An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings

Dac-Khuong Bui¹, Tuan Ngoc Nguyen¹, Tuan Duc Ngo¹*, H. Nguyen-Xuan²,³*

¹Department of Infrastructure Engineering, the University of Melbourne, Parkville, VIC 3010, Australia
²Center for Interdisciplinary Research in Technology, Ho Chi Minh City University of Technology (Hutech), Ho Chi Minh City, Vietnam
³Department of Architectural Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea

Abstract

In this study, a new hybrid model, namely the Electromagnetism-based Firefly Algorithm - Artificial Neural Network (EFA-ANN), is proposed to forecast the energy consumption in buildings. The model is applied to evaluate the heating load (HL) and cooling load (CL) using two given datasets. Each dataset was obtained by monitoring the effect of the façade system and dimensions of the building, respectively, on energy consumption. The performance of EFA-ANN is validated by comparing the obtained results with other methods. It is shown that EFA-ANN provides a faster and more accurate prediction of HL and CL. A sensitivity analysis is conducted to identify the impact of each input on the energy performance of the building. From the results of this study, it is evident that EFA-ANN can assist civil engineers and construction managers in the early designs of energy-efficient buildings.

Keywords: Electromagnetism-based firefly algorithm; artificial neural network; machine learning; energy consumption.

* Corresponding author. Email address: dingo@unimelb.edu.au (Tuan Duc Ngo) and ngx.hung@hutech.edu.vn (H. Nguyen-Xuan)
1. Introduction

There are three major economic sectors in the world, namely energy, including transportation, industry, and building [1]. A substantial share of global energy is consumed by buildings, which is expected to increase to 32.4% by 2040 [2]. In participating nations of the Organization for Economic Cooperation and Development (OECD), including Australia, New Zealand, United Kingdom and the United States, energy consumption in buildings has grown by 1.5% per year from 2012 to 2040. In non-OECD nations, being mostly developing countries, the growth rate is 2.1% per year in the same period [3]. Therefore, the development of energy efficient building systems is essential and consequently, many efforts have been devoted to this area [4, 5].

In the majority of cases, heating and cooling energy demands mostly account for building energy consumption [6]. Therefore, the early prediction and reduction of heating and cooling loads plays a vital role in designing an energy-efficient building. This provides designers with access to various building designs or Heating Ventilation and Air Conditioning (HAVC) system control options to find the optimal solution for reducing energy consumption in buildings. For instance, Zemella et al. optimized the design of façades of energy-efficient buildings by making early predictions of energy consumption due to heating, cooling, and lighting [7]. Also, Magnier and Haghighat applied a method to predict the energy consumption due to heating, cooling, and fan systems, and optimized the building design based on these predictions [8]. In another study, Ferreira et al. proposed a model-based predictive control methodology to control the HAVC system in a building and reported savings of around 50% in energy consumption [9]. Ghahramani et al. provided a systematic approach to optimize the setpoint and deadband parameter of the HVAC system by pre-calculating the energy
consumption of HVAC [10]. Therefore, the early prediction of heating and cooling loads is critical to reducing the total energy consumption of a building.

However, many aspects, including temperature, sunlight equipment, occupant behavior, wall materials, glazing area, surface, height and volume of the building, have interactions between the overall energy requirements of the building [11-13]. For instance, Ihara et al. concluded that all façade properties including solar reflectance, U-value, solar heat gain coefficient (SHGC) have different effects on the energy efficiency of buildings. They suggested that the reduction of SHGC is the most effective method for reducing energy consumption [14]. In contrast, Liu et al. showed that reducing SHGC does not improve the energy efficiency of a building without an appropriate U-value [15]. Many other factors affect the total energy consumption in a building, including weather conditions, building dimensions, and the behavior of occupants. Therefore, it is quite challenging to calculate energy consumption given that all the above parameters and their interactions should be considered.

There are three main categories in building energy assessment, including engineering calculations, numerical simulations, and machine learning. The first approach focuses on using physical laws to calculate the energy consumption of an entire building. This approach is only suitable for preliminary analysis as it is mathematically intensive. In this research, the numerical simulation approach was used to simulate the energy performance of a building and overcome the limitations of engineering calculations. Several building energy simulation programs, including Energy Plus, DOE-2, Window, Autodesk Ecotect, TRNSYS, and eQUEST, were used to simulate and predict energy consumption of a building. However, this method employs physics-based simulations, which are often time-consuming and resource-intensive, and the complexity
and demand increase with the size and complexity of the project [16]. Also, the optimization process must be manual based on user experience [17]. In many cases, energy models cannot reflect the actual performance of a building in reality as they lack the required details and need to make simplifications [18]. Thus, the conventional procedures are often unsatisfactory for designing a façade because they are mainly based on a particular design condition and the experience of experts. Accordingly, a significant development of an advanced approach is required.

To this effect, machine learning (ML), was proposed to design energy-efficient building services. This method applies a learning process to infer a relationship between the building data and energy consumption of buildings. In recent years, several advantages of ML have been demonstrated over conventional approaches [19, 20]. Consequently, many ML techniques have been applied to solve energy problems [21-23]. Robinson et al. successfully applied various ML techniques to predict the energy consumption of a commercial building based on various building features [21]. Ahmad et al. proposed four ML approaches to forecast short, medium and long-term energy consumption in a building [22]. In another study, Chou and Tran used a hybrid ML model to estimate the energy consumption of residential householders with high accuracy [23]. ML can also be used to support several adaptive systems in a building [24, 25]. For example, Ghahramani et al. developed a novel adaptive hybrid metaheuristic algorithm based on ML and other smart components to optimize the energy of an HVAC system [24]. Moon et al. proposed an ML model for controlling the temperature of an adaptive double skin envelope [25].

