Deep Highway Networks and Tree-Based Ensemble for Predicting Short-Term Building Energy Consumption

Muhammad Waseem Ahmad [1,*], Anthony Mouraud [2], Yacine Rezgui [1] and Monjur Mourshed [1]

[1] BRE Trust Centre for Sustainable Engineering, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK; rezguiY@cardiff.ac.uk (Y.R.); mourshedm@cardiff.ac.uk (M.M.)

[2] Commissariat á l'énergie atomique et aux énergies alternatives (CEA), CEA Tech en Région (CTREG), Département Grand Ouest (DGDO), 44340 Bouguenais, France; anthony.mouraud@cea.fr

*   Correspondence: AhmadM3@cardiff.ac.uk

Abstract: Predictive analytics play a significant role in ensuring optimal and secure operation of power systems, reducing energy consumption, detecting fault and diagnosis, and improving grid resilience. However, due to system nonlinearities, delay, and complexity of the problem because of many influencing factors (e.g., climate, occupants' behaviour, occupancy pattern, building type), it is a challenging task to get accurate energy consumption prediction. This paper investigates the accuracy and generalisation capabilities of deep highway networks (DHN) and extremely randomized trees (ET) for predicting hourly heating, ventilation and air conditioning (HVAC) energy consumption of a hotel building. Their performance was compared with support vector regression (SVR), a most widely used supervised machine learning algorithm. Results showed that both ET and DHN models marginally outperform the SVR algorithm. The paper also details the impact of increasing the deep highway network's complexity on its performance. The paper concludes that all developed models are equally applicable for predicting hourly HVAC energy consumption. Possible reasons for the minimum impact of DHN complexity and future research work are also highlighted in the paper.

Keywords: HVAC systems; deep learning; energy efficiency; tree-based ensemble algorithms; machine learning; support vector regression

1. Introduction

Globally, there are growing concerns regarding the total energy consumption from the building sector, which is one of the main substantial users of energy. Buildings account for 40% of the world's total energy consumption and contribute towards 30% of the total CO_2 emissions [1]. According to the current European Union (EU) roadmap, the EU is committed to reducing greenhouse gas emission by 20%, reaching a share of renewable energy in gross final energy by 20%, and reducing total primary energy consumption by 20%—by 2020 as compared to the 1990 levels [2]. In the non-domestic sector, hotels and restaurants are the third largest consumers of energy, accounting for 30%, 18%, 16% and 14% in Spain, France, the UK and the USA, respectively [3,4]. In Greece and Spain, hotels are responsible for about 1/3 of the total energy demand [5]. In hotels, nearly half of the electricity is used for space conditioning purposes [6]. Because of a significant amount of energy consumption, there have been increasing concerns on hotels' energy use and efforts to effectively manage energy consumption. Predicting energy consumption over a wide range of time horizons is one of the key features of smart girds. It allows for building managers/owners to make informed decisions; e.g., increasing share of renewable energy sources and shifting energy use to off-peak times.

1.1. Context and Objectives

The gap between actual measured performance and predicted energy performance of buildings, typically addressed as 'the performance gap', is an increasing concern for the building industry [7]. The gap can be explained by a wide range of factors that get amplified during the lifecycle of the building [8–10]. The energy performance gap negatively impacts occupants' comfort and energy consumption. Therefore, it is critical for energy managers/building owners to identify causes of operational energy performance gap and to take countermeasures as quickly as possible. Different researchers have tackled this problem by identifying performance gap and fault detection and diagnosis using time-dependent and steady-state analytical modelling, data-driven modelling or knowledge-based methods. The objective of this paper is to detail the performance of the developed machine learning models, which could be used to identify an energy performance gap, make informed decisions by energy managers/building owners, and detect and diagnose faults in a hotel building.

Accurate prediction of energy consumption is an exigent task due to system nonlinearities, delays, and complexity of the studied problem. Recently, a number of different techniques has been developed and applied to predict energy consumption at a building level. These techniques can be divided into data-driven and first principle methods.

First principle-based methods (e.g., TRNSYS, EnergyPlus, DOE-2) require detailed information about building features, and installed energy systems.　These methods, because of their multi-domain modelling capabilities, often enable users to assess different design strategies with lower uncertainties [11]. However, because of the complex nature of human behaviour inside buildings, these methods do not perform well for occupied buildings [12]. Physical models can be computationally intensive and therefore an exhaustive exploration of parameter space for optimal control strategies could be infeasible. Because of these factors, a physical model of a building/energy system is mostly avoided, and a simpler, efficient and accurate data driven model is created.　Data-driven techniques do not require detailed information about building characteristics or heating, ventilation and air conditioning (HVAC) systems. However, the computational cost of learning and hyper-parameters could be high for data-driven models. The prediction accuracy is also influenced by the quantity and quality of the available training dataset. The authors acknowledge the fact that, for both data-driven and detailed models, there is a trade-off between computational cost and prediction accuracy. Mostly, data-driven techniques are better suited for near real-time optimisation applications as they need significantly less prediction time and require less prior information about the buildings of interest.

Accurate energy prediction models are used by facility managers and utilities to effectively schedule and control continuously fluctuating energy supply and demand, and avoid penalties that could occur due to the difference between predicted and consumed energy. Machine learning models are often the preferred choice for real-time control applications because of their fast response time as compared to detailed simulation models [3]. Due to instability issues, most of the widely used machine leaning techniques are likely to be unreliable. As the developed models in this paper could be used for optimal control, the stability of developed models is critical. In recent years, more advanced prediction methods (ensemble and deep learning) were developed to overcome the limitations of traditional methods.

1.2. Related Work

Deep-learning methods are one of the most efficient methods for classification problems and have been tested for numerous applications [13–16]. Among these methods, the recurrent neural network is capable of instantiating almost arbitrary dynamics and allowing the information flow to be "memorized" during the computation to enrich further processing. Recently, the performance of "Long Short-Term Memory" (LSTM)—a type of recurrent neural network method—has been tested in different research studies [17,18]. However, these techniques have been extensively applied in the vision domain; various other research works have also applied deep-learning methods for other research areas, e.g., cancer (diagnosis and detection) [19], chatbots and NLP (Natural Language Processing) [20], games (scoring

and human-level playing) [21], and heart diseases [22]. One of the limitations of deep-learning methods was that their performance did not improve with the increase in network's depth. Recently, a number of improvements have been performed to tackle this issue, e.g., by using recurrent like behaviour in feed-forward models (ResNets [23], Inception/Xception [24], Highways [25]), introducing reinforcement learning, previous input auto-encoding and incremental layer learning, and evolution of learning rules (e.g., dropout, Adam, Nesterov, batches, blurring inputs). These improvements have significantly enhanced the performance of deep-learning methods.

With the growth in performance of deep learning methods and despite the fact that historical prediction models are less complex, numerous studies in the past few months proposed diverse deep learning approaches for estimating building energy consumption [26] and forecast/prediction [27–29] with both feed-forward and recurrent neural network models. Optimization of energy consumption through reinforcement learning has also recently been studied with deep learning models [30]. Deep learning is being used either as a predictive modeling tool or a feature extractor as in [31]. The study also showed that the feature extraction property is of more interest than the predictive property itself, which does not perform better than eXtreme Gradient Boosting in this case. In most cases, deep learning models' hyperparameters are empirically chosen and in only a few cases do these parameters span a large part of the possible space. In particular, studies often show comparison with simplest networks' models.

