energies

MDPI

Article

Prediction-Learning Algorithm for Efficient Energy Consumption in Smart Buildings Based on Particle Regeneration and Velocity Boost in Particle Swarm Optimization Neural Networks

Sehrish Malik and DoHyeun Kim *

Computer Engineering Department, Jeju National University, Jeju-si 63243, Korea; serrym29@gmail.com
* Correspondence: kimdh@jejunu.ac.kr; Tel.: +82-64-754-3658

Received: 20 April 2018; Accepted: 14 May 2018; Published: 17 May 2018

Abstract: Electricity, the most important form of energy and an indispensable resource, primarily for commercial and residential smart buildings, faces challenges requiring its hyper efficient consumption and production. Therefore, accurate energy consumption predictions are required in order to manage and optimize the energy consumption of smart buildings. Many studies have taken advantage of the power and robustness of neural networks (NN) when it comes to accurate predictions. A few studies have also used the particle swarm optimization (PSO) algorithm along with NNs to enhance and optimize the predictions. In this work, we study prediction learning using PSO-based neural networks (PSO-NN) and propose modifications in order to increase prediction accuracy. Our proposed modifications are re-generation based PSO-NN (R-PSO-NN) and velocity boost-based PSO-NN (VB-PSO-NN). The performance metrics used are: prediction accuracy, number of particles used, and number of epochs required. We compare the results of NN, PSO-NN, R-PSO-NN and VB-PSO-NN based on the performance metrics.

Keywords: energy prediction; smart homes; neural networks (NN); particle swarm optimization (PSO)

1. Introduction

With the growing global population and escalating industrialization, the thirst for energy in the world is reaching aberrant levels. According to the World Energy Council [1], per capita global energy demand will spike in 2030, and electricity demand is predicted to double by 2060; eventually necessitating larger investments in smart buildings, households, or smart infrastructures in order to ensure energy-efficient environments.

Electricity, the preeminent source of energy for lighting, cooling and different appliances working in smart homes, is a rapidly developing source of energy. In Korea, the total consumption of electricity in residential and commercial sector is about 40.5% more than the total liquid natural gas (LNG) and city gas consumption in the residential and commercial sector [2]. Being the most important form of energy, electricity is facing a continuous challenge of increased demand worldwide. According to the U.S. Energy Information Administration (EIA), a country's energy use, mainly electricity use, is linked to its economic growth. For growing economies in the world, the increasing population is becoming a vital reason for the rise in electricity demand. The International Energy agency (IEA) has also stated that there is a strong correlation between a country's energy usage and its wealth. It has also predicted an increase of 28% in electricity usage over the period of 2015–2040, mainly in residential and commercial buildings [3]. Electricity is generated by many different energy sources as coal, natural gas, petroleum fuel, fossil fuel, nuclear, hydro, renewable fuels, geothermal and others. Typically, it takes 3

to 3.3 units of source energy to produce one unit of site energy of electricity [4]. Due to the expensive sources and generation overheads, electricity costs more in comparison to natural gas. Hence, we need to optimize our electricity energy consumption through timely and accurate energy predictions.

In recent years, our homes have been getting smarter with connected and remotely communicating devices. In order to save household energy, the Internet of Things (IoT) also called the "next great disruptor" is emerging as a significant digital revolution [5]. In South Korea, the smart home industry has come a long way since 2003. It was the time when the apartment owners focused on home networking systems at a larger scale and the government planned to build a digitalized nation. Nowadays, the idea has gained in prevalence as more buildings can be found equipped with smart environments that can control heating or lights and are saving energy consumption, although continued criticism suggests that there is still a long way to go to meet household requirements [6]. As there is relatively high proportion of energy-demanding industries in Korea, as compared to other countries, the recorded energy consumption was 244,626 $\times$ 10^6 kWh in 2013, which is the eighth largest in the world [7]. In South Korea, 38% of total electricity is consumed by residential and commercial buildings, 55% by the industrial sector, 6% by public sector and 1% by transportation (Figure 1) [2].

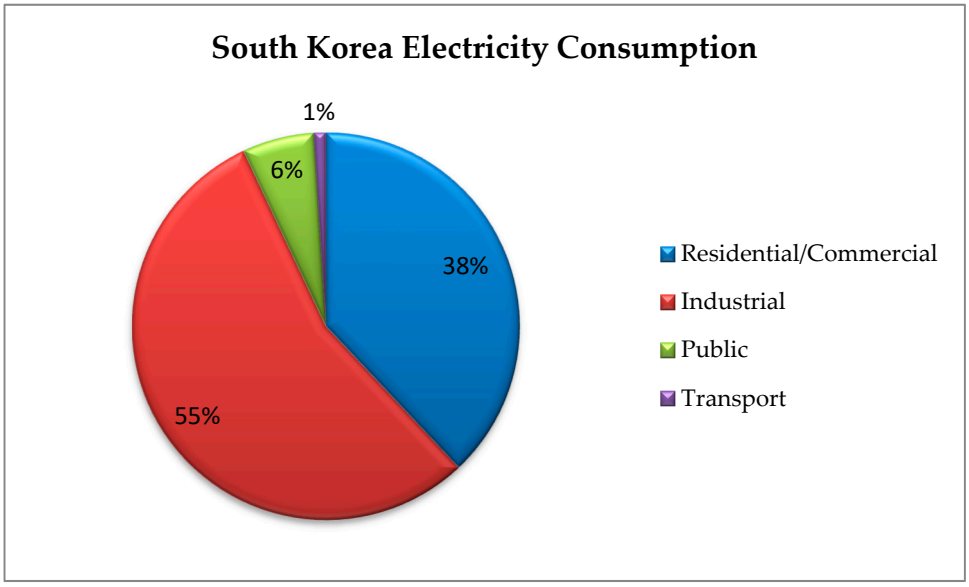

Figure 1. Energy consumption distribution [2].

As mentioned earlier, IoT-based smart homes can help in energy optimization through real-time monitoring. The primary goal of real-time energy monitoring should be exceptional predictions of energy usage. It also must help to implement equipment with accurate load balance. Earlier, when IoT systems were not in use, traditional energy management systems used to collect energy consumption data samples at an interval for monitoring purposes. Although, these traditional management systems were good at collecting energy consumption data, they failed to predict the demand, extract usage patterns, to alarm the users in case of spikes, or to suggest effective settings.

IoT is helping us save energy through the use of energy-efficient appliances, smart meters, transferring to off-peak power hours, and smart heating systems [8]. IoT-based energy management systems are very effective in order to record energy consumptions as well as to predict consumption in future. Through the use of IoT devices, we are able to collect different data about the smart home environment e.g., its temperature, humidity level, user occupancy, the operational status of appliances etc. This sort of data can be further used in different machine-learning approaches to train systems in order to predict future energy consumption. The importance of building energy consumption predictions in order to make the most informed real-time decisions has been highlighted in many

recent studies; and some contributions have come up with prediction models through data-driven or machine-learning approaches [9,10]. According to [11], these predictions are equally important and effective for both the user of a residential building and power-generating companies. These predictions can also help us optimize electricity usage at peak hours, enabling the user to be aware of his energy usage patterns. A user can use these systems to take decisions e.g., to switch between energy sources (e.g., electrical to solar), according to his requirements. In [11], it has also been shown that in South Korea commercial and residential sectors cover around 40% of national energy consumption. Therefore, predicting the energy consumption will not just help a smart home user to effectively utilize resources, it will also have a growing impact on the national economy as well. This has led us to develop an IoT-based prediction model which will help users know their electricity consumption based on previous data also to effectively manage and optimize their energy consumption.

This paper focuses on the prediction through the use of machine-learning algorithms along with optimization algorithms. For this we have proposed modification in the hybrid approach of particle swarm optimization (PSO) and neural networks (NN) which can be called the PSO-NN algorithm. As the name suggests, it combines the properties of NN and the PSO algorithm to come up with optimal weights and bias values for neural network training and hence the most accurate prediction results. For the purpose of experiments, we have used a data set of 4 buildings located in Seoul.

The rest of the paper is structured as follows. A brief state-of-the-art overview is given in Sections 2 and 3 to present our prediction methodology using PSO and NN. Section 4 describes the data set and experimental set up. The results are presented in Section 5, and Section 6 is based on discussions about possible future directions which conclude the paper.

2. Literature Review

Among all the energy sources, electricity plays a vital role. Therefore, in order to make a well-informed energy policy, it is also important to have knowledge about electricity consumption, its usage pattern and demand. There have been many studies focusing on electricity forecasting methods, and energy prediction in commercial and residential buildings has received an exceptional amount of consideration in the research community [12–15].

These forecasting methods for building energy consumption mainly include time-series analysis [16], computational intelligence methods [17–19], statistical methods and traditional multivariant regression [20] methods etc. In [13,21], a detailed review of prediction methods has been presented. As residential buildings are the substantial consumers of electricity in every country, therefore, they are being focused on for the consumption forecast. However, forecasting energy consumption for residential buildings is not simple, as the factors required for the prediction are complex and intertwined, and data unavailability for residential buildings still creates challenges for researchers leaving them with a requirement for comprehensive techniques [22]. This study has focused on providing a contemporary review of different modeling methods and techniques in order to model the energy consumption of residential buildings. Primarily, two different techniques have been identified i.e., bottom up and top down approaches. Each approach depends on various level of information used as an input, various calculations, and the output or results are generated with various applicability. Therefore, a comprehensive and critical analysis is presented along with a detailed explanation of each technique's pros and cons and comparisons of all techniques.

In order to curtail the complications of precise methods, some studies have also proposed simplified methods [23,24]. The key variables used as input information for the energy consumption prediction are very important. In a study conducted by [25], for the forecasting of electricity of the coming one month, a comprehensive list of input features is used. In total, the features amount to 20 significant variables including 14 variables related to weather, 5 variables related to social factors, and monthly electricity consumption. The proposed model, selects the significant input features by using the PSO algorithm along with support vector regression (SVR) and fuzzy rough feature selection. All the features which do not lead towards effective predictions are discarded. In order to validate the

prediction results, South Korea's historical data from 1991–2012 was used. The data was divided into training and testing data sets. The first 20 years of data was used as training data and the remainder for testing. For evaluation along with other standard matrices, it was also analyzed against the results obtained from artificial NNs (ANNs), regression and other models proposed in previous studies. The method proposed in this study appears to have an advantage over previous models due to the automatic selection of significant features for safe predictions.

