             search_space.set_gbest()
67             if (abs(search_space.gbest_value - search_space.target) <= search_space.target_error):
68                 break
69             search_space.move_particles()
70             iteration += 1
71         return search_space.gbest_position[0], search_space.gbest_position[1]

```

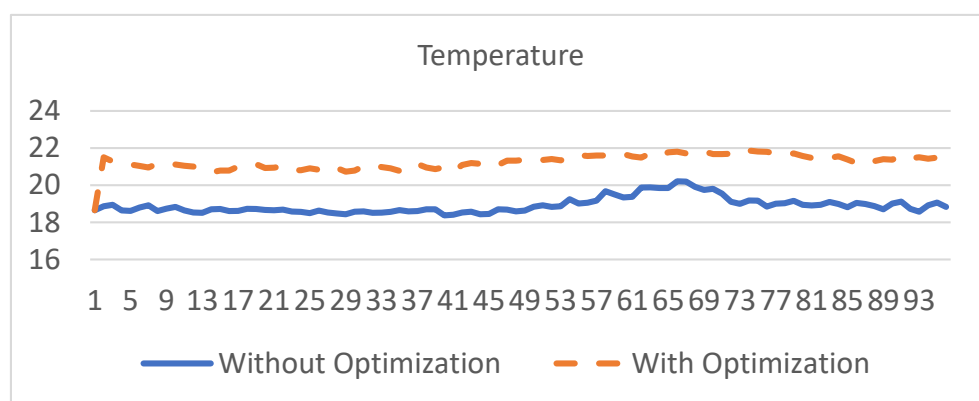
Figure 6. Source Code of Implemented PMV based PSO Algorithm.

## 5. Performance and Evaluation

For evaluating our proposed resident environment maintenance with thermal comfort index based on PSO optimization scheme in smart homes. We conduct a comparison between the real data with optimized data. We experiment with our proposed system with one day of the ORNL dataset [42]. The ORNL dataset was sensing from a home where it is located in Campbell, United States. The dataset is composed of 35,040 rows and

several sensing data of indoor and outdoor climate such as temperature and humidity. We predict the PMV of thermal comfort, which is inference with the optimized temperature and humidity for maintenance house thermal environment. Once the indoor environment data is provided from the simulated smart home, then the PSO-based optimization engine would invoke to predict the optimized temperature and humidity then applied to the smart home to maintain thermal comfort of resident in a house.

The following Figure 7 shows the results of the temperature data collected in our experiment. With the graph, you can easily understand the difference in temperature that using the optimization method and the unused data. The bottom graph reflects changes from the original temperature data. The indoor temperature varies between 18.37 and 20.21, with an average temperature of 18.90, with most readings below 20. Conversely, the experimental results using our proposed optimization method lie between 18.65 and 21.92. Compared to the original data, the average temperature using our proposed optimization method is 21.25, maintaining a warmer environment.



