Developments in the Built Environment 5 (2021) 100037

Contents lists available at ScienceDirect

Developments in the Built Environment

journal homepage: www.editorialmanager.com/dibe/default.aspx

Energy consumption prediction by using machine learning for smart building: Case study in Malaysia

Mel Keytingan M. Shapi [a], Nor Azuana Ramli [b,*], Lilik J. Awalin [c]

[a] Electrical Engineering Section, Universiti Kuala Lumpur British Malaysian Institute, Gombak, Selangor, Malaysia
[b] Centre for Mathematical Sciences, Universiti Malaysia Pahang, Gambang, Pahang, Malaysia
[c] Sekolah Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya, Indonesia

ARTICLE INFO

Keywords:
Building energy management system
Machine learning
Microsoft Azure
Energy consumption
Prediction

ABSTRACT

Building Energy Management System (BEMS) has been a substantial topic nowadays due to its importance in reducing energy wastage. However, the performance of one of BEMS applications which is energy consumption prediction has been stagnant due to problems such as low prediction accuracy. Thus, this research aims to address the problems by developing a predictive model for energy consumption in Microsoft Azure cloud-based machine learning platform. Three methodologies which are Support Vector Machine, Artificial Neural Network, and k-Nearest Neighbour are proposed for the algorithm of the predictive model. Focusing on real-life application in Malaysia, two tenants from a commercial building are taken as a case study. The data collected is analysed and pre-processed before it is used for model training and testing. The performance of each of the methods is compared based on RMSE, NRMSE, and MAPE metrics. The experimentation shows that each tenant's energy consumption has different distribution characteristics.

1. Introduction

Recently, smart building concept has been adapted more frequently as an initiative to create an intelligent space area by taking advantage of the rapid development of computational and communication architecture (Cheng and Kunz, 2009). This concept is not only limited to Malaysia but other countries as well. General public understanding of smart building concept rotates on the idea of automated process, which is able to automatically control the building's operation through the usage of instrumentation measures and microcontrollers in two-way communication (Qolomany et al., 2019). Other than automated control, a smart building also consists of an intelligent system which provides energy consumption forecasts as an energy efficiency initiative. This is due to its advantage of yielding economical savings and as a sustainable approach for energy management to minimize energy wastage (Xu et al., 2018).

A smart energy consumption forecasting is important, especially for buildings as buildings' energy usage is increasing and almost reaches 40% of primary energy use in developed countries (Berardi, 2015). In Malaysia alone, energy consumption has been increase gradually due to the growth of population. The growth of population lead to the increasing of energy demand in this country and have been estimated to reach 116 million tons of oil equivalents (mtoe) by this year. Energy provided in Malaysia is influenced by the main fossil fuel sources which included coal, natural gas and fuel oil. Buildings which including commercial, residential and industrial in our country utilises a total of 48% of the electricity that have been created (Hassan et al., 2014). The increasing of energy consumptions towards buildings from day to day create enforcement to this country in managing and reducing the energy consumption as much as possible in order to improve energy efficiency.

This study is a continuing research from our previous work where previously statistical analysis and k-nearest neighbour (k-NN) method were proposed as the methodologies and SPSS was used as the platform (Mazlan et al., 2020). In our previous study, only k-NN was proposed as the method to predict energy consumption. It is difficult to know whether the method proposed was the best since there is no comparison had been made. Hence, another two methods from machine learning are added in this study.

This research has utilised Microsoft Azure Machine Learning Studio, which is a web service solution for the development of prediction model. Starting from data analysis until performance evaluation, AzureML has been successfully employed for the implementation of energy demand forecasting. A major advantage of using Microsoft Azure over SPSS is it is user friendly and easy to use even the user only has basic knowledge in cloud computing and machine learning. One of the distinguishing

* Corresponding author.
E-mail addresses: mel.keytingan15@s.unikl.edu.my (M.K.M. Shapi), azuana@ump.edu.my (N.A. Ramli), lilik.j.a@stmm.unair.ac.id (L.J. Awalin).

https://doi.org/10.1016/j.dibe.2020.100037
Received 3 July 2020; Received in revised form 30 November 2020; Accepted 30 November 2020
Available online 9 December 2020

features of AzureML was its ability to manoeuvre through a visualization workflow. The workflow that was conducted inside the environment was manipulated through a graphical drag and drop procedure. Other than that, parsing data for experiment was simply done by joining of modules. Additionally, the platform also supports script packages and algorithms written in external programming language, particularly R programming.