Recently, among many ML techniques, an artificial neural network (ANN) has been widely used in optimizing the design of an energy-efficient building. Jin et al. used
an ANN-based thermal control logic model to optimize the initial conditions and heating
system operations in a building [26]. In another study, Wang et al. forecasted the dynamic
building cooling load by combining ANN and an ensemble model [27]. Chung et al.
proposed an ANN model to design a comfortable indoor thermal environment in an
energy-efficient manner [28].

Notwithstanding many advantages, the ANN model depends on several initial
parameters, including weights and biases [29]. Therefore, many studies were carried out
to improve the performance of ANN by combining it with optimization algorithms. For
example, Yam and Chow optimized the initial weights of ANN by using linear algebraic
methods [30]. Liu et al. enhanced the generalizations and accuracies of ANN by using an
ensemble method [31]. Also, Chang et al. [32] and Lee et al. [33] proposed a genetic
algorithm and a harmony search algorithm, respectively, to optimize the initial weights
of the ANN model.

Among many optimization algorithms, our approach is to propose an
Electromagnetism-based Firefly Algorithm (EFA), which is found to be an efficient
optimization tool that can solve complicated problems. For instance, Shammari et al.
combined the firefly algorithm (FA) and support vector machine model to predict the
heating load of a heating system [34]. Coelho and Mariani proposed FA and a Gaussian
distribution function to optimize the loading of a chiller for energy conservation [35].
Chu and Chang used the Electromagnetism Algorithm (EA) to solve the resource
allocation problem in stochastic networks [36]. However, both FA and EA still show
several disadvantages, including premature convergence and divergence [37-39], which
will be discussed in the following section. Therefore, in this study, we propose a new
approach based on a hybrid model of FA and EA, called EFA, which aims to enhance the capability of ANN for predicting the energy consumption of a building.

Section 2 and 3 describe the theory behind of EFA and ANN, respectively. Details of the proposed EFA-ANN are provided in Section 4. Section 5 introduces the two datasets used to train EFA-ANN. The proposed method is validated in Section 6. Concluding remarks and recommendations for future research are provided in Section 7.

2. Electromagnetism-based Firefly Algorithm – Artificial Neural Network Model

2.1 An electromagnetism-based firefly algorithm

The Electromagnetism-based Firefly Algorithm (EFA) is a new hybrid optimization algorithm that incorporates the advantages of the firefly algorithm (FA) and Electromagnetism Algorithm (EA). Firstly, FA, which was proposed by Yang [37], is a swarm intelligence method and is based on the flashing patterns and behavior of tropical fireflies. In FA, the brightness of a firefly is determined by an objective function. A firefly will tend to move closer to the brighter firefly and is not affected by a darker one. On the other hand, the EA was introduced by Birbil and Fang [39]. It imitates the attraction-repulsion mechanism to solve global optimization problems. In EA, all points converge to the highly attractive valleys and move further away from steeper hills. This idea is the same as the attraction-repulsion mechanism described by the theory of electromagnetism. However, better points are easily distracted if there are many worse points in the population. Also, EA is not effective in a local search because it randomly moves all points without any advanced technique.

In EFA, all fireflies are assumed to be magnetized, and the charge of a firefly depends on its objective value. Therefore, fireflies will move close to attractive fireflies and move away from the repulsive ones. Thus, all fireflies can contribute to the search
process and improve the performance of EFA. Based on their charges, all fireflies are ranked in each iteration, and a firefly only has attraction and repulsion by an attractive or repulsive neighboring firefly, respectively. In this way, fireflies are not distracted by many forces as in EA. Consequently, EFA can quickly find promising areas in the exploration phase and then perform local searches in these areas to find the optimal solution in the exploitation phase. Local searches in EFA are performed using Lévy flight inspired by FA.

2.1.1 Initialization

Firstly, EFA is initialized in the same manner as FA, when the coordinates of the initial population of fireflies are uniformly distributed between the corresponding lower and upper bounds as follows

$$x^k = lb_k + \text{rand}(0,1)(ub_k - lb_k)$$  \hspace{1cm} (1)

where $lb_k, ub_k$ are the lower and upper boundaries of the $k^{th}$ coordinate, respectively; $\text{rand}(0,1)$ is a random number following a normal distribution within [0,1]. This study uses a logistic map to improve the performance of EFA. The logistic map generates the initial coordinates of fireflies using Eq. (2). It can provide an initial diverse population and also reduce the probability of premature occurrence [40]:

$$x^r_{o+1} = ax^r_o \left(1 - x^r_o \right)$$  \hspace{1cm} (2)

where $x^r_o$ is a chaotic number at $\bar{r}$ iteration of the initialization process; $a$ is fixed at 4 [41].

2.1.2 Local search

EFA inherits the exploitation capacity of FA and EA for searching conducting a local search in the vicinity of each coordinate. In other words, the local search of EFA
follows Eq. (3) and the improvement of each firefly is sought by coordinate. In this way, the probability of finding a better point is increased.

\[ x_{i+1}^t = x_i^t + \alpha^t \varepsilon L(s) \]  

(3)

where \( \varepsilon \) is a vector of random numbers and is defined by Eq. (4) as follows:

\[ \varepsilon = rand - 1/2 \]  

(4)

where \( rand \) is a random number generated by a uniform distribution in \([0, 1]\).