The forecasting models for sensor-based energy are primarily based on machine-learning algorithms as these algorithms make use of historical data for training purpose in order to find the relationship between energy usage and key variables influencing the results e.g., peak hours etc. ANNs [20,26–28] and support vector machines (SVM) [29–31] are two basic algorithms comprehensively used in previous studies for sensor-based energy consumption predictions. ANNs have also being used for the forecasting of energy consumption of solar buildings. In another study [32], three modeling approaches were introduced for the forecasting of electricity consumption. These three approaches include the NNs, decision trees, along with the traditional regression analysis.

Two possible cases are investigated by [33] for a building which has been insulated and for a building with at least one brick wall. The investigations were conducted for both winter and summer seasons with some experimental setups. The purpose of this work was to use ANNs to generate a simulation program in order to model the thermal behavior of the building. The learning algorithm was back propagation used in a multilayer recurrent architecture.

A recent study [11] performed on energy consumption prediction also used residential buildings data from Seoul, South Korea. In this paper, the proposed model is based on the hidden Markov model based on an algorithm. The predicted results are evaluated against three main prediction algorithms namely, ANNs, SVM and classification and regression trees (CART). An improvement of 2.96%, 6.09%, and 9.03% is seen in the ANN, SVM and CART algorithms respectively. The proposed model was also analyzed on different aggregates of data, e.g., hourly, and weekly etc.

We aim to apply PSO-based NNs for making predictions for smart building energy consumption. In PSO-NN the weights and bias values for neural network training are optimized using the PSO algorithm. PSO is very well known for solving optimization problems and its many modified versions have also been tailored for different applications depending on the scenario and problems. In [34], a modified PSO approach is proposed based on the fitness of the particles and the distance between the particles in the population. In [35], a modified approach for the social and cognitive learning factors of PSO is proposed by pre-defining a predicted velocity index. Some other modifications of PSO are presented in [36–40]. For prediction purposes, PSO comes in hybrid with some learning algorithms. One more focused approach in recent times is the PSO-NN, as it has been used for many [41–48] prediction problems.

To the best of our knowledge, PSO-NN has not been used for predicting smart building energy consumption. Hence, we consider applying PSO-NN to our smart building data and also offer two modified approaches and make comparisons among them.

3. Particle Swarm Optimization-Based Neural Networks (PSO-NN) Prediction Methodology

We introduce neural networks in Section 3.1, particle swarm optimization in Section 3.2, and then elaborate the prediction methodology of particle swarm optimization based neural networks (PSO-NN) for building energy consumption prediction data in Section 3.3. In Section 3.4, we present a modified approach for PSO-NN prediction learning to improve the accuracy of our implemented PSO-NN approach.

3.1. Neural Networks (NNs)

The computational model (named as threshold logic) proposed in 1943 by McCulloch and Pitts led to the research of artificial intelligence-based neural networks [49]. Artificial neural networks

started to flourish once the processing power of computers increased dramatically, as computation power was one of the key issues faced in the progress of ANNs at the initial stages [50].

Biologically inspired ANNs are known to produce most accurate prediction results [51]. ANN learning has two operational modes of training and testing, the system has a set of inputs, weights associated with the inputs, hidden layers and a number of outputs. In training, the neuron learns to decide whether to fire an output for a specific pattern or not, while in testing mode the accuracy of the learned model is determined.

The structure of a three-layer neural network is shown in the Figure 2, where we have five inputs, six hidden layers and three outputs. The working of a simple neuron can be explained by Equation (1) [52], whereby a typical neuron computes the output in the following manner:

$$a_k = f(\sum_{i=0}^{n} w_{k_i} x_i) \tag{1}$$

where, a_k is the output of kth neuron. x_1, x_2, ..., x_n are the inputs to the neuron. x_0 input is bias (b_k)assigning it $+1$ value, with $w_{k0} = b_k = 1$. w_{k1}, w_{k2}, . . . , w_{kn} are the weights associated to each input. f is the activation function, which incorporates flexibility in the neural networks.

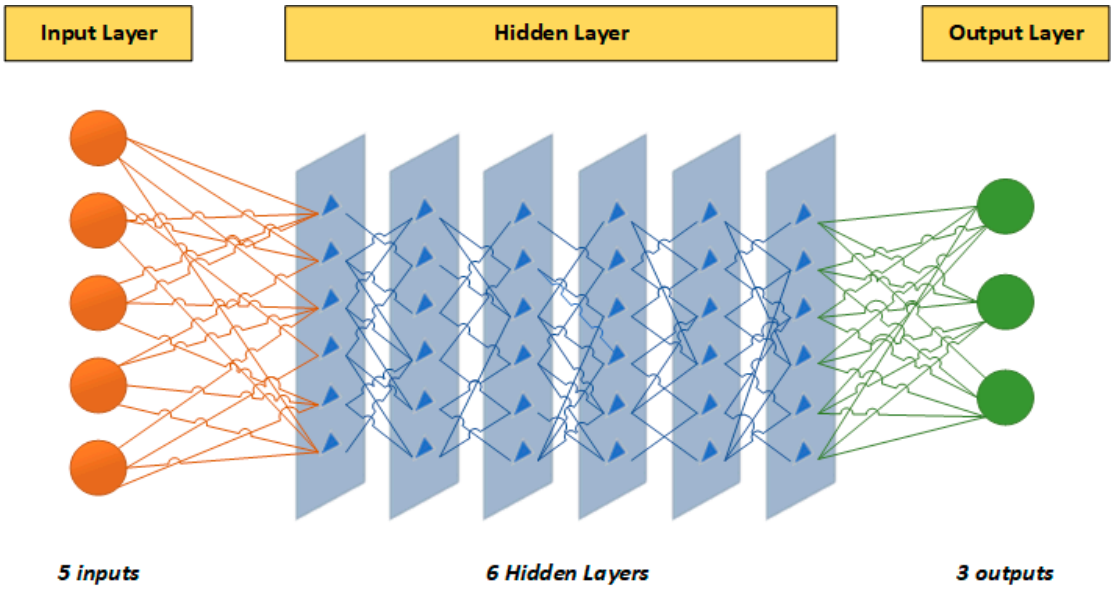

Figure 2. Neural network (NN) layers.

3.2. Particle Swarm Optimization (PSO)

In 1995, Kennedy and Eberhart proposed PSO, which is a population-based optimization technique inspired by bird flocking and fish schooling theory and also has strong ties to genetic algorithms and artificial life [53]. In the example of the search for food by flocking birds, the bird closest to the food leads and others follow. As soon as some other bird thinks it has a food source close to it, it makes a sound and all birds start following it, changing the direction. Each particle in PSO represents a bird in the flocking example, moving at a certain velocity looking for the optimal solution in the search space.

In PSO, first a population of particles is initialized defining the number of particles that will carry the search for the optimal solution. Each particle has velocity with which it moves through the search space and fitness values. The particles move in the search space by following the particles with the best solution so far. Each particle maintains the track of two values as particle's best (pbest) and global

best (gbest); pbest is the best solution achieved by the particle itself, while gbest is the best solution found by any particle in the entire population. After finding the pbest and gbest, the particle updates its velocity and position using the following equations [54].

$$v = v + c1 \times rand \times (pbest - present) + c2 \times rand \times (gbest - present) \quad (2)$$

$$present = present + v \quad (3)$$

where, v is the particle's velocity, *present* is the current particle position (solution), *pbest* is particle's personal best solution found so far in the search process, *gbest* is the global best solution found by any particle so far in the search, *rand* is a random number generated between 0 and 1, and $c1$, $c2$ are the learning factors; usually both $c1$ and $c2$ are kept 2.

3.3. Training NNs with PSO

Recently, many researchers have been focusing on using PSO for learning ANNs. PSO improves the learning and training process of an ANN in an efficient manner. In this work, we present how PSO can be used to train a neural network and improve its results. The PSO algorithm can optimize the neural network by attempting to find the optimal weight and bias values for the neural network's training.

Figure 3 presents the PSO-based NN model diagram. We are using neural networks for training our building energy consumption data to make predictions. PSO is used in order to fine tune and optimize weights and biases for the neural networks. It computes the particle's positions and passes them on for the learning process. In NN-based PSO, the population of particles generated by PSO struggles its best to look for optimal weight and bias values for the neural network's training process. The local and global positions in the PSO are updated at each iteration until the best weights are found for NN training. The analyzer checks whether best outputs are computed or more learning is needed.

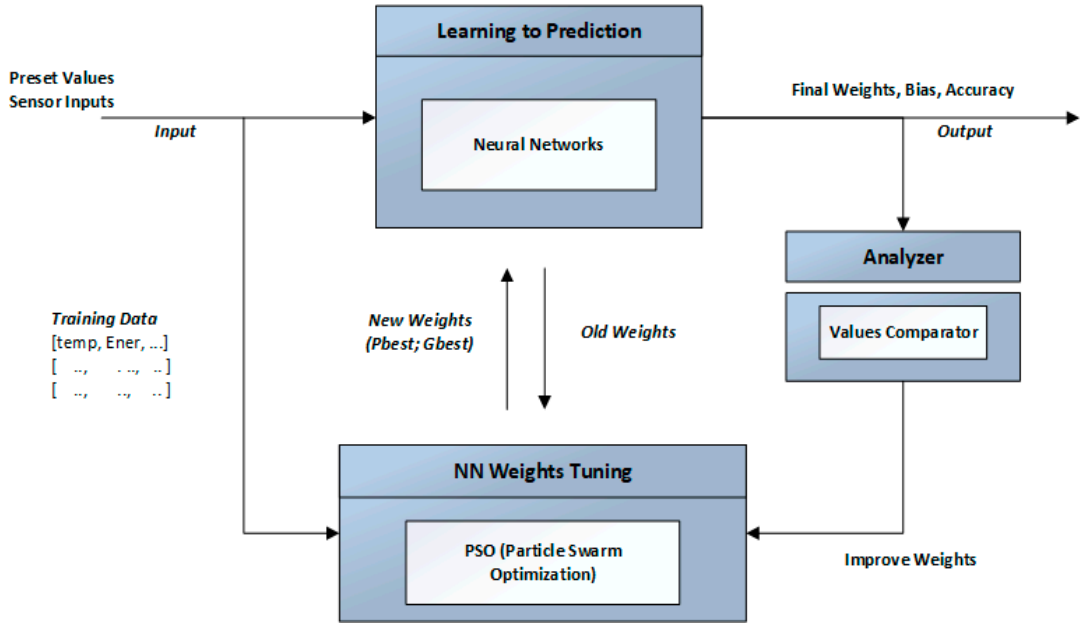

Figure 3. Particle swarm optimization-based neural network (PSO-NN) model.