The development of energy consumption predictive models that use statistical analysis and learning methodology possesses several significant challenges. Attewell and Monaghan (Attewel and Monaghan, 2015) described that statistical prediction is restricted especially in the case of a large dataset with several features, as it requires a higher computational power for modeling. Other than that, the statistical prediction method itself is comparably weak as it performs better only in the case of stationary time series and high similarity of consumption level (Abdul Karim and Alwi, 2013). Newsham and Birt (2010) also deduced that time series analysis for electricity consumption forecast performance was unsatisfactory due to variables in the chosen attributes. Moreover, the traditional development of a predictive model is usually based on the trend of maximum demand (kW) consumption only, which is known as a time series method (Xiangyu et al., 2019). The model development would neglect other electrical parameters such as reactive power changes, which causes the model to be trained only with historical data of maximum demand value. On the contrary, the inclusion of other features of electrical power data would improve the energy consumption prediction (Wei et al., 2019). Therefore, machine learning methodology is preferable when developing a predictive model of electrical consumption.

Changing from the statistical method to machine learning method itself does not solve all the problems with energy consumption prediction. Missing data that was present on a set of data was well known to cause the performance of the predictive model to deteriorate (Ahmad et al., 2016; Nugroho and Surendro, 2019). This missing data exists usually due to the interconnectivity or sensor problem which is the main complication for smart building energy metering (Ahmad et al., 2016). Additionally, the development of the machine learning model should utilise a cloud-based service to reduce the dependency of prediction on the hardware specifications (Mateev, 2019). Comprehensively, the three critical areas of energy consumption forecast discussed are machine learning prediction methodology, handling of missing data and employment of cloud-based prediction modeling platform which will be the basis of this research.

As the main objective of this research is to develop an energy consumption predictive model for smart commercial building by using several machine learning methods in a cloud-based machine learning platform, this research focuses more on the accuracy of the methodology applied in predicting energy consumption. Advances in machine learning studies have a tremendous impact on the field of smart building energy management as it is crucial to reduce energy consumption of various types of building from residential buildings to industrial buildings. Therefore, this study is essential to the following parties such as Ministry of Green Energy and Water (KETTHA) and Malaysia Green Technology Corporations in their determination to analyse energy consumption level of the existing building. It is also helpful to industrial manufacturing company in predicting electrical loading on their system for long range projection, mapping of capacity versus demand and for factory growth projection. Last but not least, academician majoring in engineering field to understand the integration of data science and analytics in engineering projects and higher education students to explore the services and possibility of utilising Microsoft Azure Machine Learning Studio for various projects.

2. Literature review

2.1. Machine learning prediction methodology

Following the aforesaid problem, this research addresses its challenges by conducting prediction modeling through the evaluation of historical power data. This method utilises a data-driven approach as described by Corgnati et al. (2013) whereby the input (regressor variables) and output variables (response) are known. Based on this data, system parameters will be estimated and thus, a mathematical model could be generated. Several previous studies have analysed the data-driven machine learning approach. Fu et al. (2015) proposed using one of ML algorithms which is Support Vector Machine (SVM) to predict the load at a building's system-level (air conditioning, lighting, power, and others) based on weather predictions and hourly electricity load input. Overall, SVM method managed to predict the total electricity load with root mean square error (RMSE) of 15.2% and mean bias error (MBE) of 7.7%. Findings by Valgaev et al. (2016) proposed a power demand forecast using k-Nearest Neighbour (k-NN) model at a smart building as part of the Smart City Demo Aspern (SCDA) project. The k-NN forecasting method was approached using a set of historical observations (daily load curves) and their successors. The k-NN method is good at classifying data but limited in forecasting future value as it only identifies similar instances in large feature space. Therefore, it must be complemented with temporal information identification whereby the prediction will be made for the next 24 h during workdays.