Support vector machines (SVMs) are used in different applications of the building energy sector, e.g., Liang and Du [32] applied the SVM method for fault detection and diagnoses (FDD) by combining SVM, and model-based FDD. Mohandes et al. [33] predicted wind speed and Esen et al. [34] modeled a ground-coupled heat pump system by using SVM. Support vector machines have recently attracted researchers' focus in the field of building energy prediction. To the authors' knowledge, the first work was reported by Dong et al. [35]. The authors applied SVM to predict monthly building energy consumption in a tropical region (Singapore). Outside dry-bulb temperature, solar radiation, and relative humidity were considered as input parameters. The weather data were collected from a weather station which was approximately 12 miles away from the buildings under investigation. Although the percent error was small, the inaccuracies in the results due to weather data were not discussed in the paper. Li et al. [36] and Li et al. [37] predicted hourly cooling load in an office building in China. The author also compared SVM with different artificial neural networks. The SVM method performed slightly better than neural network methods. This was because of the structural risk minimization principle of SVM, which is used to minimize the upper bound of the generalization error. On the other hand, artificial neural networks minimized the training error. The performance of SVM can be enhanced through combination with other computational intelligence techniques. To enhance the SVM by reducing the effect of noise and outliers in the data set, Li et al. [38] proposed fuzzy SVM. As in load prediction, the old data are less important as compared to the new data. SVM does not have the ability to distinguish a new pattern and therefore the authors proposed the fuzzy SVM method.

Decision trees (DT) are used to classify a dataset into various predefined target values or classes. A DT based model can be represented by logical statements/rules. Decision tree, tree-based ensemble algorithms, in particular, are less popular in building energy research domains. Decision tree based methods were used by Yu et al. [39] to predict energy consumption. The authors concluded that decision tree based algorithms could be used to develop reliable models. A decision support model to reduce a school building's electricity was developed by Hong et al. [40]. The authors used decision trees to form a group of educational buildings based on electricity consumption. Hong et al. [41] used decision trees for clustering a type of multifamily housing complex based on gas consumption. Tso and Yau [42] compared regression analysis, DT and artificial neural networks (ANN) for predicting electricity consumption. The authors concluded that the DT algorithm could be a viable option for predicting energy consumption. In a recent study by Ahmad et al. [3], the authors compared the performance of ANN and random forest (RF)—a tree-based ensemble algorithm. However, ANN performed marginally better in this study, and the authors concluded that

both developed models have similar prediction accuracy and are equally applicable for predicting building energy consumption.

The paper compares the performance in prediction of hourly HVAC energy consumption of a hotel building by using three different machine learning (ML) approaches: deep highway networks, extremely randomised trees, and support vector regression. The research presented in this paper mainly addresses the following aspects:

- The use of tree-based ensemble techniques and deep highway networks for predicting HVAC energy consumption from contextual data;
- Studying the impact of networks' depth on prediction performance of DHN models;
- Demonstrate a prediction error of nearly 6% (normalised root mean square error)on hourly data for two of the best currently known machine learning algorithms (tree-based ensembles and deep learning).

The paper addresses the problem of predicting energy consumption of a hotel building. Predicting energy consumption of hotels and restaurants is a challenging task as it does not exhibit clear patterns. From the literature review, it was found that none of the previous studies compared the performance of recently developed deep learning and tree-based ensemble methods. The presented research work also discusses whether there is a need to develop deep learning models for high-resolution prediction of energy consumption or the current state-of-the-art methods are equally comparable.

The rest of the paper is organised as follows: the methodology of the developed models is presented in Section 2. In Section 2.1, principles of deep highway networks, support vector regression, and extremely randomised trees are described. Prediction results are described in Section 3, whereas Section 4 presents comparison between three developed machine learning models along with discussions. Concluding remarks are drawn at the end of the paper.

2. Materials and Methods

This section introduces the three proposed data-driven techniques for predicting HVAC energy consumption. The section also details training and testing datasets along with the evaluation metrics used for comparing the studied techniques. In this paper, SVR is used for comparison purposes, as it is one of the most widely machine learning techniques in a built environment research domain. Figure 1 illustrates the schematic overview of the proposed research. Historical weather, energy consumption and occupancy data were retrieved from a database for developing prediction models. The data was pre-processed for model development by removing outliers and treating missing values. The data was also normalised for SVR algorithms. Important features were selected by using random forest and extra trees algorithms. The models' hyper-parameters were tuned by using either a genetic algorithm (GA) (for deep highway network) or a step-wise search method (support vector regression and extra trees).
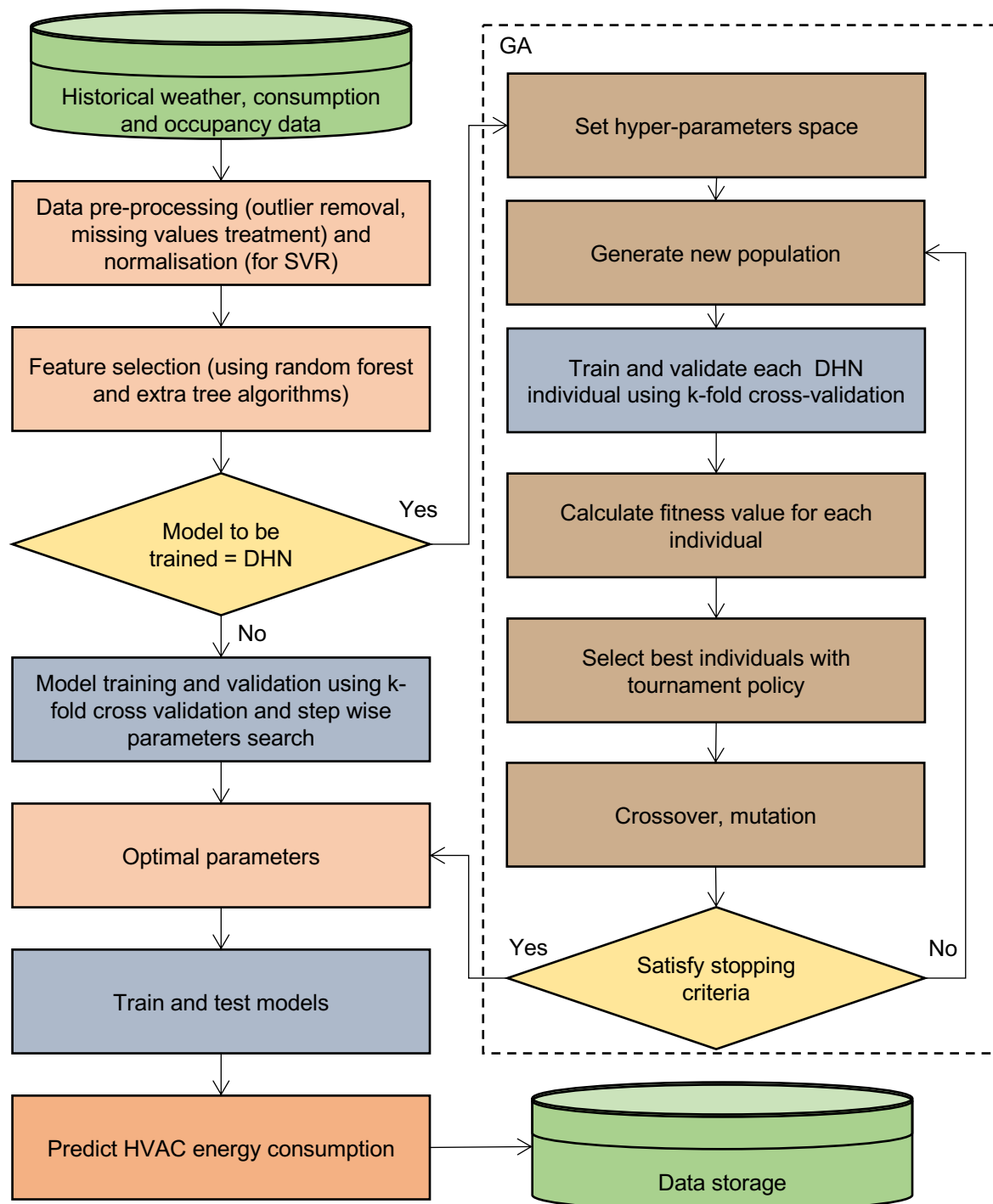
Figure 1. The schematic overview of the proposed research methodology.