In addition, \( \alpha^t \) is a trade-off constant at \( t \) iteration [37] and is calculated as follows:

\[ \alpha^t = \alpha_0 \theta^t \]  

(5)

where \( \alpha_0 \) and \( \alpha^t \) are the initial trade-off coefficient and the trade-off coefficient at the \( t \)th iteration, respectively; and \( \theta \) is the adaptive parameter \((0 < \theta < 1) \) [37].

Finally, \( L(s) \) is the Lévy distribution, which can be defined as follows:

\[ L(s) \sim s = \frac{u}{|v|^{1/\tau}} \]  

(6)

where \( s \) is a power-law distribution, \( \tau \) is an index, and \( v \) and \( u \) are calculated to follow a normal distribution as follows:

\[ v \sim N(0, \sigma_v^2) \]  

(7)

\[ u \sim N(0, \sigma_u^2) \]  

(8)

where:

\[ \sigma_v = 1 \]  

(9)

\[ \sigma_u = \left( \frac{\Gamma(1+\tau)\sin(\pi\tau/2)}{\Gamma[(1+\tau)/2]2^{(\tau-1)/2}} \right)^{1/\tau} \]  

(10)
where $\Gamma(z)$ is the Gamma function, which is determined by:

$$
\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt
$$

(11)

2.1.3 Movement

EFA benefits from the advantages of both EA and FA. Firstly, EFA has inspired the concept of attraction and repulsion forces from EA. However, EFA ranks all fireflies based on their objective function values and a firefly $i$ is only affected by the attraction $(F_g)$ and repulsion force $(F_r)$ from the next better $(i-1)$ and worse firefly $(i+1)$. In contrast, all fireflies in the population are considered in EA. In this way, EFA can mitigate the interference from many fireflies and quickly find the optimal solution. The movement of a magnetic firefly is written as:

$$
x_i^{t+1} = x_i^t + \left( x_{i-1}^t - x_i^t \right) \frac{F_g}{F_g} (1-\alpha) + \left( x_{i+1}^t - x_i^t \right) \frac{F_r}{F_r} \alpha + \alpha \varepsilon L(s)
$$

(12)

where $F_g$ and $F_r$ are attraction and repulsion forces, which are defined by Eq. (13) and Eq. (14), respectively:

$$
F_g = \left| x_{i-1} - x_i \right| \frac{q_i q_{i-1}}{\left\| x_{i-1} - x_i \right\|^2}
$$

(13)

$$
F_r = \left| x_i - x_{i+1} \right| \frac{q_i q_{i+1}}{\left\| x_i - x_{i+1} \right\|^2}
$$

(14)

where $q_i$ is the charge of the point $i$, which is defined as follows:
\[ q_i = \exp \left( -\frac{f(x^i) - f(x^{\text{best}})}{\sum_{k=1}^{n} (f(x^k) - f(x^{\text{best}}))} \right) \]  

(15)

where \( f(x) \) is the objective function. A firefly with better objective values has a higher charge. The objective function used in EFA-ANN model will be discussed in more detail in Section 2.3.

EFA uses the weights \( 1 - \alpha \) and \( \alpha \) to modify the effects of the attraction \( (F_g) \) and repulsion \( (F_r) \) forces during the optimization process. As alpha decreases from 1 to 0 during the optimization process in the early stages, the fireflies are significantly influenced by \( F_r \) to explore the promising area. In the final stages, the effect of \( F_g \) and \( F_r \) is increased and decreased, respectively, to help the fireflies exploit the optimal solution in the best promising area. Therefore, these improvements help EFA find a better solution as opposed to the separate applications of EA and FA. The efficiency of EFA will be validated in the following sections.

In each iteration, EFA chooses the best firefly based on the value of the objective function. The best firefly then performs a local search around its place by Levy flight while the rest of the population searches for other areas. At the end of the iteration, the best firefly is chosen, and the process continues until the termination criteria are reached. The termination criteria helps to reduce the computing time without affecting the solution quality. The objective function and termination criteria in this study will be discussed in Section 2.3. Figure 1 summarizes the pseudo-code of EFA, while Figure 2 illustrates a flowchart of the search process of EFA.
\textbf{Start} EFA

\begin{itemize}
\item Eq. (1) \hspace{1em} \textbackslash\text{Generate initial population of fireflies by logistic map}
\item Define objective function \( f(x) \) for all fireflies
\item Rank the fireflies based on objective function and find the current best
\end{itemize}

\textbf{while} (\( t < \text{MaxGeneration} \))

\begin{itemize}
\item \textbf{for} \( i = 1: n \) \hspace{1em} \( n \) is number of fireflies
\item \textbf{if} \( i \neq 1 \) \hspace{1em} \text{the current best firefly is not attracted by others}
\item \hspace{1em} Eq. (12) \hspace{1em} \text{Move firefly} \( i \) \text{in} \( d \)-dimension (Exploration phase)
\item \textbf{end if}
\item Eq. (3) \hspace{1em} \text{Local search (Exploitation phase)}
\item Evaluate new solutions and update the coordinate and the objective function
\item \textbf{end for} \( i \)
\item \( t = t + 1 \)
\end{itemize}

\textbf{end while}

Post process results and visualization

\textbf{End} EFA

Figure 1. Pseudocode for EFA.
2.2 Artificial Neural Network

Technical details of ANN can be found in a previous study by [42, 43]. This study outlines the main concepts of the ANN model. Figure 3 illustrates the layout of an ANN model with a set of artificial neurons. Each neuron in a layer sends a signal to another
neuron in the next layer by a connection, which is assigned a weight, and the weight represents the strength of the signal [44, 45].