Figure 4 shows the configurations of the prediction system diagram for the PSO-based NN model. The input is divided into test and training data, training data to be trained by the PSO-based neural networks, and test data for making predictions and computing accuracy results. In the processing unit, first the neural networks and particles of PSO are created. Then the initialization

for the weights, random positions and velocities of the particles ins undertaken. The PSO algorithm shuffles the particles at each iteration, and finds the new positions to be passed to neural networks for training until the best positions are found. Neural networks use two activation methods such as tanh (Equation (4), [55]) and softmax (Equation (5), [56]) for the training, and for computing the output values. The mean square (Equation (6), [57]) is computed by neural networks each time and sent back to the PSO unit for comparing whether global best value is found or not. Once the total epochs have finished running, then the final weights and biases are used to make predictions for the test data and the accuracy of the test data is computed.

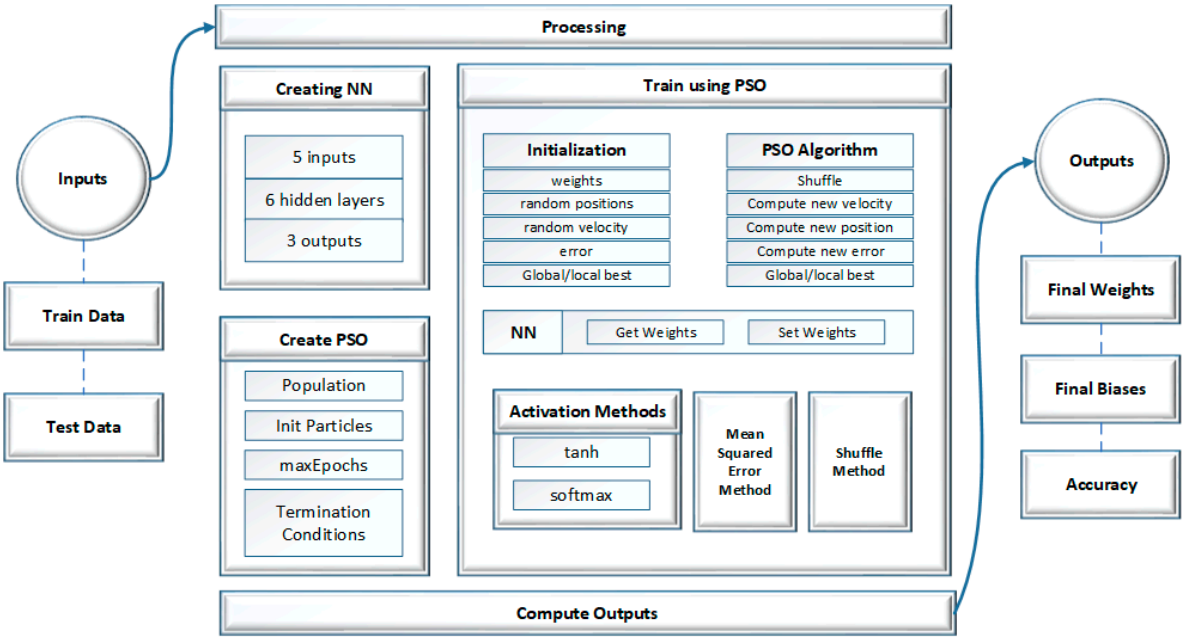

Figure 4. Configurations diagram.

$$\tanh x = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{4}$$

$$\text{Softmax}(X_i) = \frac{Exp(X_i)}{\sum_{j=0}^{k} Exp(X_j)} \tag{5}$$

$$MSE = \frac{\sum_{i=0}^{n} (X_i - X'_i)}{n} \tag{6}$$

3.4. Modified PSO-NN

In this sub-section, we present two modified versions of the typical PSO-NN; the first is regeneration-based PSO-NN (R-PSO-NN) where we consider the distance and performance of particles, and the second is velocity boost based PSO-NN (VB-PSO-NN) where we consider the inertia weight for updating the next velocity.

3.4.1. Re-Generation Based PSO-NN (R-PSO-NN)

In re-generation based PSO-NN, the first step is cluster-based regeneration. In cluster-based regeneration, a close cluster of particles is found then half of the particles from the cluster are moved to some other position where no such cluster exists (Figure 5). The distances between the particles position arrays are calculated using Euclidean distance (Equation (7), [58]) to identify the clusters.

$$Euclidean\ Distance\ (x,\ y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{7}$$

Firstly, Euclidean distance for all particles is calculated and then, based on Euclidean distance, the clusters are identified. A cluster is identified if more than three particles are found close to each other with a distance between them less than a threshold distance. The threshold distance is the minimum inter-particle distance defined as below:

$$Inter\ Particle\ Distance\ (IPD) = c \times NumParticles \tag{8}$$

where, IPD is the minimum distance threshold between two particles, c is a constant value for limiting inter-particle distance; c = 0.15; and $NumParticles$ is the total number of particles in the population.

The constant c is set after experimentation with varying values. Particles having less than the defined threshold distance between them will count as a cluster and be considered for re-generation.

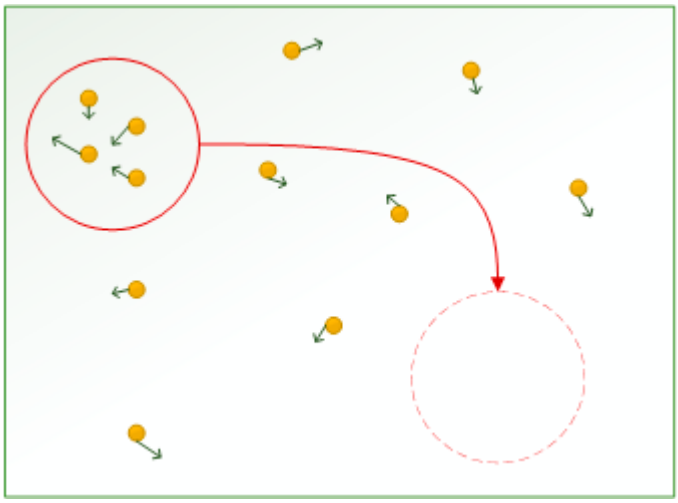

Figure 5. Particles' re-generation based on a cluster.

Before re-generating the particle forming a closed cluster, their past performance is checked. Particles in a cluster are not re-generated if they are all contributing to finding the global optimal value; they are moved only when stuck in local optima. Figure 6 elaborates the steps to find particle clusters and re-generate particles. Clusters are made based on the distance between them and the duplicate clusters (if any) are removed. For each particle P_i, its last updated pbest value is checked. If the last update in the pbest is found to be 15 iterations ago from current iteration, then it is added to the list of particles eligible to be re-generated (R_List). If the particles added to the re-generation list are about 90% or more of the total, then all the particles must be re-generated as they are stuck in local optima. Otherwise, half of the particles from the list are re-generated randomly to new positions.

If no cluster-based re-generation is made, then the possibility for performance-based regeneration is evaluated; for which each particle's pbest value's track is maintained. A counter re-generation threshold (RT) is maintained until the particle has no improvement in its pbest then it must be re-generated to a new position. At each epoch, it checks whether the particle's pbest value is improved or not. If pbest is improved with consecutive epochs, then it is not re-generated. Otherwise, if the particle's pbest value does not change over a set number of iterations RT and also it is not the current global best, then the particle is believed to be stuck in local optima.

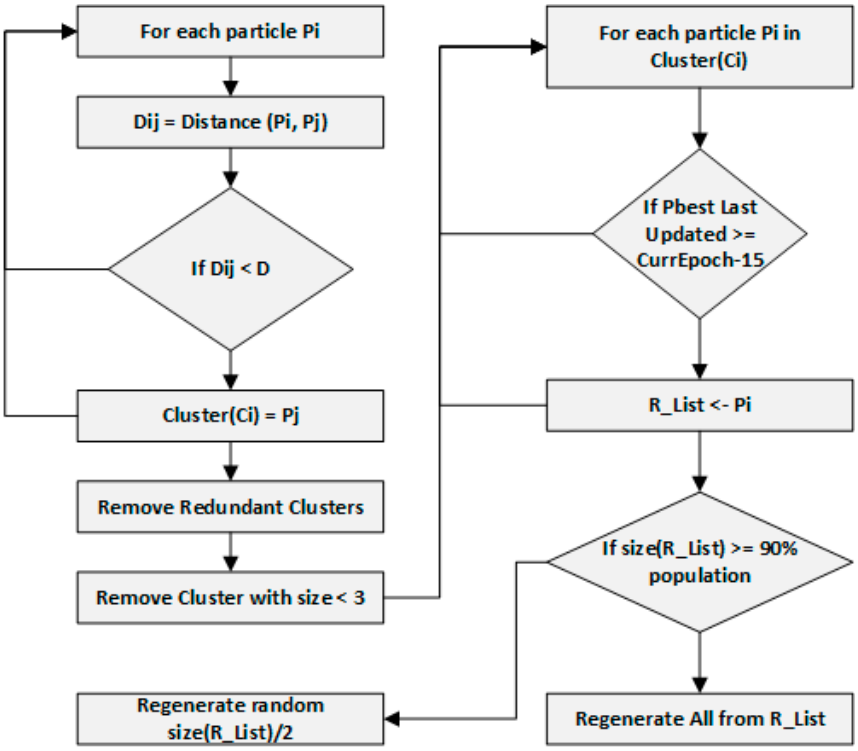

Figure 6. Re-generation of particles in a closed cluster.