2.1. Machine Learning Algorithms

Three data-driven algorithms for predicting energy consumption are introduced in this section. These algorithms include deep highway networks (DHN), extremely randomised trees (ET) and support vector regression (SVR).

2.1.1. Support Vector Regression

Support vector machines are one of the most widely used machine learning approaches to predict energy consumption. They are divided into two main categories: support vector classification (SVC) and support vector regression (SVR) [43]. As the name suggests, SVR is used for regression problems and the main objective is to find a relationship between input and output features while assuming that the joint distribution of the features is unknown. The SVR algorithms map the input data into a high-dimensional feature space through a nonlinear mapping and performing a linear regression in this feature space [43].

For modelling a process, suppose that the normalized inputs vector is X_i (represents vector of input parameters) and Y_i represents the outputs; then, the set of samples is defined as $\{(X_i, Y_i)\}_{i=1}^{N}$, where N is the length of training data set. The algorithm approximates the relationship between the outputs and inputs, while projecting input space in a higher dimensional space. In the present work, we make use of the framework defined in [44,45]:

$$Y = f(X) = W \cdot \phi(X) + b, \tag{1}$$

where $\phi(X)$ represents the high-dimensional space, which is nonlinearly mapped from the input data. The coefficients W and b are estimated by using Equation (2) i.e., by minimising a regularised risk function [44,45].

$$Minimize: \ \frac{1}{2}||W||^2 + C\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(Y_i, f(X_i)), \tag{2}$$

$$L_\varepsilon(Y_i, f(X_i)) = \begin{cases} 0, & if \ |Y_i - f(X_i)| \leq \varepsilon, \\ |Y_i - f(X_i)| - \varepsilon, & others. \end{cases} \tag{3}$$

Minimizing $||W||^2$ ensures as small W as possible. In the above equations, C is a penalty parameter (also known as regularisation parameter) that determines the balance between model flatness and tolerance with regard to errors that are larger than ε. The empirical error is denoted by the second term of Equation (2), which is measured by the ε-intensity loss function (Equation (3)). The loss is zero, if the predicted value is within the ε-tube. On the other hand, the loss is the difference between the radius ε of the tube and predicted error, if the predicted value is outside the tube [36]. In order to relax constraints in the estimation of W and b, slack variables ζ_1 and ζ_1^* are introduced leading the above equation to become the primal objective function given by Equation (4) [44,45]:

$$\underset{\zeta_1, \zeta_1^*, W, \ b}{Minimize} \ : \ \frac{1}{2}||W||^2 + C\frac{1}{N}\sum_{i=1}^{N} (\zeta_1 + \zeta_1^*), \tag{4}$$

$$Subject \ to: \ \begin{cases} Y_i - W \cdot \phi(x_i) - b \leq \varepsilon + \zeta_1, \\ W \cdot \phi(x_i) + b \leq \varepsilon + \zeta_1^*, \ i = 1, 2, \dots, N, \\ \zeta_1 \geq 0 \ \ \zeta_1^* \geq 0. \end{cases}$$

This primal form of the optimization problem can be solved expressing the Lagrangian and then the dual form of the optimization problem [44,45].

Finally, the Mercer kernel K is defined as:

$$K(X_i, \ X_j) = exp\left(-\frac{||X_i - X_j||^2}{\delta^2}\right), \tag{5}$$

which allows for expressing the inner products in the infinite dimensional features space ϕ so that Equation (1) becomes [44,45]:

$$Y = f(X) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \, K(X_i, \, X) + b, \tag{6}$$

where α_i, α_i^* are the Lagrange multipliers (constrained to be ≥ 0) and N the number of support vectors.

The normalized predicted output Y, which is obtained from an SVR model, should be transformed into the actual prediction value by using the following equation:

$$\hat{q} = q_{min} + Y \cdot (q_{max} - q_{min}). \tag{7}$$

2.1.2. Deep Highway Networks

Deep highway network (DHN) is a concept introduced in [25] by taking advantage of some of the properties of LSTM models in a purely feedforward fashion. In this work, the model proposed proved to be more stable in learning with an increase in the number of hidden layers than previous fully connected feedforward deep neural network models. This ability is obtained by introducing the following concepts. Generally, a feedforward Neural Network transformation of the input is given in Equation (8) [29]:

$$y = H(x, W_H). \tag{8}$$

In the above equation, y is the output of the network, and H is the nonlinear transformation applied to the input x and weighted by a parameters matrix W_H. H can consist of multiple layers of nonlinear transformation and their corresponding weights. Each of these layers receives inputs from the preceding layer's outputs and outputs to the next layer. In a simplified form, a highway network can be defined by Equation (9) [29]:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C). \tag{9}$$

The transform gate, $T(x, W_T)$, transforms the input, whereas the carry gate (x, W_C) allows for carrying input in a possibly unchanged form through different layers of the network, depending on the weights applied. This property resembles the residual behavior of ResNets, where unchanged input is propagated in the deep structure of the network, helping the network to keep learning even with a high number of hidden layers (efficient ResNets can have up to 1000 layers in recent works [46]). Similarly to the work of Srivastava et al. [25], we used a carry gate defined by $C = 1 - T$. The resulting transformations achieved by the network are summarized in Equation (10) [29]:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_T)). \tag{10}$$

Rectifier Linear Units (ReLu) [47] are used to populate the regular hidden layers and transform gate layers use a sigmoidal activation function. ReLu units are defined by Equation (11) [29]:

$$y = max(0, x), \tag{11}$$

with

$$x = \sum_{i=1}^{K} w_i * x_i. \tag{12}$$

In the above equation, $w_i * x_i$ is the weighted output value of the connected input neurons.

2.1.3. Extremely Randomized Trees

Extremely randomized trees or Extra-Trees (ET) [48] introduce stochasticity during the induction production of classical decision trees [49]. ET was developed as an extension of another tree-based

ensemble method (random forest) to be a more computationally efficient algorithm. From several experiments, it was found that Extra-Trees (ET) outperform other tree-based methods, including random forests (RF) [50]. The key difference between RF and ET are highlighted in [51]: (1) ET uses the entire data set for training a model (i.e., it does use tree bagging), whereas a random forest algorithm uses a bootstrap replica for training a machine learning model, and (2) ET randomly picks the best feature along with the corresponding value to split the node. Due to these main differences, ET is less likely to overfit a dataset and has reported better performance [48]. The ET splitting procedure for numerical attributes is detailed in [48]. ET relies on three main hyper-parameters that will be further optimized as detailed in Section 3.3. It consists of three factors: K is the number of randomly selected variables for splitting a node, n_{min} represents the minimum number of samples required for splitting an internal node, and M, the number of trees formed in the ensemble model [45].

2.2. Data Description

The historical dataset of HVAC energy consumption was gathered from a hotel building in Madrid, Spain. Social parameters (e.g., number of rooms booked, the total number of guests on a particular day) were also retrieved from the hotel's reservation system. Weather parameters were also collected from a nearby weather station. As hotels and restaurants do not exhibit clear energy consumption patterns as compared to other building types e.g., school buildings, which makes prediction a challenging task. Figure 2 shows electricity consumption of a school in Wales, UK and a hotel building in Madrid, Spain. It can be seen that there is a clear energy consumption pattern for the school building as the energy consumption is lower during the night and weekends. On the other hand, this is not the case for the hotel building.

For this study, one and a half year data was collected from the hotel's building management system (BMS) with a collection interval of 5 min. The collected dataset contained air temperature, relative humidity, wind speed, dew point air temperature, hour of the day, day of the week, month of the year, total rooms booked, total number of guests and value of HVAC energy consumption. Table 1 summarizes the variables used in the development of prediction models. Figure 3 shows the hourly HVAC consumption values of the studied building from 15 January 2015 to 15 January 2016. Training data is taken as 80% of the dataset such that the first 7680 samples were used in training and validation phases and the remaining 3290 samples composed the test dataset. For each algorithm training, the dataset was shuffled with identical random seeds to ensure that the exact same training set was used.