In the hidden layer, the signals from the input layer are calculated by a linear function (see Eq. 16) and a transfer function (see Eq. 17) to generate the output signal of a hidden node [46] as depicted in Figure 4.

$$net_i = \sum_{p=1}^{P} w_{i,p} I_p + b_i$$  \hspace{1cm} (16)
where \( net_i \) is the value of the \( i^{th} \) net; \( w_{ip} \) and \( b_i \) are the weight of the \( p^{th} \) input to the \( i^{th} \) hidden node and the bias parameter of the \( i^{th} \) hidden node, respectively; \( I_p \) is the value of the \( p^{th} \) input node.

The transfer function is defined as:

\[
y_i = f (net_i) = \frac{1}{1 + \exp(-net_i)}
\]

(17)

where the transfer function in this study is a sigmoid function and \( y_i \) is the output signal of the \( i^{th} \) hidden node.

During the learning process, ANN uses the Mean Square Error (MSE), to evaluate the performance of the model as follows:

\[
MSE = \frac{1}{NN_{out}} \sum_{n=1}^{N_{out}} \sum_{o=1}^{N_{out}} (e_{n,o})^2
\]

(18)
where $N$ and $N_{out}$ are the number of instances and the number of outputs, respectively;

$$e_{n,o} = \bar{y}_{n,o} - y_{n,o}$$
is the training error at the $o^{th}$ output with $n^{th}$ instance; $y$ is the actual output and $\bar{y}$ is the predicted output by ANN.

The Levenberg–Marquardt algorithm [47, 48] is used to update the weights and biases to minimize the MSE. The calculation of the Levenberg–Marquardt algorithm is presented as:

$$w^{k+1} = w^k - \left( J^T J + \mu I \right)^{-1} J^T e^k$$  \hspace{1cm} (19)

where $w^k$ is the weight and bias matrix at the $k^{th}$ iteration; $I$ is the identity matrix; $\mu$ is the combination coefficient ($\mu > 0$); and $J$ is the Jacobian matrix [49]:

$$J = \begin{bmatrix}
\frac{\partial e_{1,1}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{1,1}}{\partial w_{1,2}} & \frac{\partial e_{1,1}}{\partial w_{2,1}} & \frac{\partial e_{1,1}}{\partial w_{2,2}} \\
\frac{\partial e_{1,2}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{1,2}}{\partial w_{1,2}} & \frac{\partial e_{1,2}}{\partial w_{2,1}} & \frac{\partial e_{1,2}}{\partial w_{2,2}} \\
\frac{\partial e_{1,2}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{1,2}}{\partial w_{1,2}} & \frac{\partial e_{1,2}}{\partial w_{2,1}} & \frac{\partial e_{1,2}}{\partial w_{2,2}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial e_{N_{out},1}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{N_{out},1}}{\partial w_{1,2}} & \frac{\partial e_{N_{out},1}}{\partial w_{2,1}} & \frac{\partial e_{N_{out},1}}{\partial w_{2,2}} \\
\frac{\partial e_{N_{out},2}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{N_{out},2}}{\partial w_{1,2}} & \frac{\partial e_{N_{out},2}}{\partial w_{2,1}} & \frac{\partial e_{N_{out},2}}{\partial w_{2,2}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial e_{N_{out},N_{out}}}{\partial w_{1,1}} & \cdots & \frac{\partial e_{N_{out},N_{out}}}{\partial w_{1,2}} & \frac{\partial e_{N_{out},N_{out}}}{\partial w_{2,1}} & \frac{\partial e_{N_{out},N_{out}}}{\partial w_{2,2}} \\
\end{bmatrix}$$  \hspace{1cm} (20)
where the error vector $e$ at each neuron is written as [46]:

$$
\begin{bmatrix}
e_{1,1} \\
e_{1,2} \\
\vdots \\
e_{1,N_{out}} \\
\vdots \\
e_{N,1} \\
e_{N,2} \\
\vdots \\
e_{N,N_{out}}
\end{bmatrix}
$$

(21)

2.3 EFA-ANN implementation

A flowchart of EFA-ANN is shown in Figure 5. Firstly, historical data is divided into learning and test data following k-fold cross-validation. The learning data is then divided into training and validation data with a ratio of 90% and 10%, respectively. The training data are trained by the Levenberg–Marquardt algorithm, while the validation data are used to validate the trained ANN. After finding an optimal ANN model, the test data is used to evaluate its performance. The EFA-ANN will be validated in Section 5.