3.4.2. Velocity Boost-Based PSO-NN (VB-PSO-NN)

In velocity boost-based PSO-NN we update the particle's velocity using a new inertia weight value, occasionally. Initially, we keep our inertia weight constant (w = c = 0.729) as given in [37]. After each epoch in the PSO-NN learning algorithm, we examine the change in the particle's pbest, maintain a counter for each particle and have a defined threshold as a velocity boost threshold (VBT); reaching VBT, if there's no improvement in the particle's pbest then the new inertia weight is used to update the velocities in that epoch. The threshold VBT is further finalized after extensive experimentation with different values. We use constant inertia weight [40] shown in Equation (9) and random inertia weight [41] shown in Equation (10) to tailor our new inertia weight as shown in Equation (11):

$$Contant\ Inertia\ Weight = c1 = 0.7 \tag{9}$$

$$Random\ Inertia\ Weight = 0.5 + \frac{Rand()}{2} \tag{10}$$

$$New\ Inertia\ Weight\ (W_i) = c1 + \frac{Rand()}{3} \tag{11}$$

In our tailored inertia weight equation; $c1 = 0.801$, which is selected after playing around with the variations of constant and inertia weight combinations.

Figure 7 shows the detailed flowchart for our proposed modifications of the PSO-NN algorithm. Firstly, the neural network and PSO are generated and initialized. PSO initializes velocities, positions, pbest and gbest and error for each particle in the population. The maximum number of epochs (MaxEpochs) is the total number of iterations for finely optimizing the neural network's weights using PSO. In each epoch, first the cluster particle regeneration possibility is checked. The detailed flow of cluster-based particle regeneration is given in Figure 6. If cluster-based particle regeneration is not done, then it checks for the performance-based velocity boost first, comparing the last update of pbest count with VBT. If the last update count equals VBT, then for the current iteration the particle's next

velocity is calculated using new inertia weight (Equation (8)). After updating velocity using the new inertia weight, the inertia weight is reset for the next iterations. The new inertia weight is only used when any particle's last update counter condition is met. If the inertia weight isn't updated too, then it checks for performance-based particle regeneration. The particle's pbest last update count is compared to RT; if the last update count equals RT then, assuming the particle to be stuck at position with further improvement in pbest possible, the particle is regenerated to a new position. Upon re-generation, the re-generation counter for the particle is reset.

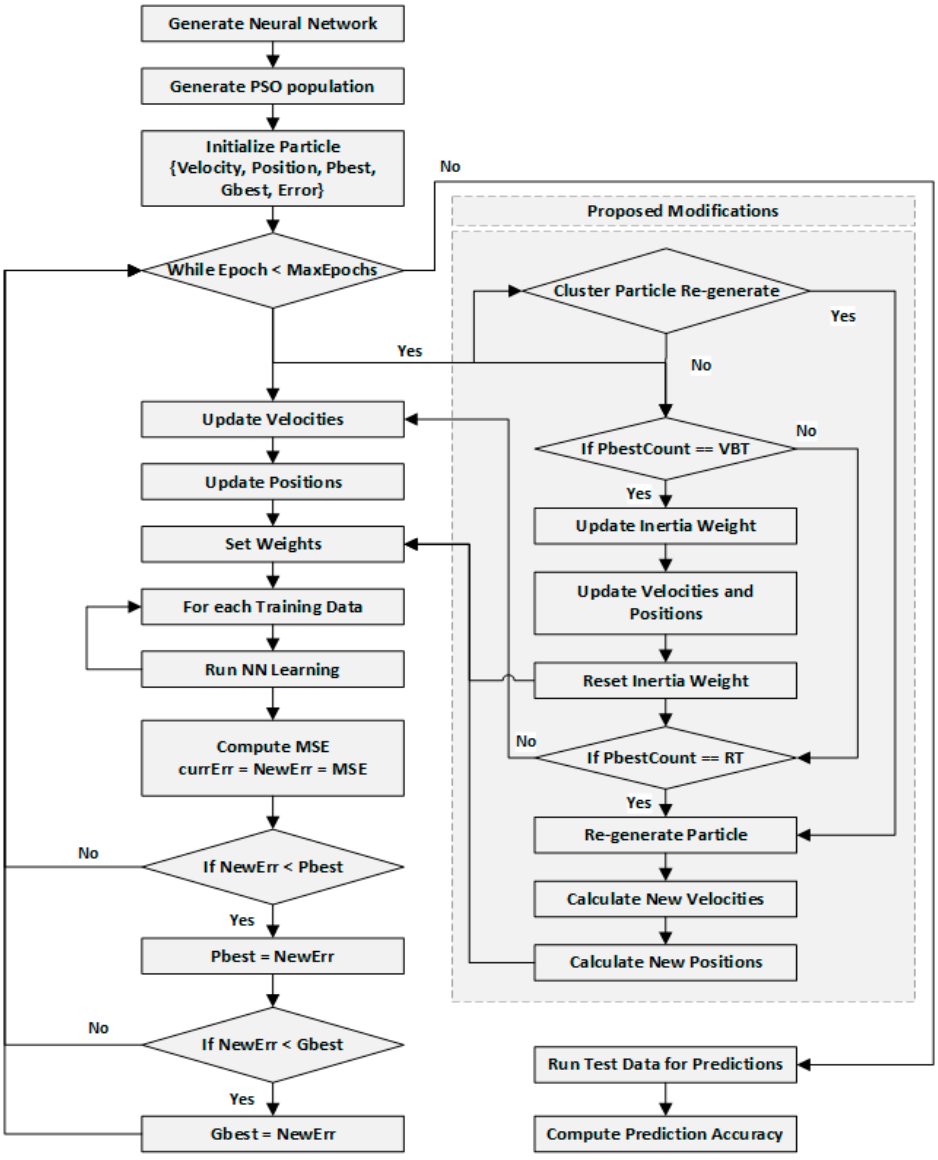

Figure 7. PSO-NN based on regeneration and velocity boost.

We chose to use the inertia weight on a specific condition only, as moving a particle's position too frequently can also be a hurdle in finding the optimal solution. Hence, we try to give a boost to the velocity when a particle's performance becomes relatively slow after a certain number of iterations and then reset it back to the old velocity pace. This procedure proved to be more effective. In our experimentation, we tested different values for the particle's re-generation threshold (RT) and inertia velocity boost threshold (VBT). These two thresholds are supposed to improve particle pbest once it is believed to be stuck in the local optima. After multiple comparisons among different input values, we

finalized VBT = 11 and RT = 25, as these two values brought maximum improvements in the prediction accuracies for building energy consumption.

4. Data Set and Experimental Setup

In this section we present our dataset and experimental environment. We look in depth at the data and try to find helpful trends in it also.

4.1. Data Set

Figure 8 elaborates the data collection phase. The data set is gathered from four residential buildings of Seoul, South Korea, from January to December 2010.

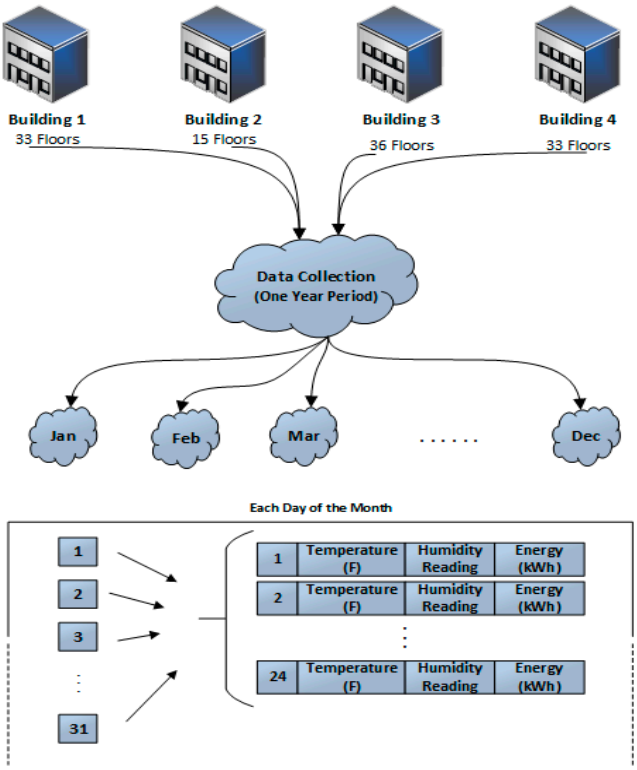

Figure 8. Data collection—Year 2010.

We have energy consumption data of four residential buildings in Seoul, South Korea. The available data comprises hourly temperature, relative humidity and energy consumption readings. The energy consumption readings are collected for each available floor and also overall consumption per hour for each building is added up. The buildings are constructed of reinforced concrete, the same as most of the residential buildings in South Korea. The first building has 33 floors (394 ft. tall), the second building has 15 floors (183 ft. tall), the third building has 36 (440 ft. tall) floors, and the fourth building has 33 floors (394 ft. tall). The primary source of energy used in these buildings is electricity, which is used for lighting, cooking, cooling, washing, drying, entertainment etc. Space heating consumption is a combination of electricity via the use of electric heaters and gas via floor heating. Our collected data only considers total electricity consumption per floor of each building. The energy consumption in many aspects is related to the weather of a region. Seoul has wet and humid summers, dry and cold winters, a windy spring and mild autumn. The average range of temperature in winter is [−3 °C, 5 °C], in summer is [12 °C, 30 °C], in spring is [5 °C, 15 °C] and in autumn is [6 °C, 14 °C] [59].

Figure 9 shows an overview of the collected data. We have plotted the first 60 h (the first 2.5 days of January) sample readings in the graph. The *x*-axis shows the sample reading number and the *y*-axis shows the values recorded for each hour. Figure 9a shows the energy consumption readings, Figure 9b shows the readings for relative humidity and Figure 9c shows the sample readings for temperature. We have relative humidity, temperature and energy consumption readings for the four buildings. The temperature range for the plotted sample days is between [−13, 0]. Since the plotted days are at the peak of winter, the temperature readings are all negative, reaching −13, but the overall average of winter-collected data is same as mentioned above. The relative humidity readings for the plotted sample are in the range of [30, 90]. Energy consumption levels for building 1 and building 2 are between 20 kW to 60 kW, while the energy consumption levels for building 3 and building 4 are between 40 kW to 100 kW.