Table 1. Summary of the variables used for training and testing.

Input/Output Variables	Min	Max	Mean	Median
Outdoor air temperature (°C)	−5.5	40.5	14.65	13
Dew point air temperature (°C)	−12	20	4.84	5
Outdoor air Relative Humidity (%)	7.95	100	58.69	58.53
Wind speed (mph)	0	33.65	6.35	4.6
No. of rooms booked (-)	23	111	79.50	83
No of guests (-)	40	201	124.71	127
HVAC hourly energy consumption (kWh)	9.3	167.8	47.78	40.2
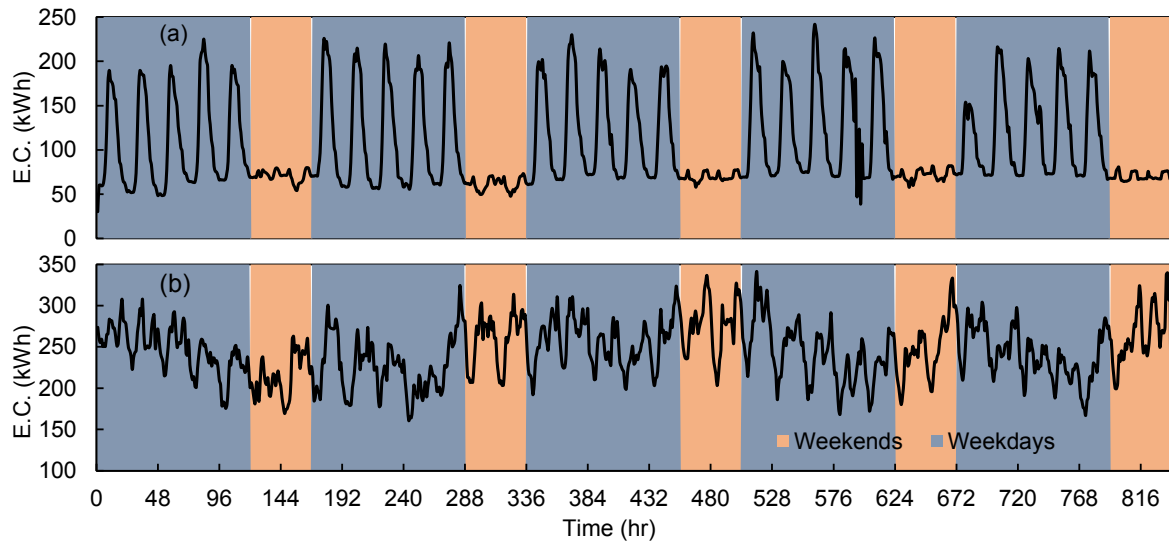
Figure 2. Building electricity consumption of (**a**) a school; (**b**) a hotel. Reproduced from Ahmad et al. [3], 2017.
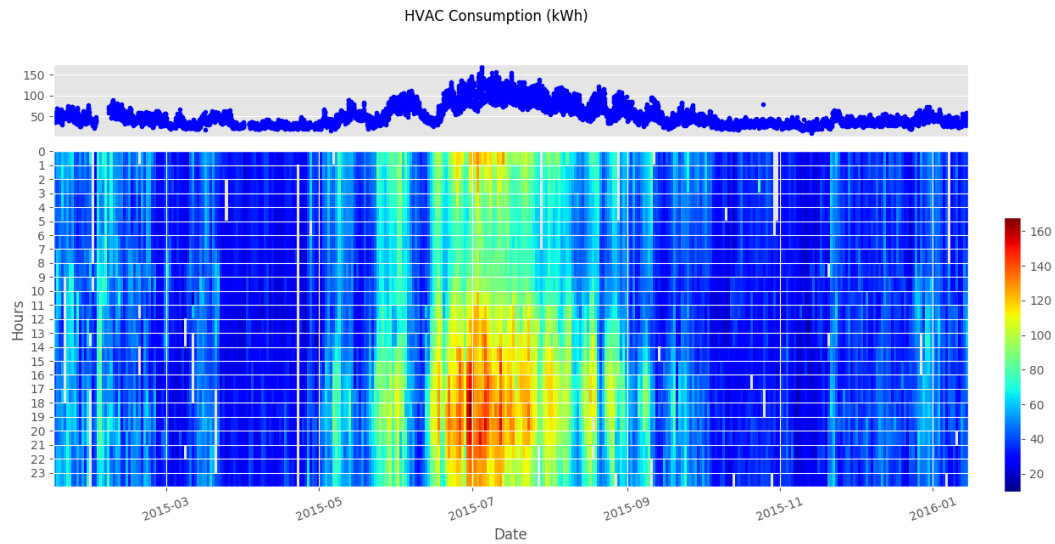
Figure 3. Actual hourly HVAC energy consumption. The data shown in the figure is from 15 January 2015 to 15 January 2016.

2.3. Model Evaluation Metrics

To evaluate the predictive performance of the developed models, four different metrics, i.e., root mean squared error (RMSE), mean absolute error (MAE), coefficient of determination (R²) and normalised root mean squared error (NRMSE) were used. Determination coefficient is used to measure the correlation between the actual and predicted HVAC energy consumption values. The remaining three metrics are described as below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{13}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{14}$$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \qquad (15)$$

where $\hat{y}_i$ is the predicted value, y_i is the actual value, N is the total number of samples, and y_{max} and y_{min} are the maximum and minimum values of actual HVAC energy consumption, respectively.

For this work, the implementation of extra trees and support vector regression included in the scikit-learn (a machine learning library for Python programming language) [52]. DHN was implemented in Python programming language. All developed models are encapsulated in a Python library to allow online data acquisition and predictions updates. It is worth mentioning that the model presented by Dieleman [53] was adapted for this paper. The models were trained and tested on personal computers (Intel Core i7 3.20 GHz with 32 GB of installed memory (for SVR and ET) and Intel Xeon 32 CPU 1.30 GHz with 64 GB of installed memory (for DHN models)).

3. Results

This section details the impact of different algorithms associated with hyper-parameters on a model's performance. A stepwise searching method was used to find optimal hyper-parameters values for SVR and ET models. For deep highway network models, a genetic algorithm based method was used for hyper-parameter tuning.

3.1. DHN Hyper-Parametric Tuning

A two stage hyper-parameter estimation was performed to find the best parameters for deep highway network models. At the first stage, a grid search was performed to highlight the best range of hyper-parameters. DHN models were trained by using a 5-fold cross-validation process, and setting a batch size inside epochs to 64 samples. Mean squared error was used as a criterion to evaluate training process. A Nesterov accelerated SGD (Stochastic Gradient Descent) update [54] was applied during the training phase. During the second stage, hyper-parameters of DHN were optimized using a genetic algorithm. The optimization problem and objective function of DHN hyper-parametric tuning can be represented by Equation (16):

$$\min_{\mathbf{x} \epsilon X} f(\mathbf{x}) \qquad (16)$$

where: $\mathbf{x}$ is the decision variables vector $[x_1, x_2, ..., x_n]$, n being the number of decision variables; and $f(\mathbf{x})$ is the objective function. In this case, eight decision variables were used i.e., number of neurons in each hidden layer, number of hidden layers, number of training epochs, learning rate, momentum, output layer activation function, bias value of gate layers neurons, and model inputs. Among the input variables of the dataset, preliminary studies showed that the most impacting variables were: outside temperature, relative humidity, number of guests and HVAC consumption values. The model inputs' decision variable states how many preceding values of each of these four input variables are used as input to the DHN model. Equation (16) is subject to ranges of values for each decision variable listed in Table 2.

The objective function of the problem is to minimize the normalized root mean squared error (NRMSE) on a testing dataset. The genetic algorithm optimisation was performed by considering 300 generations with a population size, mutation rate and crossover rate of 30, 0.1 and 0.5, respectively. The tournament selection method was used for selecting the five best individuals from a population. It is also worth mentioning that various previous values for input variables were also tried to improve the predictive performance of a DHN model.