This study also applied initial weights and biases in the range of [-0.5, 0.5] at the beginning of the training process. Chang et al. [32] showed that this is the optimal range for finding initial weights and biases of ANN. Figure 5 shows that the weights and biases are first updated inside the ANN. EFA automatically memorizes and optimizes these parameters to minimize prediction errors. As a result, the computing time can be significantly reduced. The Root Mean Square Error (RMSE) is used to calculate the objective function of EFA-ANN as follows:

$$
f = RMSE_{Validation-data}
$$

(22)
It is essential to clarify that RMSE is the objective function of EFA-ANN while the Mean Square Error (MSE) mentioned in Section 2.2 is the error function inside the ANN model. Also, the maximum generation is used as the termination criterion in this study.
Figure 5. A flowchart of the EFA-ANN model.
3. Model performance evaluation methods

Five performance measures were used to evaluate the performance of the proposed model. These include the linear correlation coefficient ($R$), determination coefficient ($R^2$), root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). Details of these performance measures can be found in the literature [42, 50, 51] and are calculated as follows:

$$R = \frac{\sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o} \bar{y}_{n,o} - \left( \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o} \right) \left( \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} \bar{y}_{n,o} \right)}{\sqrt{\sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o}^2 - \left( \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o} \right)^2} \sqrt{\sum_{n=1}^{N} \sum_{o=1}^{N_{out}} \bar{y}_{n,o}^2 - \left( \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} \bar{y}_{n,o} \right)^2}}$$  \hspace{1cm} (23)

$$R^2 = \frac{\sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o} \bar{y}_{n,o}}{\sum_{n=1}^{N} \sum_{o=1}^{N_{out}} y_{n,o} \bar{y}_{n,o}}$$  \hspace{1cm} (24)

$$RMSE = \sqrt{\frac{1}{NN_{out}} \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} (\bar{y}_{n,o} - y_{n,o})^2}$$  \hspace{1cm} (25)

$$MAE = \frac{1}{NN_{out}} \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} |y_{n,o} - \bar{y}_{n,o}|$$  \hspace{1cm} (26)

$$MAPE = \frac{1}{NN_{out}} \sum_{n=1}^{N} \sum_{o=1}^{N_{out}} \left| \frac{y_{n,o} - \bar{y}_{n,o}}{y_{n,o}} \right|$$  \hspace{1cm} (27)

4. Data collection

4.1 Dataset 1

Dataset 1 was generated by modeling a typical single-family house with a lightweight wood-frame structure in Istanbul, Turkey [52]. In this case study, the effect of the façade system on the total heating and cooling energy were investigated by changing its properties. Fig. 6 shows the details of the façade system including glass wool...
insulation (layer iii), two layers of oriented strand board (OSB) applied as sheathing (layer ii and v), two layers of gypsum board (layer i and iv) and a layer of cement-bonded particleboard used as the exterior finishing of the wall.

![Diagram of façade system](image)

Figure 6. The details of the façade system in dataset 1.

The façade system was modeled using different properties for the insulation layer, namely thickness and thermal conductivity (K-value), which are listed in Table 1. Five types of façades with different layer thicknesses were also investigated. The thickness of layer i, ii, iv, v and vi of type 1 are 1.5 cm, 2.5 cm, 2 cm, 2.5 cm and 1.5 cm, respectively. The details of all five types are listed in Table 1. A total of 180 simulations were carried out to evaluate how the thermal conductivity and thickness of the façade affect the total heating and cooling energy in the building.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Values</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insulation K-value (iii)</td>
<td>W/m-K</td>
<td>0.03, 0.04, 0.05, 0.08</td>
<td></td>
</tr>
<tr>
<td>Insulation thickness (iii)</td>
<td>cm</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 [i, ii, iv, v, vi: 1.5, 2.5, 2, 2.5, 1.5 (cm)]</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 [i, ii, iv, v, vi: 2, 2.5, 2, 1.5 (cm)]</td>
<td></td>
</tr>
<tr>
<td>Façade type</td>
<td>N/A</td>
<td>3 [i, ii, iv, v, vi: 1, 3, 1.5, 3, 1.5 (cm)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 [i, ii, iv, v, vi: 2.5, 1.5, 3, 1.5, 1.5 (cm)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 [i, ii, iv, v, vi: 1, 3.5, 1.9, 1, 1.5 (cm)]</td>
<td></td>
</tr>
<tr>
<td>Total heating and cooling energy</td>
<td>kWh</td>
<td>Min: 6094.24; Max: 11095.06</td>
<td>Output</td>
</tr>
</tbody>
</table>

Table 1. Statistical parameters for dataset 1.

After determining the simulated cases, the building models were drawn in Sketchup, and the thermal properties of the wall components, including window location, direction and orientation, were then inputted into EnergyPlus to simulate the energy consumption of the building. As the primary purpose of the simulation is to validate the effect of the façade system on the total heating and cooling energy in the building, the variables of these simulations are restricted to the type of façade material. Other factors of the building were maintained constant, and the properties used in the simulation are listed in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building type</td>
<td></td>
<td>Residential building</td>
</tr>
<tr>
<td>Building location</td>
<td></td>
<td>Istanbul, Turkey</td>
</tr>
<tr>
<td>Floor area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First story</td>
<td>m²</td>
<td>81.7</td>
</tr>
<tr>
<td>Second story</td>
<td>m²</td>
<td>48.4</td>
</tr>
<tr>
<td>Run period</td>
<td>year</td>
<td>1</td>
</tr>
<tr>
<td>Electric equipment</td>
<td>W/m²</td>
<td>10</td>
</tr>
<tr>
<td>Lighting</td>
<td>W/m²</td>
<td>12</td>
</tr>
</tbody>
</table>
Thermostat

<table>
<thead>
<tr>
<th></th>
<th>°C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating setpoint (constant)</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Cooling setpoint (constant)</td>
<td>26</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Design information in EnergyPlus simulation.