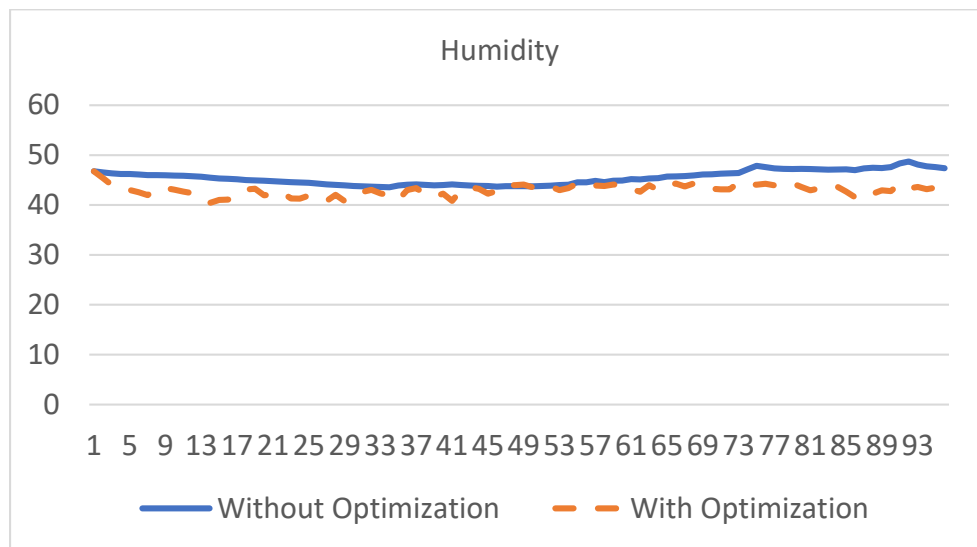
**Figure 7.** Comparison Results between Current and Optimized Indoor Temperature for Efficient Thermal Comfort Control.

Figure 8 shows the current indoor humidity and optimized indoor humidity data. When using our optimization method, changes in humidity are not as sharp as changes in temperature. The current humidity fluctuates at 45.49 degrees left and right, while the optimized temperature is changing at 43.02 degrees left and right. However, we have much better humidity stability since the deviation was less than 1.3, where for original data it was almost 2.

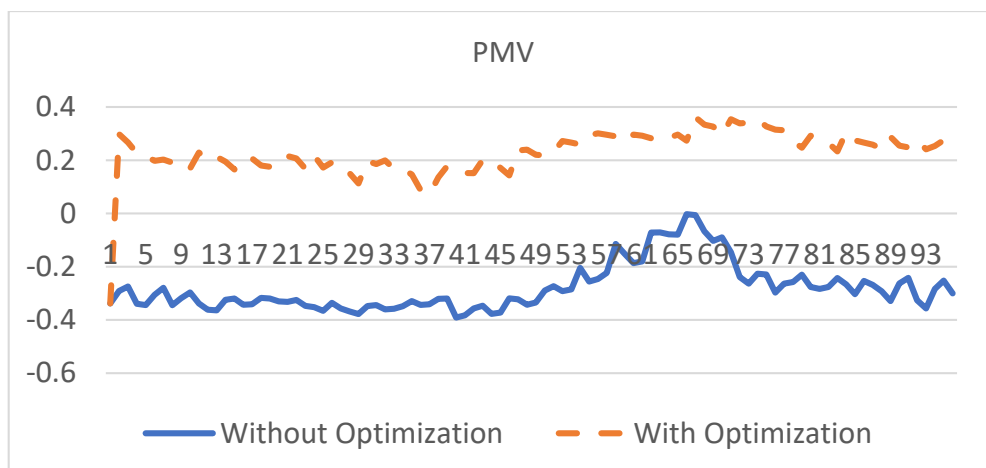
Figure 9 shows the current indoor thermal comfort index and optimized indoor thermal comfort index data. The current PMV fluctuates at  $-0.280$  degrees left and right, while the optimized temperature is changing at  $0.228$  degrees left and right. The results show us that the optimized PMV is warmer than the current PMV.

Figure 10 is the result of power usage. The following figure shows the data of energy used to maintain the indoor environment as a graph. As can be seen from the figure, the use of energy is more stable while using the optimization method. The current indoor overall power usage is 3667 and the optimized power usage is 3871. Our proposed system maintains a comfortable indoor environment for residents and uses more energy than the original data in terms of power, but there is no noticeable difference, with a difference of 204.

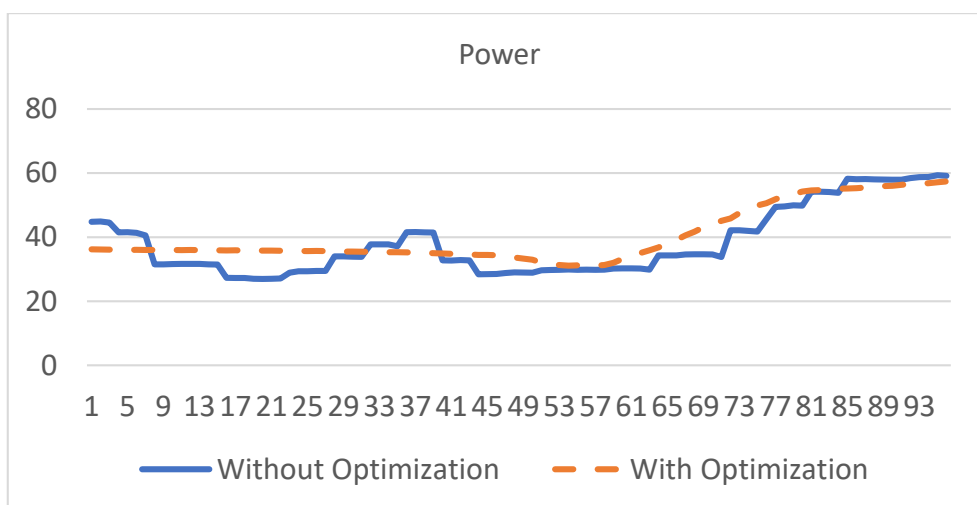
Figure 11 shows the result of measuring the time required of the optimization function. The unit is milliseconds and the maximum time required is 54, which returns optimal results in a very short time. Most of the travel time is on the left and right of 33 and 42. In conclusion, it can be seen that it is executed almost in real-time.