Table 2. Possible values of hyper-parameters defining the space explored by the genetic algorithm.

Decision Variable	Possible Values
Number of hidden layer neurons	[5, 10, 20, 30]
Number of hidden layers	[1, 5, 10, 20, 50]
Epochs	[150, 200, 500, 1000, 2000]
Learning Rate	[0.005, 0.05, 0.01]
Momentum	[0.95, 0.99, 0.995]
Activation function	['VlReLu', 'Sigmoid', 'Linear', 'ReLu']
Bias	[−4.0, −3.0, −2.0, −1.0, 0.0, 1.0]
Model Inputs	[[1, 1, 1, 2], [1, 1, 1, 24], [24, 24, 1, 24]]

Note: VlReLu: very leaky rectified linear unit; ReLu: Rectified linear units.

Table 3 shows the 10 best performances of DHN models in terms of NRMSE. For each model, different performance metrics are applied. The results showed that, for the top 10 best models, the prediction accuracy was always greater than R^2 value of 0.84. It was found that several combinations of networks can achieve best performance, and generally no hyper-parameter drives the predictive performance of the models. Experimental results showed that best performance was obtained with an input dimension of 73 i.e., taking as input the previous 24 h of outdoor dry-bulb air temperature, air relative humidity and HVAC energy consumption and only the past value of the number of guests. However, it is worth mentioning that the increase in performance is small as compared to the increase in the complexity of the model. Results depict that a higher number of layers does marginally improve the performance; however, the best five performances were obtained by using one-layered networks. In order to further investigate the influence of model complexity, Table 4 shows the best performances achieved by the least complex models. The results clearly show that a model with only five input units and one layer, taking as input only one or two past values of input variables can achieve an NRMSE value of nearly 6%, with an error increase of 0.1% as compared to the best performing models.

Table 3. Top 10 results among the deep highway networks (DHN) models.

Model Inputs	Number of Neurons	Activation Function	Number of Layers	Training Epochs	Bias	Learning Rate	Momentum	R^2	NRMSE (%)
$[24, 24, 1, 24]$	30	ReLu	1	150	-3.0	0.01	0.95	0.84	6.007
$[1, 1, 1, 24]$	20	VlReLu	1	500	-3.0	0.05	0.95	0.8492	6.008
$[1, 1, 1, 24]$	10	Linear	1	500	-3.0	0.05	0.95	0.8490	6.011
$[1, 1, 1, 24]$	10	Linear	1	500	1.0	0.05	0.95	0.8490	6.013
$[1, 1, 1, 24]$	5	Linear	1	500	-3.0	0.05	0.95	0.8489	6.014
$[1, 1, 1, 24]$	20	VlReLu	50	1000	-2.0	0.005	0.95	0.8484	6.024
$[1, 1, 1, 24]$	5	VlReLu	1	1000	-3.0	0.05	0.95	0.8482	6.028
$[1, 1, 1, 24]$	5	Linear	50	1000	-2.0	0.005	0.95	0.8482	6.029
$[1, 1, 1, 24]$	20	Linear	20	500	-1.0	0.005	0.95	0.8481	6.029
$[1, 1, 1, 24]$	10	VlReLu	1	1000	-3.0	0.05	0.95	0.8481	6.030
$[1, 1, 1, 24]$	5	VlReLu	50	2000	-2.0	0.05	0.95	0.8481	6.030

The "Model inputs" array is represented as [Outdoor air temperature, Relative humidity, No. of guests, previous hours HVAC energy consumption].

Table 4. Best results among the DHN models with minimal complexity.

Model Inputs	Number of Neurons	Activation Function	Number of Layers	Training Epochs	Bias	Learning Rate	Momentum	R^2	NRMSE (%)
$[1, 1, 1, 2]$	10	VlReLu	1	1000	0.0	0.05	0.95	0.8444	6.10
$[1, 1, 1, 2]$	10	ReLu	1	150	-2.0	0.05	0.99	0.8416	6.16
$[1, 1, 1, 2]$	30	VlReLu	1	200	-4.0	0.05	0.95	0.8431	6.13
$[1, 1, 1, 24]$	5	Linear	1	500	-3.0	0.05	0.95	0.8489	6.014
$[1, 1, 1, 24]$	5	Linear	1	200	-2.0	0.005	0.95	0.8445	6.10
$[1, 1, 1, 24]$	5	ReLu	1	1000	1.0	0.01	0.95	0.8425	6.14

3.2. SVR Hyper-Parametric Tuning

For support vector regression, penalty parameter (C) and radius (ϵ) are two important tunable hyper-parameters to achieve better predictive performance. A small value of C results in a small weight on the training dataset, which could result in larger prediction errors on the testing dataset. This means that the trained model will under-fit the training data [36]. However, a larger value of C would result in over-fitting the training dataset. Larger values of penalty parameter (C) will also reset the objective back to minimising the empirical risk only. On the contrary, larger values of C means a larger range of the value of support vectors, which means more data points can be selected [35]. ϵ is indirectly related to the number of support vectors, and a larger value of ϵ can result into fewer number of support vectors machines. In addition, it should be noted that a too large value of ϵ can reduce the predictive accuracy of the model [36]. In order to maximize the performance of SVR model, we tuned these two hyper-parameters.

In this paper, the stepwise searching method was used to study the performance of developed SVR models by varying parameter settings for C and ϵ. The stepwise searching method has been previously used by many researchers e.g., [35–37]. In literature, there are many methods for tuning the hyper-parameters of machine learning models, grid search being the most frequently used. However, it is computationally extensive technique. As grid search computes performance at all pairs of ϵ and C to get the performance surface, it has lower efficiency [35]. In stepwise search, we first conducted the search by fixing the value of ϵ to find C. In the next step, the first result of C was fixed to find ϵ. It is worth mentioning that stepwise search may result into sub-optimal hyper-parameters as it is assumed that all hyper-parameters are independent from each other. As a first step, the value of ϵ was fixed to 0.1 and varied C over the range between 2^{-7} and 2^7 to train an SVR model over the training dataset. The resulting models were then used to predict on a testing dataset to calculate performance metrics. From results, it was found that a model's performance increases with an increase of C. Initially, the performance of the SVR model increased with an increase of C. However, with higher values of C, the performance of the SVR model was slightly increased. The performance started to decline for values of C higher than 2^6. The higher values of C were also over-fitting the training dataset. A value of $C = 2^{15}$ was also tried and it was concluded that higher values reduced the performance by over-fitting the training dataset. In addition, it was found that models trained with higher values of C are computationally expensive to train. Therefore, from results, a value of 2^5 was selected for C.

After setting the value of C to 2^5, various values of ϵ were tried to find its optimal value. From the results, it was found that smaller values of ϵ did not have a significant influence on the performance on SVR model. The performance significantly reduced for values larger than 4. From results, a value of 2 was chosen for ϵ. Tables 5 and 6 show the results of different experiments for select C and ϵ.

Table 5. Results of different *C*, where $\epsilon = 0.1$.

C	R^2 (–)	RMSE (kWh)	MAE (kWh)
2^{-7}	-0.336	12.501	10.601
2^{-6}	-0.307	12.362	10.448
2^{-5}	-0.259	12.132	10.188
2^{-4}	-0.176	11.726	9.752
2^{-3}	0.021	10.700	8.795
2^{-2}	0.380	8.518	6.912
2^{-1}	0.670	5.926	4.658
2^0	0.801	4.821	3.644
2^1	0.829	4.472	3.316
2^2	0.839	4.343	3.188
2^3	0.843	4.288	3.123
2^4	0.844	4.274	3.102
2^5	0.844	4.269	3.091
2^6	0.844	4.268	3.088
2^7	0.844	4.269	3.087

Table 6. Results of different ϵ, where $C = 2^5$.