4.2 Dataset 2

Dataset 2 consisted of 768 entries that were generated from twelve building types using Ecotect simulation software [53]. These twelve building types were represented by 18 simple cubes (3.5m x 3.5m x 3.5m) and the shape of each type is shown in Fig.7. Therefore, the buildings have the same volume but different surface areas and dimensions. These buildings were each simulated as a residential building in Athens, Greece. This dataset is used to investigate the effect of dimension on the cooling load (CL) in a building. Hence, the same material properties of the façade system were used for all twelve buildings, including U-value of the wall (1.78 W/m²K), window (2.26 W/m²K) floors (0.86 W/m²K) and roofs (0.50 W/m²K). The lighting level and latent heat were set to 300 lux and 2 W/m², respectively. The cooling load of the residential building was simulated by using eight features including the relative compactness (RC), surface area, wall area, roof area, overall height, orientation, glazing area and glazing distribution.

Each input parameter represents a property of the building. For instance, the RC indicator [54] represents the type of building, and the RC-value of each building is shown in Fig. 7. The RC is calculated by Eq. (27) as follows:

\[
RC = 6V^{2/3}A^{-1}
\]  

(28)

where \( V \) and \( A \) are the volume and surface area of the building, respectively.
The experiments simulated the building using two configurations, namely with and without glazing. In a glazing system, three glazing-to-floor area ratios were used, including 10%, 25%, and 40%. Five glazing distributions are considered including: (1) 25% glazing for each faces; (2) 55% for the north side and 15% for the other faces; (3) 55% for the east face and 15% for the remaining faces; (4) 55% for the south face and 15% for the other faces; and (5) 55% for the west face and 15% for the other faces [53]. Finally, all building shapes were rotated to four orientations, namely north, south, east and west. Overall, 768 building configurations were simulated, and the details of the input and output parameters are provided in Table 4. By applying this process, all input values are discretized and calculated from the twelve building types. The number of possible values of each input are also listed in Table 4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>No. of possible values</th>
<th>Min.</th>
<th>Max.</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative compactness</td>
<td>N/A</td>
<td>12</td>
<td>0.62</td>
<td>0.98</td>
<td>Input</td>
</tr>
<tr>
<td>Surface area</td>
<td>m²</td>
<td>12</td>
<td>514.50</td>
<td>808.50</td>
<td></td>
</tr>
<tr>
<td>Wall area</td>
<td>m²</td>
<td>7</td>
<td>245.00</td>
<td>416.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Roof area</td>
<td>m²</td>
<td>4</td>
<td>110.25</td>
<td>220.50</td>
<td></td>
</tr>
<tr>
<td>Overall height</td>
<td>m</td>
<td>2</td>
<td>3.50</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>Orientation</td>
<td>N/A</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Glazing area</td>
<td>%</td>
<td>4</td>
<td>0.00</td>
<td>40.00</td>
<td></td>
</tr>
<tr>
<td>Glazing distribution</td>
<td>N/A</td>
<td>6</td>
<td>0.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>Cooling load (CL)</td>
<td>kW</td>
<td>768</td>
<td>10.90</td>
<td>48.03</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Descriptions of dataset 2.

5. Performance Evaluation and Discussion

5.1 Data Preprocessing and Model Application

This study used the K-fold cross-validation method to alleviate problems with overfitting data [55]. The historical data was divided into 10 folds, which is the optimal number determined by other researchers [56]. Also, the K-fold cross-validation method can help to provide equal weightings to the results to obtain a fair comparison. The models that were used to assess the performance of our proposed model also utilized the K-fold cross-validation method [50, 53, 57].

In this method, each dataset is randomly divided into 10 separate folds. There are 10 rounds in the entire process. In each round, nine folds are used for training the EFA-ANN and the remaining fold is used for testing in each round. Therefore, all data is guaranteed to be used in both the learning and testing phases. Finally, the average results in ten rounds are obtained to assess the performance of the model. The flowchart for each round is shown in Figure 5. In addition, all parameters of EFA-ANN used in this study are listed in Table 5.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fireflies</td>
<td>15</td>
</tr>
<tr>
<td>Number of Maximum generations</td>
<td>15</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>0.9</td>
</tr>
<tr>
<td>EFA</td>
<td>Adaptive inertial weight (\theta) ((10^{-2} / 0.9)^{\text{MaxGeneration}})</td>
</tr>
<tr>
<td>Percentage of training data</td>
<td>90%</td>
</tr>
<tr>
<td>Percentage of validation data</td>
<td>10%</td>
</tr>
<tr>
<td>Number of folds</td>
<td>10</td>
</tr>
<tr>
<td>Objective function</td>
<td>RMSE</td>
</tr>
<tr>
<td>ANO</td>
<td>Transfer function</td>
</tr>
<tr>
<td>Learning algorithm</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>Number of hidden layers</td>
<td>1</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5. The parameter setting in EFA-ANN.

At the beginning of the training process, the data is normalized to avoid numerical difficulties, i.e. inputs in higher numeric ranges may dominate those in smaller numeric ranges [58]. Therefore, this study normalizes the data to the range of \([-1, 1]\). The normalized value \((x')\) is calculated from the original value \((x)\) as follows:

\[
x' = \frac{2(x - \min(x))}{\max(x) - \min(x)} - 1
\]  

(29)