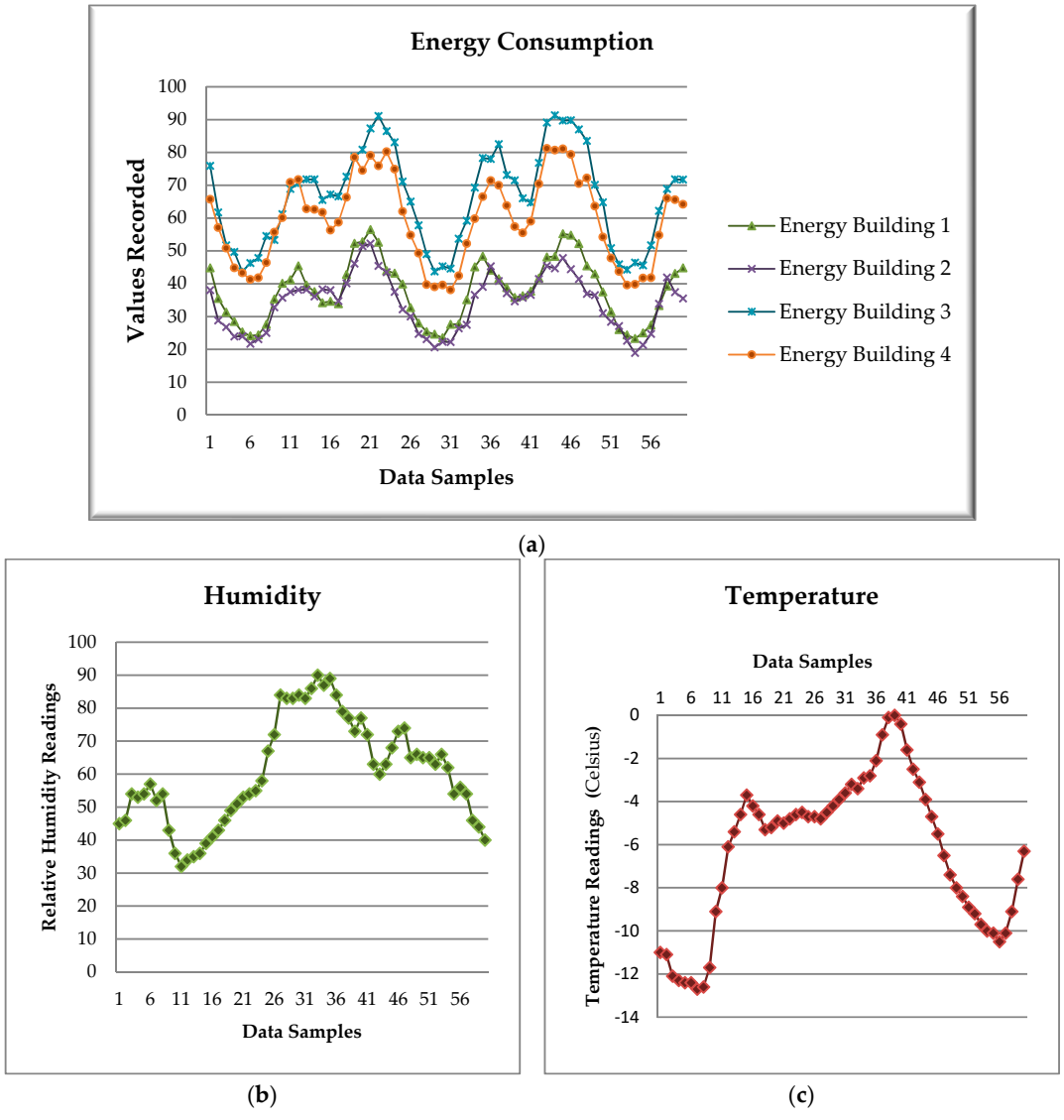

Figure 9. Collected data overview. (**a**) Sample data for energy consumption; (**b**) sample data for relative humidity; (**c**) sample data for temperature.

Figure 10 shows one day's sample data for hourly energy consumption of all the four buildings. The graph chart divides the energy consumption into five classes for a better visual understanding. We can observe that most of the hourly consumption is in the green class i.e., 40–60 kWh.

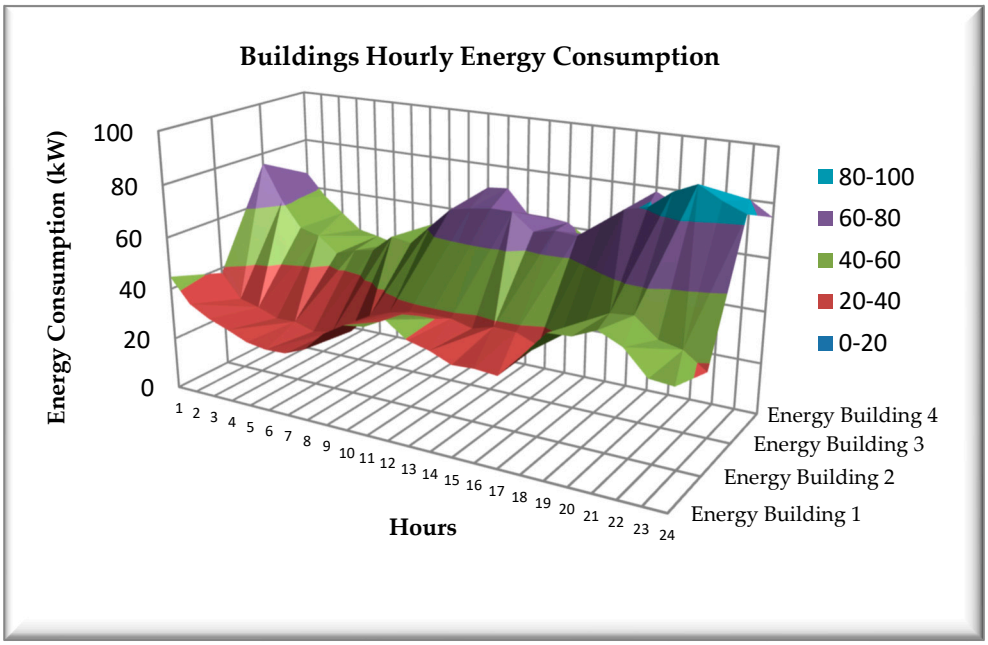

Figure 10. Hourly energy consumption for buildings—1 day.

4.2. Experimental Setup

The implementation is performed on an Intel® Core™ i5-4570 CPU (Intel, Santa Clara, CA, USA) at 3.20 GHz with 8 GB installed memory (DDR3, Samsung, Seoul, South Korea) and 64-bit operating system (10.0.17134 Build 17134, Microsoft, Redmond, WA, USA). The implementation platform is Visual Studio 2017 (Microsoft, Redmond, WA, USA) and the implementation language is C#.

The available data as described in Section 4.1 is residential building data spanning a one-year time period. We have divided our data into 16 classes for hours of the day, each containing two hours, and we aim to predict the energy consumption of a smart building every two hours. The data values are normalized before feeding to the prediction algorithm as input. We have 5 inputs to our system as temperature, humidity, energy consumption, hour of the day and day of the week. We aim to predict the hourly energy consumption of a building. The data is divided as 75% for the training phase and 25% for the testing phase.

5. Results Analysis

In this section, we present results analysis and comparisons between our implemented NN and PSO-NN algorithms with our proposed modified PSO-NN. For the purpose of detailed results analysis, we implemented our re-generation based (R-PSO-NN) and velocity boost-based (VB-PSO-NN) procedures separately. Hence, we make our comparisons between NN, PSO-NN, and the proposed modified versions. For the comparisons we consider the prediction accuracy in percentages in contrast with number of iterations, number of PSO populations, and re-generation threshold (RT) and velocity boost threshold (VBT) for the modified PSO-NN.

Figure 11 shows the output comparisons between the prediction accuracy of NN, PSO-NN, R-PSO-NN and VB-PSO-NN. Neural networks start with an accuracy of around 94.3% initially until 300 iterations and then rise to 99.08% accuracy at 400 iterations, with a maximum accuracy of 99.32%. PSO-NN achieves the accuracy of 98.04% at 100 iterations rising to an accuracy of 99.42% finally. R-PSO-NN achieves an accuracy of 99.13% at 100 iterations and reaches the maximum accuracy of 99.45%, beating both NN and PSO-NN. VB-PSO-NN starts with an accuracy of 98.77%, at 100 iterations and achieves maximum accuracy of 99.45% before reaching 200 iterations. We can observe that while PSO-NN performs better than NN, achieving high accuracy in less iteration, the modified versions of

PSO-NN beat that too. Modified versions of PSO-NN with their functionality of regenerating particles, stopping them being stuck at local optima, not only achieved higher accuracy but also with fewer iterations. R-PSO-NN took 700 epochs to reach its highest accuracy of 99.45% whereas VB-PSO-NN achieved the same accuracy in less than 200 epochs.

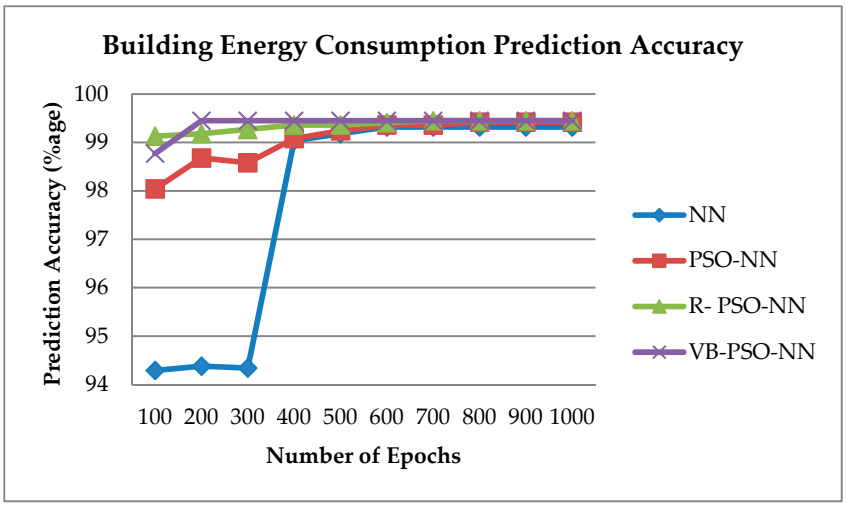

Figure 11. Prediction accuracy results comparison.