ϵ	R^2 (–)	RMSE (kWh)	MAE (kWh)
2^{-10}	0.84427	4.2675	3.0924
2^{-9}	0.84428	4.2673	3.0922
2^{-8}	0.84429	4.2671	3.0920
2^{-7}	0.84427	4.2674	3.0922
2^{-6}	0.84421	4.2682	3.0928
2^{-5}	0.84416	4.2690	3.0927
2^{-4}	0.84418	4.2686	3.0916
2^{-3}	0.84426	4.2675	3.0903
2^{-2}	0.84453	4.2639	3.0881
2^{-1}	0.84467	4.2619	3.0849
2^0	0.84477	4.2606	3.0877
2^1	0.84536	4.2525	3.0896
2^2	0.84027	4.3219	3.2041
2^3	0.79878	4.8508	3.8013
2^4	0.55341	7.2267	6.1642
2^5	-0.44957	13.0200	11.2723

3.3. ET Hyper-Parametric Tuning

For Extra trees algorithm, number of trees (*M*), number of samples required for splitting a node ($n_{\min}$) and attribute selection strength parameter (*K*) are the three important hyper-parameters. The parameter *K* represents the size of the random subsets of features to consider when splitting a node, and can be selected in the range $[1, ..., n]$, where *n* is the total number of features. The total number of trees in the forest is represented by *M*; for our case, we fixed *M* to 1000 trees. Larger values of smoothing parameter ($n_{\min}$) would result in smaller trees, higher bias and smaller variance [48]. For hyper-parameter tuning, this parameter was varied in the range $[2, ..., 10]$ to investigate its influence on model's accuracy. From results, it was found that changing $n_{\min}$ did not yield a significant accuracy improvement on the hotel's HVAC energy consumption dataset. For this problem, a value of 3 was chosen for $n_{\min}$ as it resulted in slightly better performance than the default value used in the literature (i.e., 2). We also studied the influence of parameter *K* on model's accuracy and varied the parameter in the interval $[1, ..., n]$. For a value of $K = 1$, the splits are chosen in totally independent way of the output variable. On the other hand, a value of $K = n$ (total number of features), the attributes' choice is not explicitly randomized and the effect of randomization will only act through the choice of cut-points [48]. We varied *K* over its range and found that a value of $K = 4$ slightly improved the

model's accuracy. Results demonstrated that a value of $K = 1$ resulted in an under-fitted model with an R^2 value of 0.7485. The influence of tree depth on predictive accuracy shows that deeper trees resulted in better performance. The performance started to deteriorate for d_{max} greater than 10. The trees with $d_{max} = 1$ resulted in higher values of RMSE, MAE and MSE, and lower value of R^2. From these results, it is clear that, on the studied dataset, extremely randomized trees' performance was more influenced by parameter d_{max} instead of n_{min} and K. This may vary from dataset to dataset; however, for most of the cases, default values of the parameters may result in acceptable performance. Table 7–9 show the results of various experiments for selecting ET hyper-parameters.

Table 7. Results of different n_{min}, where $K = n$ and $M = 1000$.

n_{min}	R^2 (–)	RMSE (kWh)	MAE (kWh)
2	0.819	4.601	3.405
3	0.822	4.564	3.377
5	0.828	4.485	3.312
7	0.832	4.427	3.266
10	0.837	4.372	3.223

Table 8. Results of different K, where $n_{min} = 3$ and $M = 1000$.

K	R^2 (–)	RMSE (kWh)	MAE (kWh)
1	0.7485	5.423	4.200
2	0.8091	4.724	3.555
3	0.8226	4.555	3.385
4	0.8246	4.529	3.353
5	0.8219	4.564	3.377

Table 9. Results of different d_{max}, where $n_{min} = 3$, $K = 4$ and $M = 1000$.

d_{max}	R^2 (–)	RMSE (kWh)	MAE (kWh)
1	−0.3242	12.445	10.578
3	0.5323	7.396	6.128
5	0.7629	5.265	4.199
7	0.8281	4.484	3.413
9	0.8424	4.292	3.184
10	0.8427	4.288	3.167
15	0.8353	4.389	3.227
20	0.8262	4.508	3.331

4. Comparison and Discussion

Predictive performance of SVR, ET and DHN models are nearly comparable. Figure 4 illustrates the plot of actual hourly HVAC energy consumption vs. predicted energy consumption by the ET model. Models' ability to accurately predict energy consumption is clearly illustrated by the level of linear relationship between predicted and measured values in Figure 4b. The figure shows the strong nonlinear mapping generalisation capabilities of the model, and it can be effectively used as prediction models for decision-making processes. Table 10 shows a comparison of models' performance for both training and testing datasets. According to the results, DHN performed marginally better as compared to the other two developed models. For all three models, the R^2 value is higher than 0.84 and RMSE values were in the range of 3.08 and 4.28 for both training and testing datasets. From these results, it can be concluded that the developed models have the capabilities to accurately predict the hourly HVAC energy consumption.

From Table 10, it is demonstrated that all developed models have nearly comparable performances and could be equally used to accurately predict the hourly HVAC energy consumption. It was expected that the prediction accuracy will be significantly enhanced by using a deep learning algorithm due to

the 'deep' property of the network. However, this was not the case, as DHN performed marginally better than the other two ML algorithms. From tuning the hyper-parameters of DHN, it was found that the network complexity (i.e., increasing network's number of layers, neurons and previous hours values of input variables) did not significantly improve the performance of the model. The possible reasons could be fourfold:

1. The obtained performance is optimal and no further improvement could be achieved;
2. The complexity may not have been increased enough to show significant changes in the performance of the model;
3. Some variables of interest may not have been taken into account;
4. The historical data used in this study is not sufficient to ensure the reliable training of a deep learning models' deep highway network in our case.

Table 11 presents a comparison of predicted energy consumption by all developed models and actual HVAC energy consumption data. The mean values, which shows the central tendency within a data sample, of all models' output closely resemble the mean value of actual data. Standard deviation values, which are used to quantify the amount of variations, of all models are slightly lower than the actual data. Median values of all models closely match with the actual data. The "tailedness" and "asymmetry" of the probability distribution of a real-valued random variable is measured by Skewness and Kurtosis. It was found that DHN model has slightly lower Kurtosis value as compared to the other models and actual dataset. Minimum and maximum values are used to identify outliers in the dataset, it was found that ET has comparatively higher minimum value as compared to the actual dataset. The results also show that all models have under predicted some of the higher values of HVAC energy consumption. This might be due to the fact that those values were under-represented in the training dataset.

Table 10. Comparison of extremely randomized trees (ET), support vector regression (SVR) and deep highway network (DHN) models.

| Model	Training Dataset	Testing Dataset		
	RMSE (kWh)	R^2 (–)	RMSE (kWh)	MAE (kWh)
ET	4.284	0.8427	4.288	3.167
SVR	4.252	0.8453	4.253	3.090
DHN	3.087	0.8491	4.200	3.027
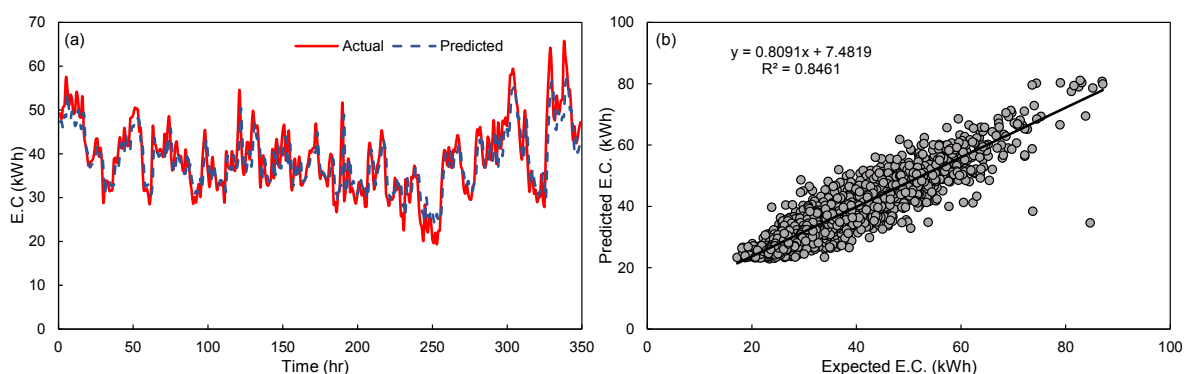
Figure 4. Results from extremely randomized trees model. (**a**) comparison between actual and predicted energy consumption from the extremely randomized trees (ET) model; (**b**) scatter chart illustrating the relationship between predicted and actual energy consumption.