5.2 Results and discussion

Table 6 lists the predictive performance measures of the proposed model for dataset 1. Several models are used for comparing the predictive accuracy of the proposed model for the same dataset. For example, Naji et al. proposed an extreme learning machine (ELM) method to forecast the energy consumption in a building [52]. Their model performs quite well \((RMSE = 74.02 \text{ (kWh)}, R = 0.999\) and \(R^2 = 0.997\)). Their comparison showed that ELM was superior to genetic programming (GP) and ANN’ models, which were run in their study, in terms of accuracy and computing time.
### Table 6. Performance measures and improvement rates of the EFA-ANN model for dataset 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Performance measure</th>
<th>Improvement rates by EFA-ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R$ (-)</td>
<td>$R^2$ (-)</td>
</tr>
<tr>
<td>ELM [52]</td>
<td>0.999</td>
<td>0.997</td>
</tr>
<tr>
<td>ANN’ [52]</td>
<td>0.971</td>
<td>0.943</td>
</tr>
<tr>
<td>GP [52]</td>
<td>0.977</td>
<td>0.954</td>
</tr>
<tr>
<td>ANN</td>
<td>0.999</td>
<td>0.998</td>
</tr>
<tr>
<td>EFA-ANN</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Note: CT is computing time which is calculated based on 1 running time. *, ** indicates significance levels higher than (1%, 5%), respectively;

However, Table 6 indicates that the performance of EFA-ANN is superior to ELM and other methods. Moreover, EFA-ANN produces the smallest error rate (RMSE), and it is 93.28% - 98.50% better than those reported by other studies. Notably, in terms of computational cost, EFA-ANN is 5 times faster than ELM and 7 times faster than ANN’ and GP. Moreover, this research also compares the performance of EFA-ANN with the ANN model to confirm that there is a prominent improvement over the combined EFA-ANN model. The ANN model, in this case, is configured with the same setting as the ANN in the EFA-ANN model. The computing time of ANN is modified to be consistent with that of EFA-ANN for a fair comparison. The comparative results show that the error rates of EFA-ANN are 18.84% - 87.47% better than those of ANN. The results indicate that EFA-ANN can provide an alternative approach for predicting the energy consumption in a building.

The predicted results from EFA-ANN and other methods for dataset 2 are listed in Table 7. The proposed approach is shown to be superior to other methods available in the literature. For example, EFA-ANN obtains the lowest RMSE (0.51 kW) compared to other models: iteratively reweighted least squares (IRLS) (3.39 kW) [53], random forests
(RF) (2.57 kW) [53], ensemble model (1.57 kW), smart artificial firefly colony algorithm-based support vector regression (SAFCA-SVR) (0.68 kWh) [57] and ANN model. Similarly, Table 7 shows that EFA-ANN has the lowest MAE and MAPE, which is equal to 0.38 (kW) and 1.71 (%), respectively. Also, the $R$ and $R^2$ of EFA-ANN are equal to SAFCA-SVR, which is better than the ensemble model. With these results, EFA-ANN has improved the error rate from 16.18% to 84.84% when compared with other models. Additionally, the proposed model is around 5 times faster than the SAFCA-SVR model in predicting CL of this dataset (47 minutes compared to 240 minutes). Table 7 shows that the error rates of EFA-ANN are 31.87% - 53.27% better than ANN. Details of the comparison between ANN and EFA-ANN will be discussed hereafter.

<table>
<thead>
<tr>
<th>Method</th>
<th>$R$</th>
<th>$R^2$</th>
<th>RMSE (kW)</th>
<th>MAE (kW)</th>
<th>MAPE (%)</th>
<th>CT (s)</th>
<th>Improvement rates by EFA-ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R$</td>
</tr>
<tr>
<td>IRLS [53]</td>
<td>N/A</td>
<td>N/A</td>
<td>3.39</td>
<td>2.21</td>
<td>9.41</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>RF [53]</td>
<td>N/A</td>
<td>N/A</td>
<td>2.57</td>
<td>1.42</td>
<td>4.62</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>Ensemble model [50]</td>
<td>0.99</td>
<td>0.97</td>
<td>1.57</td>
<td>0.97</td>
<td>3.46</td>
<td>N/A</td>
<td>1.27**</td>
</tr>
<tr>
<td>SAFCA-SVR [57]</td>
<td>1.00</td>
<td>0.99</td>
<td>0.68</td>
<td>0.47</td>
<td>2.04</td>
<td>1440</td>
<td>-</td>
</tr>
<tr>
<td>ANN</td>
<td>1.00</td>
<td>0.99</td>
<td>1.10</td>
<td>0.56</td>
<td>2.51</td>
<td>280</td>
<td>-</td>
</tr>
<tr>
<td>EFA-ANN</td>
<td>1.00</td>
<td>1.00</td>
<td>0.51</td>
<td>0.38</td>
<td>1.71</td>
<td>280</td>
<td>-</td>
</tr>
</tbody>
</table>

Note: CT is computing time which is calculated based on 1 running time.; *, ** indicates significance levels higher than (1%, 5%), respectively;

Table 7. Performance measures and improvement rates of the EFA-ANN model for dataset 2.

This study compares the EFA-ANN model with the single ANN model to investigate the effects of EFA on the RMSE value and computing time. In order to make a fair comparison, the parameters in the single ANN model were set to be identical with the ANN model in EFA-ANN. Only the number of iterations of the single ANN model,
in this case, was increased to extend the computing time (to be consistent with EFA-ANN). Both single ANN and EFA-ANN were run on the same computer. Figure 8 shows that the single ANN is stuck at the local optima after around 20 and 50 seconds for Dataset 1 (a) and Dataset 2 (b), respectively. In contrast, EFA-ANN approaches a better result because EFA helps ANN find better weights and bias values. This comparison confirms that EFA is effective in optimizing the weights and biases of ANN.