**Figure 8.** Comparison Results between Current and Optimized Indoor Humidity for Efficient Thermal Comfort Control.

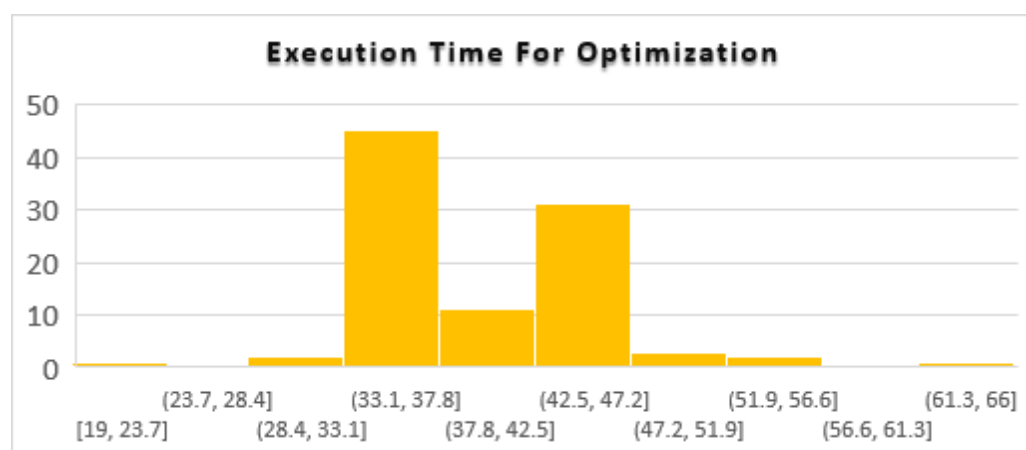


**Figure 9.** Comparison Results between Current and Optimized Indoor PMV for Efficient Thermal Comfort Control.



**Figure 10.** Comparison Results between Current and Optimized Indoor Consumption Power.





**Figure 11.** Results of Optimization Engine Execution Time for Environment Optimization.

We experiment with the proposed system in a simulated smart home environment. The smart home environment parameters are supported from one day of the ORNL dataset. From the experimental results, our proposed system maintained a better environment than the original data in temperature and humidity. Although the energy consumption was slightly overdone, it did not negatively affect the real-time processing capability of the IoT. The shortcomings of the resource shortage of IoT devices were covered by the introduction of edge computing, and the thermal comfort level was automatically maintained in the smart home through the integration of the PSO algorithm and the PMV model.

## 6. Conclusions

In this research, we investigate smart home control-related technologies. We discover that IoT devices not only lack processing and storage capabilities but also lack an automation function that monitors the indoor environment of a smart home in real-time to maintain an optimized environment for the occupant. To maintain an indoor environment with optimized temperature and humidity, we Implement an environment optimization scheme for efficient thermal comfort control in a smart home environment. The optimization scheme depends on the PMV index to consciousness thermal comfort of smart home, realized with PSO algorithm. We integrate IoT with the edge computing paradigm to enhance the capability of IoT devices. The EdgeX platform is used to develop a rules engine to sense the environment parameter of the smart home and control the indoor environment thermal comfortable. With the experimental results, the proposed system provides a better environment than the original data. Although the power consumption is not ideal, it did not affect the real-time processing capability of the IoT. Through edge computing and IoT integration, the drawbacks of the IoT are solved, and the thermal comfort level was automatically maintained in the smart home through the implementation of the PSO algorithm and the PMV model.

The future work would be oriented on executing the system for a real smart home environment or integrate a cloud computing paradigm to offloading powerful computing tasks to it to improve computing performance. Cloud computing provides flexible computing and storage space, but network latency would be a challenge to integrate with IoT and edge computing.