Figure 12 shows a close up view of the comparison between R-PSO-NN and VB-PSO-NN. In the figure, it is quite evident that although initially VB-PSO-NN starts slowly with accuracy lower than R-PSO-NN at 100 epochs, after that it drastically increases accuracy and crosses the maximum accuracy of R-PSO-NN within 120 epochs. Hence, for our given data of smart building energy consumption, the VB-PSO-NN approach is more effective and responsive for making the optimal predictions.

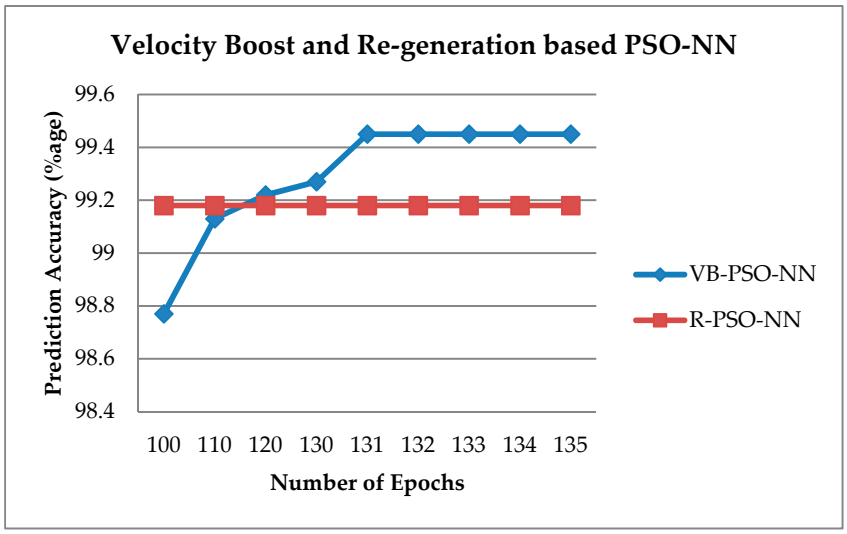

Figure 12. Comparison of re-generation based PSO-NN (R-PSO-NN) and velocity boost based PSO-NN (VB-PSO-NN).

5.1. Re-Generation Threshold (RT)

In the R-PSO-NN, we defined a threshold value for the regeneration of particles. In order to find the correct threshold value we played around with different thresholds in the range 15 to 30. We chose the window of [15, 30], considering that before 15 it is too soon to be changing a particle's position as it

needs some time to move around and we are also looking to minimize the number of iterations, which is why we consider 30 to be the ideal maximum value, after which the particle gets time to resettle in the new position for some minimum epochs.

Figure 13 shows the results of prediction accuracy of energy consumption data with the re-generation thresholds 15, 20, 25 and 30 in Figure 13a–d respectively. The results are calculated for 100 iterations each. We observe that, at re-generation threshold 15, maximum accuracy is 98.87%. The accuracies for threshold 20 went down to around 94% while again rising at threshold 25. At 25, maximum accuracy is achieved of 99.13% while after 25 the accuracy again tends to decrease. Hence, from the comparisons, we could conclude that 25 iterations is a most suitable number for regenerating the particles' positions for a quick search for an optimal solution.

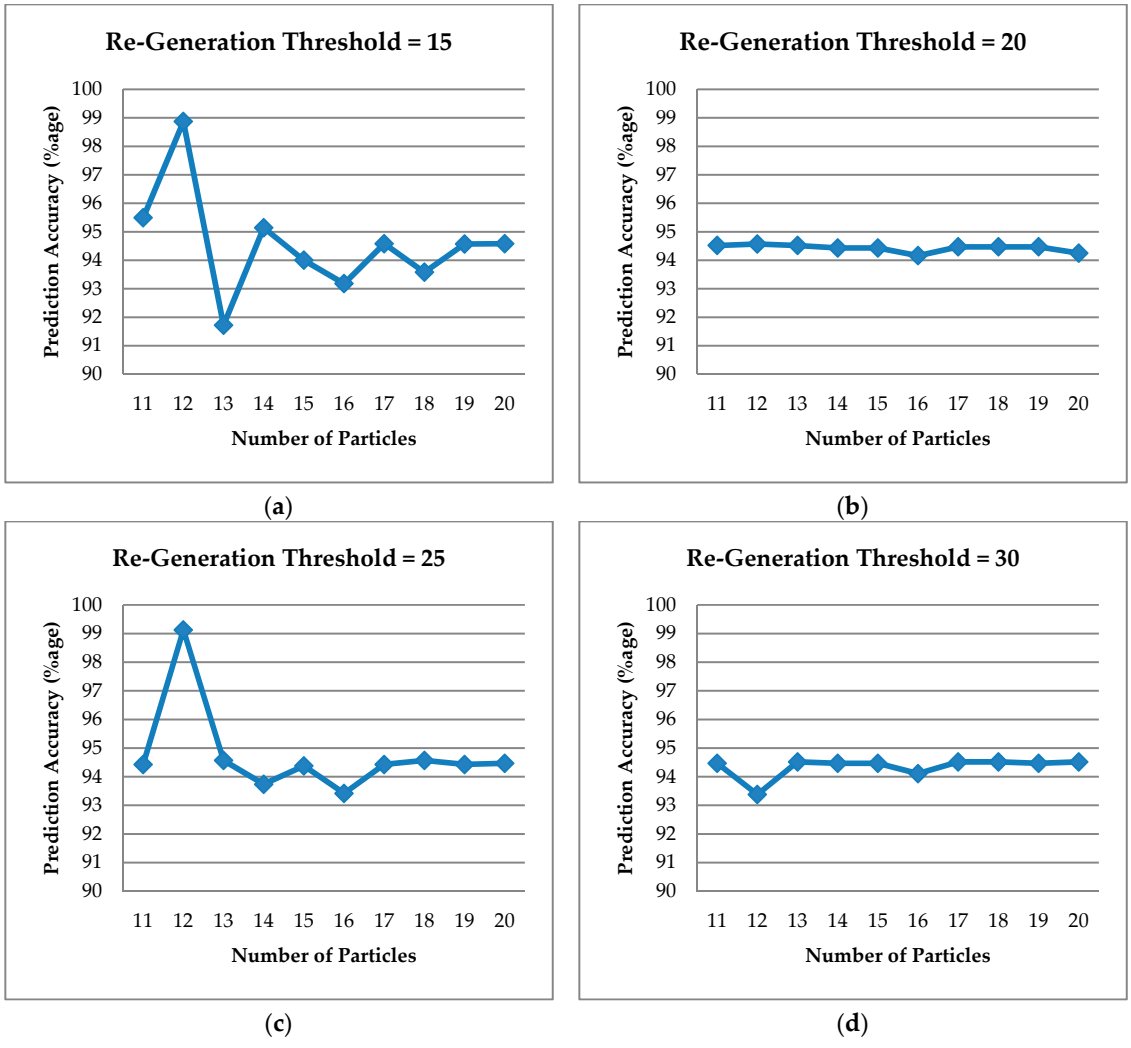

Figure 13. Searching optimal re-generation threshold (RT) for R-PSO-NN (**a**) with re-generation threshold 15; (**b**) with re-generation threshold 20; (**c**) with re-generation threshold 25; (**d**) with re-generation threshold 30.

5.2. Velocity Boost Threshold (VBT)

In order to get the optimal VBT value, we ran experiments with values ranging from 5 to 20. Figure 14 shows the output accuracies for VB-PSO-NN. From 5 to 10, we get an accuracy of 94.57%, at 11 we get a sudden rise to the accuracy of 99.45% while again after 11 the accuracy drops to 99.45 and keeps fluctuating a little bit until 20. After testing the VBT for these values, we can say that 11 is the right threshold value for VBT, as it gives maximum prediction accuracy.

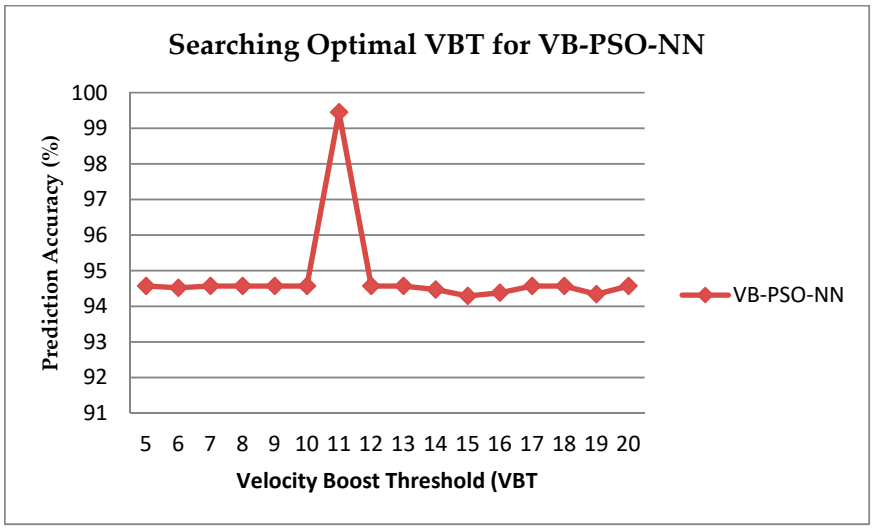

Figure 14. Energy consumption prediction.

5.3. Number of Particles and Epochs Comparison

We also compare our results to the number of particles in PSO. In Figure 13 above, we also observed that accuracy changes with the change in number of particles in the PSO population. In Figure 13, we noticed that higher accuracy is achieved with a particle population size of 12. Now we run all three PSO-NN, R-PSO-NN and VB-PSO-NN for a varying number of particles to see the effects on prediction accuracy (Figure 15). In the previous experiment, we observed a population size of 12 to be most productive for PSO-NN and R-PSO-NN. In Figure 15, we notice that for VB-PSO-NN, accuracy is higher with 17 numbers of particles in the PSO population, as the accuracy rose from 99.45% to 99.50%.