![Figure 8. Convergence results of ANN and EFA-ANN for (a) Dataset 1 and (b) Dataset 2](image)

The statistical relationship between the predicted outputs obtained by EFA-ANN and the actual outputs of the two datasets is shown in Figure 9. The $R$-values for the two datasets are almost equal to 1, which indicates that the predicted values from EFA-ANN have a strong correlation with the actual values. Notably, the MAPE in dataset 1 is equal to 0.04%, which indicates that the predicted values are approximately equal to the actual values. In other words, the proposed model can forecast the exact energy consumption for dataset 1. Meanwhile, the MAPE for dataset 2 is higher than that for dataset 1 but remains negligible (1.71%). The results of EFA-ANN are slightly different from the actual values, but these results are predicted with better accuracy than the aforementioned
methods. Therefore, the proposed EFA-ANN is an efficient model for predicting the energy consumption (including HL and CL) of a building.

![Figure 9](image.png)

**Figure 9.** The correlation between the actual and the predicted outputs for dataset 1 (a) and dataset 2 (b).

### 5.3 Sensitivity Analysis

A sensitivity analysis is performed to quantify the effects of different inputs on the predicted energy consumption. Among the different approaches, this study used the method proposed by Garson [59] because it is suitable for discrete input data such as wall type, relative compactness, orientation, and glazing distribution. This method is also suitable for the ANN model as the sensitivity analysis deconstructs the weights of the connections between neurons to quantify the influence of the various inputs. The influence of a specified input on the output can be determined by assessing all the connecting weights between the nodes of interest. Therefore, all connecting weights between a specific input and output are identified, and the importance of all the inputs is calculated (as a percentage) by the following equation:
\[ Q_p = \sum_{i=1}^{I} \left( \frac{w_{ip}}{\sum_{p=1}^{P} w_{ip}} \times \frac{w_{oi}}{\sum_{i=1}^{I} w_{oi}} \right) \]  

(30)

Where \( Q_p \) is the impact of the input \( p^{th} \) on the output in percentage; \( w_{ip} \) is the weight of the \( p^{th} \) input to the \( i^{th} \) hidden node; \( w_{oi} \) is the weight of the \( i^{th} \) hidden node to the output.

The results from the sensitivity analysis on datasets 1 and 2 are shown in Figure 10. For dataset 1, the insulation thickness has the highest impact on the total heating and cooling load (43.8%), followed by the facade type and K-value of the insulation (34.1% and 22.1%, respectively). This result is reasonable because the insulation thickness has a prominent effect on the energy consumption in a building, which was shown in previous studies [60, 61]. Meanwhile, the glazing area (29.4%) and the relative compactness (27.5%) are the most critical parameters that affect the total cooling load in dataset 2. This result is expected because the solar heat gain will increase with a larger glazing area, which thereby affects the cooling load in the building. Also, several studies confirmed that an increasing window-to-wall ratio causes an increase in the cooling load [62, 63]. In contrast, the orientation of the facade and the roof area have the least impact with just under 5%. The results of the sensitivity analysis can help a designer to quickly identify which input should be modified to improve the energy performance of the building. For example, the insulation thickness in dataset 1 or the glazing area and relative compactness in dataset 2 should be modified before the other variables to effectively improve the energy performance of the building.
Figure 10. The effect of several design parameters on the energy consumption of the building.
6. Conclusions

A novel method was developed for predicting the energy consumption in a building. The model, namely EFA-ANN, integrates EFA into ANN to improve its performance by optimizing the set of initial parameters. The proposed approach was compared with several published models in terms of its speed and accuracy in forecasting heating and cooling energy. Two datasets with various material and isolation properties (dataset 1), and building characteristics (dataset 2) were used to validate EFA-ANN. Additionally, a 10-fold cross-validation method was applied to mitigate the over-fitting problem when comparing the performance of these models.

In dataset 1, EFA-ANN showed a 93.28% - 98.50% improvement of other methods reported in the literature. Notably, EFA-ANN is 5 times faster than the ELM method and 7 times faster than GP in terms of computing time. Similarly, EFA-ANN not only obtained the lowest RMSE, MAE and MAPE, but also the highest $R$-value compared to the other techniques for dataset 2. Also, the computing time of EFA-ANN is less 5 times than that of SAFCA-SVR. The obtained results demonstrated the strong capabilities of machine learning in predicting the energy consumption of buildings.

A sensitivity analysis was also performed to identify the input with the most critical impact on the output of each dataset. The results of the sensitivity analysis indicate that the insulation thickness and the glazing area have the most significant impact for datasets 1 and 2, respectively. This result can help designers to quickly validate their design of a building and improve its energy performance by focusing on these essential inputs.

This study showed that the proposed EFA-ANN model achieved both good results and a short computational time relative to other methods. Therefore, EFA-ANN can help
energy engineers to design energy-efficient buildings whilst reducing experimental requirements and it could assist civil engineers and construction managers in the early design phase of energy-efficient buildings. Besides, the proposed approach can be used as a useful tool for quickly and accurately solving many problems in engineering, including energy-efficient buildings, construction material strength, and structure strength.

**Acknowledgments**

The first author would like to thank the University of Melbourne for offering the Melbourne Research Scholarship. This work was mainly supported by the CRC-P for Advanced Manufacturing of High Performance Building Envelope project, funded by the CRC-P program of the Department of Industry, Innovation and Science, Australia, and the Asia Pacific Research Network for Resilient and Affordable Housing (APRAH) grant, funded by the Australian Academy of Science, Australia. This work was also supported by the ARC Training Centre for Advanced Manufacturing of Prefabricated Housing (CAMP.H) at the University of Melbourne.
References