**Author Contributions:** R.X. implemented the PMV based optimization engine and rules engine to provide efficient thermal comfort control in a smart home, and wrote this paper; W.J. assisted in developing the smart home environment emulator; D.K. figured the full idea of the paper and proofread the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2016-0-00313) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation),

and This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT)), Any correspondence related to this paper should be addressed to DoHyeun Kim.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

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

## References

- Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [\[CrossRef\]](#)
- Risteska, S.B.L.; Trivodaliev, K.V. A review of Internet of Things for smart home: Challenges and solutions. *J. Clean. Prod.* **2017**, *140*, 1454–1464. [\[CrossRef\]](#)
- Alaa, M.; Zaidan, A.A.; Talal, M.; Kiah, M.L.M. A review of smart home applications based on Internet of Things. *J. Netw. Comput. Appl.* **2017**, *97*, 48–65. [\[CrossRef\]](#)
- Harper, R. *Inside the Smart Home*; Springer Science & Business Media: London, UK, 2006.
- Remo, L. Smart Home Concept and the Integration of Energy Meters into a Home Based System. In Proceedings of the Seventh International Conference on Metering Apparatus and Tariffs for Electricity Supply, Glasgow, UK, 17–19 November 1992.
- Galinina, O.; Mikhaylov, K.; Andreev, S.; Turlikov, A.; Koucheryavy, Y. Smart home gateway system over Bluetooth low energy with wireless energy transfer capability. *EURASIP J. Wirel. Commun. Netw.* **2015**, *2015*, 1–18. [\[CrossRef\]](#)
- Jin, W.; Ullah, I.; Ahmad, S.; Kim, D. Occupant comfort management based on energy optimization using an environment prediction model in smart homes. *Sustainability* **2019**, *11*, 997. [\[CrossRef\]](#)
- Gao, G.; Li, J.; Wen, Y. DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet Things J.* **2020**, *7*, 8472–8484. [\[CrossRef\]](#)
- Sukandar, S.; Pongoh, D.S.; Ramschie, A.A.S. Design of Smart Home Control System Based on Android. In Proceedings of the 2018 International Conference on Applied Science and Technology (iCAST), Manado, Indonesia, 26–27 October 2018.
- Gunawan, T.; Yaldi, I.R.H.; Kartiwi, M.; Ismail, N.; Za'bah, N.F.; Mansor, H.; Nordin, A.N. Prototype design of smart home system using internet of things. *Indones. J. Electr. Eng. Comput. Sci.* **2017**, *7*, 107–115. [\[CrossRef\]](#)
- Botta, A.; De Donato, W.; Persico, V.; Pescapé, A. Integration of cloud computing and internet of things: A survey. *Future Gener. Comput. Syst.* **2016**, *56*, 684–700. [\[CrossRef\]](#)
- Zhou, J.; Leppanen, T.; Harjula, E.; Ylianttila, M.; Ojala, T.; Yu, C.; Jin, H.; Yang, L.T. Cloudthings: A Common architecture for integrating the internet of things with cloud computing. In Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Whistler, BC, Canada, 27–29 June 2013.
- Babu, S.M.; Lakshmi, A.J.; Rao, B.T. A study on cloud based Internet of Things: CloudIoT. In Proceedings of the 2015 Global Conference on Communication Technologies (GCCT) IEEE, Thuckalay, India, 23–24 April 2015.
- Kumar, D.S.; Mohapatra, S.; Pattnaik, P.K. A survey on applications of wireless sensor network using cloud computing. *Int. J. Comput. Sci. Emerg. Technol.* **2010**, *1*, 50–55.
- Progress, M.; Khubisa, F. A smart home appliance control system for physically disabled people. In Proceedings of the 2019 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 6–8 March 2019.
- Zielonka, A.; Sikora, A.; Wozniak, M.; Wei, W.; Ke, Q.; Bai, Z. Intelligent Internet of things system for smart home optimal convection. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4308–4317. [\[CrossRef\]](#)
- Botta, A.; de Donato, W.; Persico, V.; Pescapé, A. On the integration of cloud computing and internet of things. In Proceedings of the Future Internet of Things and Cloud (FiCloud) (2014), Barcelona, Spain, 27–29 August 2014; pp. 23–30.