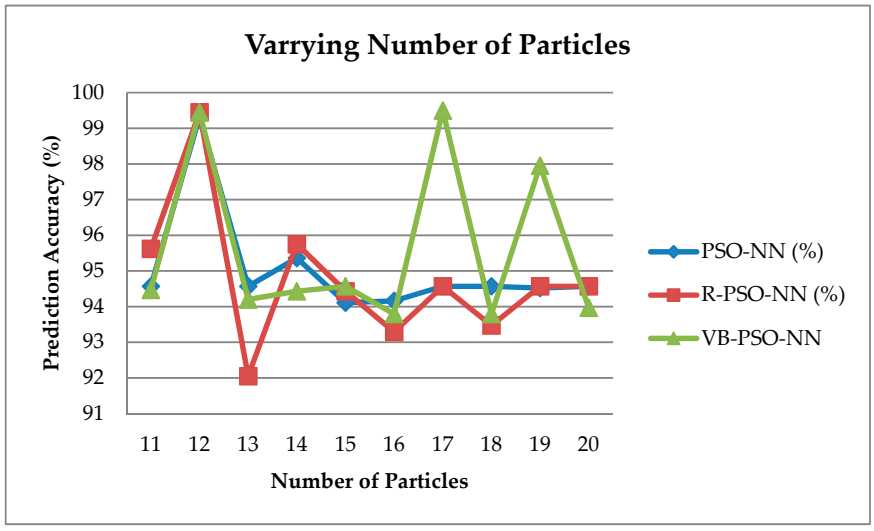

Figure 15. Change in accuracy with varying number of particles.

In Table 1, we present our maximum achieved accuracies for each of the NN, PSO-NN, R-PSO-NN and VB-PSO-NN. As discussed earlier, the results change with such parameters as the number of particles in the PSO population and the total number of epochs provided to run the training. Hence in the table below, we give our best accuracies with the number of particles and number of epochs required to achieve them. The table shows that VB-PSO-NN proves to be the best approach, as it not

only gives the maximum accuracy but also consumes the minimum number of epochs. While for the accuracy of 99.50%, VB-PSO-NN took 17 particles in the PSO population, even with a lesser PSO population of 12 it gives same maximum accuracy as R-PSO-NN, taking 133 epochs only.

Table 1. Best accuracy parameters comparison.

Algorithm	Number of Epochs	Number of Particles	Best Accuracy
ANN	600	-	99.32%
PSO-NN	800	12	99.42%
R-PSO-NN	700	12	99.45%
VB-PSO-NN	133	17	99.50%

6. Discussion

We have proposed a particle re-generation and velocity boost-based PSO-NN for residential electricity consumption prediction. The data we have used is from January to December 2010; one of the vital questions is the validity of the data gathered 8 years ago. In order to make sure that data is not outdated, we examine residential electricity trends from 2010 onwards. An increase of 7% is seen in the total electricity consumption of South Korea from January 2010 to January 2018 [60] and an increase of 1.2% is seen in the residential electricity consumption of South Korea from 2010 to 2016 [2]. Hence, we can consider our provided data still to be in context.

In order to validate our proposed model's efficiency, we chose ANN [61] and SVM [62] for comparisons; as the literature review indicates ANNs and SVMs to be the most widely used algorithms in building energy consumption predictions [14,15]. In this work, we implemented a hybrid PSO-NN algorithm for building energy predictions and proposed two modifications in it for improving its performance. PSO-NN is an optimized version of ANN. The performance of PSO-NN will be better than ANN as PSO-NN optimizes the weights of ANN. Hence, the results achieved by our two proposed modifications of PSO-NN improve the prediction accuracy even more, outperforming the performance of ANNs (Table 1). For further comparisons, we run the SVM algorithm on our input data. We have used MATLAB R2018a (The MathWorks, Inc., Natick, MA, USA) for training our data on the SVM model. The implementation of SVM in MATLAB automatically scales data into range of [−1, +1], thus we do not need to pre-process data any further. We trained our data on six different kernels as linear, cubic, quadratic, fine Gaussian, medium Gaussian and coarse Gaussian SVM. The results with the quadratic kernel (97.89%) and cubic kernel (97.89%) were best in comparison to all six SVM implementations but still lower in comparison to ANN, PSO-NN and our proposed R-PSO-NN and VB-PSO-NN (Figure 16).

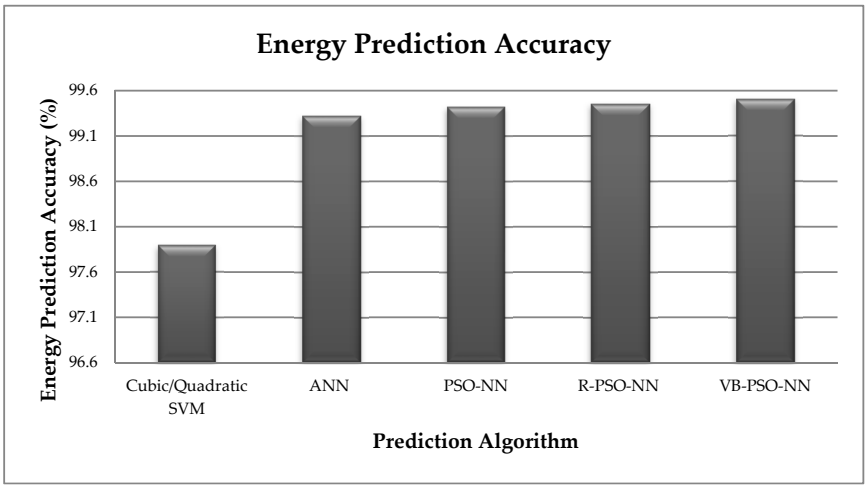

Figure 16. Comparison of final prediction accuracies.

The R-PSO-NN approach, while it increases accuracy and decreases the number of epochs, still is not as efficient as V-PSO-NN (Table 1 and Figure 16). One of the reasons could be the far re-generation of particles from possible solution positions most of the time. The particle re-generation involves a random factor and we are not fully controlling the re-generation position of the particle. While in V-PSO-NN the accuracy is higher with a drastic decrease in the number of epochs, we have carefully developed an equation for velocity boost after using multiple combinations of the equation constants and have fine-tuned the VBT value after experimenting with different threshold values. Hence, for V-PSO-NN, we can conclude that our fine-tuned parameter values allow the particle to make a quick move towards the right direction before it moves towards a non-solution direction or towards getting trapped in local optima.

Our main motivation in this work is to further improve the performance of PSO-NN for domains where ANN prediction algorithm is best suited. In the context of previous implementations of ANN/PSO-NN, the proposed modifications will bring an improvement in the results and can easily be incorporated into any ANN/PSO-NN algorithm without changing the core of the algorithm, resulting in a more efficient hybrid approach. The conducted study and proposed mechanisms for accurate and efficient energy prediction will be of benefit in terms of timely predicting upcoming energy demands for smart buildings and managing energy consumption schedules accordingly. Moreover, accurate energy consumption also plays a vital role in load-shifting from peak hours to semi-peak or off-peak hours. One of the key factors in managing and optimizing the energy consumption of smart buildings is the detailed availability of multi-dimensional data. True user behavior observations and energy consumption trends can help us focus and set the right direction for finding solutions and optimizing energy. Hence, in future, we strive to collect timely and advanced data for smart buildings and explore various aspects of smart buildings' energy consumption such as user activity-based consumption trends, peak and off-peak hours consumption patterns etc.

7. Conclusions

In this work, we aim to address the issue of efficient energy predictions in order to optimize energy consumption in the smart buildings sector. In order to achieve this, we focused on prediction algorithms and their respective accuracy in different areas. We applied PSO-NN, a combined approach of using neural networks for learning along with one of the widely used optimization algorithms i.e., the particle swarm optimization (PSO) algorithm. The proposed algorithms were implemented on a dataset of smart buildings in Seoul, South Korea. This data was collected for a period of one year. Jn recent years neural networks and advanced machine learning techniques have been used extensively for prediction related problems, and so it is high time for advanced and hybrid approaches to be used in order to produce more accurate and optimized results. In this study, we have proposed two variants of PSO-NN, re-generation based PSO-NN (R-PSO-NN) and velocity boost based PSO-NN (VB-PSO-NN). Both the proposed approaches are experimented on using the aforementioned data set. The VB-PSO-NN approach showed higher performance in terms of the accuracy of results as compared with NN, PSO-NN and R-PSO-NN. Although the accuracy achieved through R-PSO-NN was slightly lower than VB-PSO-NN, it outperformed NN and PSO-NN. The results obtained in the form of prediction accuracy of energy consumption for smart buildings has given us the confidence to explore more dimensions of data and continue to build prediction models for smart buildings using PSO-NN and its variants.

Author Contributions: S.M. designed the PSO-NN-based prediction methodology for energy prediction in residential smart buildings, performed the experiments, analyzed the data and wrote the paper. D.K. managed the data curation process, conceived the overall idea of energy prediction in residential smart buildings, and supervised this work. Both authors contributed to this paper.

Acknowledgments: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (2014-1-00743) supervised by the IITP (Institute for Information & communications Technology Promotion), and this work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government

(MSIT) (No. 2017-0-00756, Development of interoperability and management technology of IoT system with heterogeneous ID mechanism). Any correspondence related to this paper should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr.