Table 11. Statistical measures on testing dataset for DHN, ET and SVR.

Factor/Variable	Actual E.C. Data	DHN	ET	SVR
Mean	37.08	37.16	37.48	37.09
Median	35.2	35.46	35.88	35.68
Standard Deviation	10.82	9.88	9.51	9.82
Sample Variance	116.98	97.68	90.51	96.51
Kurtosis	1.14	0.62	1.10	0.97
Skewness	0.96	0.82	0.93	0.88
Range	69.90	64.30	58.25	63.88
Minimum	17.2	16.84	22.87	18.31
Maximum	87.1	81.14	81.12	82.19
Sum	121,981.7	122,245.60	123,308.75	122,015.85

Figure 5 shows the violin plot for probability distribution for extremely randomized model during different hours of the testing dataset. A violin plot is similar to box-and-whisker plot, a box plot indicates variability outside the upper and lower quartiles with a box and whiskers. In a violin plot, the full probability density function in a mirrored form is presented on a vertical axis [55]. The white circle in the middle of the plot indicates median, and the upper and lower ends of the box inside the violin plot indicates the quartiles. In a violin plot, a long slender shape (e.g., such as for hour 9:00 a.m in Figure 5) indicates a large variation and therefore uncertainty in prediction. Short, wide shapes (e.g., hour 2:00 a.m in Figure 5) indicates low variance and concentrated probability mass. The violin charts in Figure 5 show that there is large variation for prediction error during the early morning and late afternoon (4:00 p.m.–5:00 p.m.).
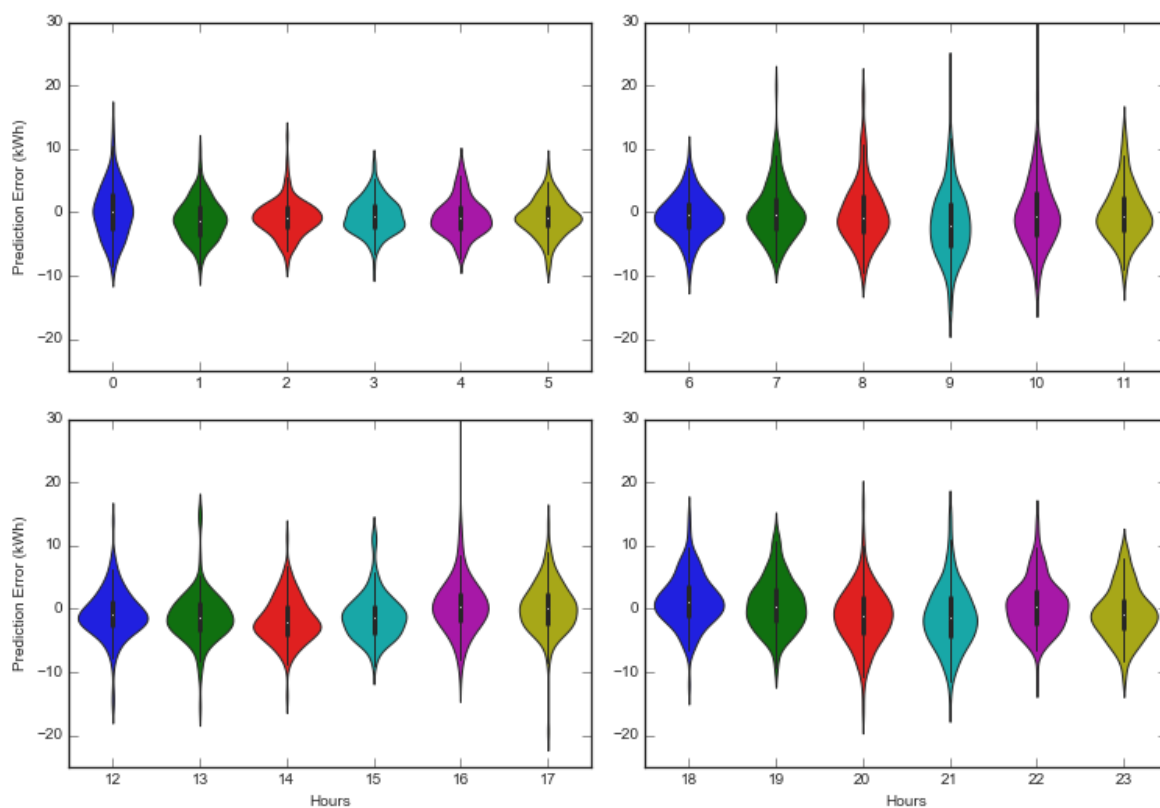
Figure 5. Violin plot showing probability distribution shapes for ET model during different testing hours, with quartiles and median indicated.

5. Conclusions

Deep learning has shown promising learning and prediction capabilities for different applications. On the other hand, ensemble-based methods were recently developed to overcome problems in traditional methods (e.g., decision trees). This study proposed three machine learning methods to predict hourly HVAC energy consumption of a hotel. Notably, it presented the use of deep highway network—a deep learning method, extra trees—a tree-based ensemble method and a most widely used support vector regression. The paper compared their performance in terms of accuracy and computational efficiency. The analysis performed showed that all three methods have nearly comparable performances. Therefore, the proposed models can achieve accurate and reliable hourly prediction of HVAC energy consumption. The developed models can be used for demand-side management, optimal control and scheduling of HVAC systems, fault detection and diagnosis and predicting behaviour of energy system to mitigate potential uncertainties in smart grids. One of the aspects of this research was to find an answer as to whether deep learning is suitable for predicting high-resolution energy consumption or not. As DHN performed marginally better than the other two studied algorithms; for this problem, it may not be a favourable solution (considering the effort and time required number of different hyperparameters). As from DHN results, it was found that the network complexity did not significantly improve the model's performance. Therefore, some remaining aspects to reflect on are to investigate the possible reasons in more details i.e., whether (1) the obtained performance is optimal and no further improvements could be achieved; (2) the network complexity has been increased enough to show significant changes in the performance; (3) some variables of interest are missing; and (4) enough historical data is used to ensure the reliable training of a deep learning model. These aspects will be further investigated in the future.

References

1. Ahmad, M.W.; Mourshed, M.; Mundow, D.; Sisinni, M.; Rezgui, Y. Building energy metering and environmental monitoring— A state-of-the-art review and directions for future research. *Energy Build.* **2016**, *120*, 85–102. [CrossRef]
2. Ponta, L.; Raberto, M.; Teglio, A.; Cincotti, S. An Agent-based Stock-flow Consistent Model of the Sustainable Transition in the Energy Sector. *Ecol. Econ.* **2018**, *145*, 274–300. [CrossRef]
3. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* **2017**, *147*, 77–89. [CrossRef]
4. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
5. Dascalaki, E.; Balaras, C.A. XENIOS—A methodology for assessing refurbishment scenarios and the potential of application of RES and RUE in hotels. *Energy Build.* **2004**, *36*, 1091–1105. [CrossRef]
6. Sloan, P.; Legrand, W.; Chen, J.S. (Eds.) Chapter 2—Energy Efficiency. In *Sustainability in the Hospitality Industry*; Butterworth-Heinemann: Boston, MA, USA, 2009; pp. 13–26. [CrossRef]
7. de Wilde, P. The gap between predicted and measured energy performance of buildings: A framework for investigation. *Autom. Constr.* **2014**, *41*, 40–49. [CrossRef]

8.   Meyers, R.J.; Williams, E.D.; Matthews, H.S. Scoping the potential of monitoring and control technologies to reduce energy use in homes. *Energy Build.* **2010**, *42*, 563–569. [CrossRef]

9.   Reddy, T.A. Literature Review on Calibration of Building Energy Simulation Programs: Uses, Problems, Procedures, Uncertainty, and Tools. *ASHRAE Trans.* **2006**, *112*, 226.