- Suciu, G.; Vulpe, A.; Halunga, S.; Fratu, O.; Todoran, G.; Suciu, V. Smart cities built on resilient cloud computing and secure internet of things. In Proceedings of the 2013 19th International Conference on Control Systems and Computer Science, Bucharest, Romania, 29–31 May 2013.
- He, L.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Netw.* **2018**, *32*, 96–101.
- Faye, S. *Teaching Homes to Be Green: Smart Homes and the Environment*; Green Alliance, Buckingham Palace Road: London, UK, 2007.
- Marikyan, D.; Papagiannidis, S.; Alamanos, E. A systematic review of the smart home literature: A user perspective. *Technol. Forecast. Soc. Chang.* **2019**, *138*, 139–154. [\[CrossRef\]](#)
- Chakraborty, T.; Datta, S.K. Home automation using edge computing and internet of things. In Proceedings of the 2017 IEEE International Symposium on Consumer Electronics (ISCE), Kuala Lumpur, Malaysia, 14–15 November 2017.
- Liu, H.; Li, S.; Sun, W. Resource Allocation for Edge Computing without Using Cloud Center in Smart Home Environment: A Pricing Approach. *Sensors* **2020**, *20*, 6545. [\[CrossRef\]](#) [\[PubMed\]](#)

24. Morabito, R.; Petrolo, R.; Loscri, V.; Mitton, N. LEGIoT: A lightweight edge gateway for the Internet of Things. *Future Gener. Comput. Syst.* **2018**, *81*, 1–15. [CrossRef]
25. Chen, C.; Lin, M.; Guo, X. High-level modeling and synthesis of smart sensor networks for Industrial Internet of Things. *Comput. Electr. Eng.* **2017**, *61*, 48–66. [CrossRef]
26. Chen, C.-H.; Lin, M.-Y.; Lin, W.-H. Designing and implementing a lightweight WSN MAC protocol for smart home networking applications. *J. Circuits Syst. Comput.* **2017**, *26*, 1750043. [CrossRef]
27. Zhou, S.; Zhang, L. Smart home electricity demand forecasting system based on edge computing. In Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018.
28. Chang, X.; Li, W.; Xia, C.; Ma, J.; Cao, J.; Khan, S.U.; Zomaya, A.Y. From insight to impact: Building a sustainable edge computing platform for smart homes. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018.
29. Chen, C.-H.; Lin, M.-Y.; Liu, C.-C. Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers. *IEEE Netw.* **2018**, *32*, 24–32. [CrossRef]
30. Microsoft Azure. Available online: <https://azure.microsoft.com/en-us/overview/iot> (accessed on 1 June 2021).
31. CORD. Available online: <https://opennetworking.org/cord> (accessed on 1 June 2021).
32. EdgeX Foundry. Available online: <https://www.edgexfoundry.org> (accessed on 1 June 2021).
33. Talbi, E.-G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons; Lille 1 University Science and Technology: Hoboken, NJ, USA, 2009; Volume 74.
34. Beheshti, Z.; Shamsuddin, S.M.H. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–35.
35. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.; Alyasseri, Z. Optimization methods for power scheduling problems in smart home: Survey. *Renew. Sustain. Energy Rev.* **2019**, *115*, 109362. [CrossRef]
36. Nawaz, F.; Ahmad, G.; Ihsanullah; Javed, K.; Khan, I.; Khan, W. An optimal Home energy management system based on time of use pricing scheme in smart grid. *Int. J. Sci. Eng. Res.* **2017**, *8*, 882–894.
37. Ranjini, A.; Zoraida, B.S.E. Intelligent residential energy management in smart grid. *Indian J. Sci. Technol.* **2016**, *9*. [CrossRef]
38. Lugo-Cordero, H.M.; Fuentes-Rivera, A.; Guha, R.K.; Ortiz-Rivera, E.I. Particle swarm optimization for load balancing in green smart homes. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011.
39. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S. Multi-objective power scheduling problem in smart homes using grey wolf optimiser. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3643–3667. [CrossRef]
40. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
41. Breima, L. Random Forests. *Mach. Learning* **2001**, *45*, 5–32. [CrossRef]
42. Long-Term Energy/Environment Data for ORNL Research House # 3. Available online: <https://openei.org/datasets/dataset/ornl-research-house-3> (accessed on 1 July 2018).