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

References

1. Global Demand for Energy Will Peak in 2030, Says World Energy Council. Available online: https://www.theguardian.com/business/2016/oct/10/global-demand-for-energy-will-peak-in-2030-says-world-energy-council (accessed on 5 April 2018).
2. Energy Information Korea. 2016. Available online: https://www.keei.re.kr/web_keei/en_publish.nsf/XML_Portal_New/B9C4089854585599492580BF0009503E/$file/EnergyInfo2016.pdf (accessed on 4 May 2018).
3. International Energy Outlook. 2017. Available online: https://www.eia.gov/outlooks/ieo/pdf/0484(2017).pdf (accessed on 5 April 2018).
4. Deru, M.; Torcellini, P. *Source Energy and Emission Factors for Energy Use in Buildings (Revised)*; No. NREL/TP-550-38617; National Renewable Energy Laboratory (NREL): Golden, CO, USA, 2007.
5. The Internet of Things: Making the Most of the Second Digital Revolution. Available online: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/409774/14-1230-internet-of-things-review.pdf (accessed on 5 April 2018).
6. Korea's Smart Homes Need New Lessons. Available online: http://koreajoongangdaily.joins.com/news/article/article.aspx?aid=3029201 (accessed on 5 April 2018).
7. Korea Energy Agency. 2014. Available online: http://www.energy.or.kr/web/kem_home_new/energy_issue/mail_vol22/pdf/publish_05_201507.pdf (accessed on 5 April 2018).
8. 4 Ways the Internet of Things Is Saving Energy, Water & Money. Available online: http://dadolabs.com/energyefficiency/ (accessed on 5 April 2018).
9. Amasyali, K.; El-Gohary, N. Building lighting energy consumption prediction for supporting energy data analytics. *Procedia Eng.* **2016**, *145*, 511–517. [CrossRef]
10. Hu, Y.-C. Electricity consumption prediction using a neural-network-based grey forecasting approach. *J. Oper. Res. Soc.* **2017**, *68*, 1259–1264. [CrossRef]
11. Israr, U.; Ahmad, R.; Kim, D. A Prediction Mechanism of Energy Consumption in Residential Buildings Using Hidden Markov Model. *Energies* **2018**, *11*, 358. [CrossRef]
12. Perez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
13. Zhao, H.-X.; Magoules, F. A review on the prediction of building energy consumption. *Renew. Sustain. Energy Rev.* **2012**, *16*, 3586–3592. [CrossRef]
14. Ahmad, A.S.; Hassan, M.Y.; Abdullah, M.P.; Rahman, H.A.; Hussin, F.; Abdullah, H.; Saidur, R. A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renew. Sustain. Energy Rev.* **2014**, *33*, 102–109. [CrossRef]
15. Fumo, N. A review on the basics of building energy estimation. *Renew. Sustain. Energy Rev.* **2014**, *31*, 53–60. [CrossRef]
16. Nogales, F.J.; Contreras, J.; Conejo, A.J.; Espínola, R. Forecasting next-day electricity prices by time series models. *IEEE Trans. Power Syst.* **2002**, *17*, 342–348. [CrossRef]
17. Massimiliano, M.; Caputo, P.; Costa, G. Paradigm shift in urban energy systems through distributed generation: Methods and models. *Appl. Energy* **2011**, *88*, 1032–1048.
18. Adel, A.; Hemmati, M.; Abdoos, A.A. Short term load forecasting using a hybrid intelligent method. *Knowl. Based Syst.* **2015**, *76*, 139–147.
19. Gallego, C.; Costa, A.; Cuerva, A. Improving short-term forecasting during ramp events by means of regime-switching artificial neural networks. *Adv. Sci. Res.* **2011**, *6*, 55–58. [CrossRef]
20. Tutun, S.; Chou, C.-A.; Erdal, C. A new forecasting framework for volatile behavior in net electricity consumption: A case study in Turkey. *Energy* **2015**, *93*, 2406–2422. [CrossRef]
21. Lu, X.; Lu, T.; Kibert, C.J.; Viljanen, M. Modeling and forecasting energy consumption for heterogeneous buildings using a physical-statistical approach. *Appl. Energy* **2015**, *144*, 261–275. [CrossRef]

22. Swan, L.G.; Ugursal, V.I. Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renew. Sustain. Energy Rev.* **2009**, *13*, 1819–1835. [CrossRef]
23. Al-Homoud, M.S. Computer-aided building energy analysis techniques. *Build. Environ.* **2001**, *36*, 421–433. [CrossRef]
24. Wang, S.W.; Xu, X.H. Simplified building model for transient thermal performance estimation using GA-based parameter identification. *Int. J. Therm. Sci.* **2006**, *45*, 419–432. [CrossRef]
25. Hyojoo, S.; Kim, C. Short-term forecasting of electricity demand for the residential sector using weather and social variables. *Resour. Conserv. Recycl.* **2017**, *123*, 200–207.
26. Yang, J.; Rivard, H.; Zmeureanu, R. On-line building energy prediction using adaptive artificial neural networks. *Energy Build.* **2005**, *37*, 1250–1259. [CrossRef]
27. Karatasou, S.; Santamouris, M.; Geros, V. Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy Build.* **2006**, *38*, 949–958. [CrossRef]
28. Wong, S.L.; Wan, K.K.W.; Lam, T.N.T. Artificial neural networks for energy analysis of office buildings with daylighting. *Appl. Energy* **2010**, *87*, 551–557. [CrossRef]
29. Dong, B.; Cao, C.; Lee, S.E. Applying support vector machines to predict building energy consumption in tropical region. *Energy Build.* **2005**, *37*, 545–553. [CrossRef]
30. Kavaklioglu, K. Modeling and prediction of Turkey's electricity consumption using support vector regression. *Appl. Energy* **2011**, *88*, 368–375. [CrossRef]
31. Zhao, H.-X.; Magoulès, F. Feature selection for predicting building energy consumption based on statistical learning method. *J. Algorithms Comput. Technol.* **2012**, *6*, 59–78. [CrossRef]
32. Tso, G.K.; Yau, K.K. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy* **2007**, *32*, 1761–1768. [CrossRef]
33. Kalogirou, S.A.; Bojic, M. Artificial neural networks for the prediction of the energy consumption of a passive solar building. *Energy* **2000**, *25*, 479–491. [CrossRef]
34. Peram, T.; Veeramachaneni, K.; Mohan, C.K. Fitness-distance-ratio based particle swarm optimization. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS'03, Indianapolis, IN, USA, 24–26 April 2003; pp. 174–181.
35. Cai, X.; Cui, Y.; Tan, Y. Predicted modified PSO with time-varying accelerator coefficients. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 50–60. [CrossRef]
36. Chen, X.; Li, Y. A modified PSO structure resulting in high exploration ability with convergence guaranteed. *IEEE Trans. Syst. Man Cybern. Part B* **2007**, *37*, 1271–1289. [CrossRef]
37. Sedghi, M.; Aliakbar-Golkar, M.; Haghifam, M.R. Distribution network expansion considering distributed generation and storage units using modified PSO algorithm. *Int. J. Electr. Power Energy Syst.* **2013**, *52*, 221–230. [CrossRef]
38. Ghomsheh, V.S.; Shoorehdeli, M.A.; Teshnehlab, M. Training ANFIS structure with modified PSO algorithm. In Proceedings of the IEEE Mediterranean Conference on Control & Automation, MED'07, Athens, Greece, 27–29 June 2007; pp. 1–6.
39. Wang, S.; Watada, J. A hybrid modified PSO approach to VaR-based facility location problems with variable capacity in fuzzy random uncertainty. *Inform. Sci.* **2012**, *192*, 3–18. [CrossRef]
40. Cui, G.; Qin, L.; Liu, S.; Wang, Y.; Zhang, X.; Cao, X. Modified PSO algorithm for solving planar graph coloring problem. *Prog. Nat. Sci.* **2008**, *18*, 353–357. [CrossRef]
41. Chau, K.W. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *J. Hydrol.* **2006**, *329*, 363–367. [CrossRef]
42. Chatterjee, S.; Sarkar, S.; Hore, S.; Dey, N.; Ashour, A.S.; Balas, V.E. Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings. *Neural Comput. Appl.* **2017**, *28*, 2005–2016. [CrossRef]
43. Song, Y.; Chen, Z.; Yuan, Z. New chaotic PSO-based neural network predictive control for nonlinear process. *IEEE Trans. Neural Netw.* **2007**, *18*, 595–601. [CrossRef] [PubMed]
44. Chau, K.W. Application of a PSO-based neural network in analysis of outcomes of construction claims. *Autom. Constr.* **2007**, *16*, 642–646. [CrossRef]
45. Lu, W.Z.; Fan, H.Y.; Lo, S.M. Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong. *Neurocomputing* **2003**, *51*, 387–400. [CrossRef]

46. Ahmadi, M.A. Neural network based unified particle swarm optimization for prediction of asphaltene precipitation. *Fluid Phase Equilib.* **2012**, *314*, 46–51. [CrossRef]
47. Gordan, B.; Armaghani, D.J.; Hajihassani, M.; Monjezi, M. Prediction of seismic slope stability through combination of particle swarm optimization and neural network. *Eng. Comput.* **2016**, *32*, 85–97. [CrossRef]
48. Sheikhan, M.; Mohammadi, N. Time series prediction using PSO-optimized neural network and hybrid feature selection algorithm for IEEE load data. *Neural Comput. Appl.* **2013**, *23*, 1185–1194. [CrossRef]
49. McCulloch, W.; Walter, P. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
50. Minsky, M.; Papert, S. *Perceptrons: An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 1969.
51. Artificial Neural Networks as Models of Neural Information Processing | Frontiers Research Topic. Available online: https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing (accessed on 30 March 2018).
52. Artificial Neuron Output. Available online: https://en.wikipedia.org/wiki/Artificial_neuron (accessed on 4 May 2018).
53. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
54. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
55. Hyperbolic Function. Available online: https://en.wikipedia.org/wiki/Hyperbolic_function (accessed on 4 May 2018).
56. Softmax Function. Available online: https://en.wikipedia.org/wiki/Softmax_function (accessed on 4 May 2018).
57. Mean Squared Error Function. Available online: https://en.wikipedia.org/wiki/Mean_squared_error (accessed on 4 May 2018).
58. Euclidean Distance. Available online: https://en.wikipedia.org/wiki/Euclidean_distance (accessed on 4 May 2018).
59. Seoul Average Weather. Available online: https://www.ncdc.noaa.gov/ (accessed on 4 May 2018).
60. Korea Electricity Consumption. Available online: https://www.ceicdata.com/en/korea/electricity-generation-and-consumption/electricity-consumption (accessed on 4 May 2018).
61. Bebis, G.; Georgiopoulos, M. Feed-forward neural networks. *IEEE Potentials* **1994**, *13*, 27–31. [CrossRef]
62. Krammer, K.; Singer, Y. On the algorithmic implementation of multi-class SVMs. *Proc. JMLR* **2001**, *2*, 265–292.