10.   Raftery, P.; Keane, M.; O'Donnell, J. Calibrating whole building energy models: An evidence-based methodology. *Energy Build.* **2011**, *43*, 2356–2364. [CrossRef]

11.   Ahmad, M.W.; Mourshed, M.; Yuce, B.; Rezgui, Y. Computational intelligence techniques for HVAC systems: A review. *Build. Simul.* **2016**, *9*, 359–398. [CrossRef]

12.   Manfren, M.; Aste, N.; Moshksar, R. Calibration and uncertainty analysis for computer models—A meta-model based approach for integrated building energy simulation. *Appl. Energy* **2013**, *103*, 627–641. [CrossRef]

13.   Brea, J.; Senn, W.; Pfister, J.P. Matching Recall and Storage in Sequence Learning with Spiking Neural Networks. *J. Neurosci.* **2013**, *33*, 9565–9575. [CrossRef] [PubMed]

14.   Diehl, P.U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.C.; Pfeiffer, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In Proceedings of the 2015 IEEE International Joint Conference on Neural Networks (IJCNN), Killarney, UK, 12–16 July 2015; pp. 1–8.

15.   He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. arXiv **2015**, *arXiv:1502.01852*.

16.   LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]

17.   Gers, F. A.; Schmidhuber, J.; Cummins, F.A. Learning to forget: Continual prediction with LSTM. In Proceedings of the 9th International Conference on Artificial Neural Networks: ICANN '99, Edinburgh, UK, 7–10 September 1999.

18.   Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *arXiv* **2015**, arXiv:1506.04214.

19.   Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115. [CrossRef] [PubMed]

20.   McCann, B.; Keskar, N.S.; Xiong, C.; Socher, R. The natural language decathlon: Multitask learning as question answering. *arXiv* **2018**, arXiv:1806.08730.

21.   Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; others. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529. [CrossRef] [PubMed]

22.   Yıldırım, Ö.; Pławiak, P.; Tan, R.S.; Acharya, U.R. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Comput. Biol. Med.* **2018**, *102*, 411–420. [CrossRef] [PubMed]

23.   He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.

24.   Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *a*rXiv **2017**, arXiv:1610.02357.

25.   Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks. *arXiv* **2015**, arXiv:1505.00387.

26.   Mocanu, E.; Nguyen, P.H.; Gibescu, M.; Kling, W.L. Deep learning for estimating building energy consumption. *Sustain. Energy Grids Netw.* **2016**. [CrossRef]

27.   Marino, D.L.; Amarasinghe, K.; Manic, M. Building Energy Load Forecasting using Deep Neural Networks. In Proceedings of the 42nd IEEE Annual Conference of the IEEE Industrial Electronics Society (IECON), Florence, Italy, 24–27 October 2016.

28.   Li, C.; Ding, Z.; Zhao, D.; Zhang, J.Y.G. Building Energy Consumption Prediction: An Extreme Deep Learning Approach. *Erngies* **2017**, *10*, 1525. [CrossRef]

29.   Mouraud, A. Innovative time series forecasting: auto regressive moving average vs deep networks. *Entrep. Sustain. Issues* **2017**, *4*, 282–293. [CrossRef]

30.   Mocanu, E.; Mocanu, D.C.; Nguyen, P.H.; Liotta, A.; Webber, M.E.; Gibescu, M.; Slootweg, J. On-Line Building Energy Optimization using Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1707.05878.

31.   Fan, C.; Xiao, F.; Zha, Y. A short-term building cooling load prediction method using deep learning algorithms. *Appl. Energy* **2017**, *195*, 222–233. [CrossRef]

32.   Liang, J.; Du, R. Model-based Fault Detection and Diagnosis of HVAC systems using Support Vector Machine method. *Int. J. Refrig.* **2007**, *30*, 1104–1114. [CrossRef]

33. Mohandes, M.A.; Halawani, T.O.; Rehman, S.; Hussain, A.A. Support vector machines for wind speed prediction. *Renew. Energy* **2004**, *29*, 939–947. [CrossRef]

34. Esen, H.; Inalli, M.; Sengur, A.; Esen, M. Modeling a ground-coupled heat pump system by a support vector machine. *Renew. Energy* **2008**, *33*, 1814–1823. [CrossRef]

35. 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]

36. Li, Q.; Meng, Q.; Cai, J.; Yoshino, H.; Mochida, A. Applying support vector machine to predict hourly cooling load in the building. *Appl. Energy* **2009**, *86*, 2249–2256. [CrossRef]

37. Li, Q.; Meng, Q.; Cai, J.; Yoshino, H.; Mochida, A. Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks. *Energy Convers. Manag.* **2009**, *50*, 90–96. [CrossRef]

38. Li, X.; Deng, Y.; Ding, L.; Jiang, L. Building cooling load forecasting using fuzzy support vector machine and fuzzy C-mean clustering. In Proceedings of the 2010 International Conference On Computer and Communication Technologies in Agriculture Engineering (CCTAE), Chengdu, China, 12–13 June 2010; Volume 1, pp. 438–441.

39. Yu, Z.; Haghighat, F.; Fung, B.C.; Yoshino, H. A decision tree method for building energy demand modeling. *Energy Build.* **2010**, *42*, 1637–1646. [CrossRef]

40. Hong, T.; Koo, C.; Jeong, K. A decision support model for reducing electric energy consumption in elementary school facilities. *Appl. Energy* **2012**, *95*, 253–266. [CrossRef]

41. Hong, T.; Koo, C.; Park, S. A decision support model for improving a multi-family housing complex based on CO2 emission from gas energy consumption. *Build. Environ.* **2012**, *52*, 142–151. [CrossRef]

42. 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]

43. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.

44. Ahmad, M.W.; Reynolds, J.; Rezgui, Y. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *J. Clean. Prod.* **2018**, *203*, 810–821. [CrossRef]

45. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression. *Energy* **2018**, *164*, 465–474. [CrossRef]

46. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 630–645.

47. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011, pp. 315–323.

48. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]

49. Scalzo, F.; Xu, P.; Asgari, S.; Bergsneider, M.; Hu, X. Regression analysis for peak designation in pulsatile pressure signals. *Med. Biol. Eng. Comput.* **2009**, *47*, 967–977. [CrossRef] [PubMed]

50. Nattee, C.; Khamsemanan, N.; Lawtrakul, L.; Toochinda, P.; Hannongbua, S. A novel prediction approach for antimalarial activities of Trimethoprim, Pyrimethamine, and Cycloguanil analogues using extremely randomized trees. *J. Mol. Gr. Model.* **2017**, *71*, 13–27. [CrossRef] [PubMed]

51. John, V.; Liu, Z.; Guo, C.; Mita, S.; Kidono, K., Real-Time Lane Estimation Using Deep Features and Extra Trees Regression. In *Image and Video Technology: 7th Pacific-Rim Symposium, PSIVT 2015, Auckland, New Zealand, November 25–27, 2015, Revised Selected Papers*; Bräunl, T.; McCane, B.; Rivera, M.; Yu, X., Eds.; Springer: Cham, Switzerland, 2016; pp. 721–733. [CrossRef]

52. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; others. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

53. Dieleman S. Highway Networks. Available online: https://github.com/Lasagne/Lasagne/blob/highway_example/examples/Highway%20Networks.ipynb (accessed on 5 December 2018).

54. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1139–1147.

55. Carstens, H.; Xia, X.; Yadavalli, S. Low-cost energy meter calibration method for measurement and verification. *Appl. Energy* **2017**, *188*, 563–575. [CrossRef]