Five methods of machine learning techniques were used for short-term load forecasting by El Khantach et al. (El Khantach et al., 2019) with an initial decomposition of the historical data done periodically into time series of each hour of the day, which finally constituted 24-time series that represented every past hour. The five machine learning methods used are multi-layer perceptron (MLP), support vector machine (SVM), radial basis function (RBF) regressor, REPTree, and Gaussian process. The experimentation was done based on data derived from the Moroccan electrical load data. The result showed that MLP method came out as the most accurate with MAPE percentage of 0.96 while SVM came second and although far from the result of MLP, it was still better than the rest. Although the prediction of energy consumption usually uses a classification-based machine learning method, prediction could also be made based on the regression method as studied by González-Briones et al. (2019). The research constructed a predictive model by analysing the historical data set using Linear Regression (LR), Support Vector Regression (SVR), Random Forest (RF), Decision Tree (DT) and k-Nearest Neighbour (k-NN). The parameters of the research used one day-before electricity consumption (kWh) as an additional attribute. The results showed that LR and SVR models had the best performance with 85.7% accuracy.

2.2. Management of missing data

Techniques in handling missing data have been vastly studied before and methodologies have been deduced. There are two types of methodology that are removing the portion of the data which has missing value and imputation method which is based on close estimation (Hegde et al., 2019). The first method which omit the missing part of data is not feasible as this causes valuable information to be removed (Manly and Wells, 2015). Without the data, a biased estimation would be made. Therefore, the imputation method is a preferable technique. Newgard and Lewis (2015) presented several imputation techniques such as Mean Value Imputation, Last Observation Carried Forward, Maximum Likelihood Estimate (MLE) and Multiple Imputation (MI). The mean value imputation basically substitutes the missing data with the mean value of the dataset. However, this method is not suitable for data which is not strictly random as it will introduce inequality in the data (Kang, 2013). Another method presented was Last Observation Carried Forward, in which imputation is made for historical data that was collected through (Newgard and Lewis, 2015).

The more advanced methods presented were Multiple Imputation (MI) and Maximum Likelihood Estimate (MLE) (Newgard and Lewis, 2015). The Multiple Imputation method substitutes the missing data by gradually supplanting the missing data for every iteration made. This method utilises statistical analysis based on observed data to handle the uncertainty that is introduced by the missing portion. An example of a popular MI method is Multiple Imputation Using Chained Equations (MICE) (Azur et al., 2011). Maximum Likelihood Estimate conducts substitution through assumption made by initially identifying the parameters and boundaries based on the distribution of the data. The imputation would then be made based on the assumed parameters. This method of imputation was employed by Probabilistic Principal Component Analysis (PPCA). Both advanced imputation methods have been compared by Hegde et al. (2019) in which imputation method was made on sampled dataset consisting of 87 numeric-converted categorical variables and 29 continuous variables. The study used RMSE metrics to evaluate imputation technique performance. From the research, the PPCA method showed a much promising result compared to MICE, in which 65% of data variables were successfully imputed by PPCA and only 38% correct imputation by MICE. This was further supported by Schmitt et al. (Schmitt et al., 2015) wherein the research compared the performance of six imputation methods including PPCA and MICE on a real dataset of various sizes. The result showed that MICE managed to perform well in a small dataset, but in a large dataset case, the MICE method performed poorly.

2.3. Employment of cloud-based prediction modeling

There are various available cloud-based prediction modeling platforms which are able to support machine learning process with additional capabilities to analyse big data and streaming data. When selecting the preferable machine learning tool and platform, several important factors need to be considered such as ascendable, pace, scope, practicability, flexibility, and programming language (Landset et al., 2015). Machine learning platform and tool are observed to be frequently utilised with big data techniques for real-time analytics. Review of previous papers on utilising these ML platforms with big data analytics shows three top platforms that are widely used for research which are Apache Spark's Machine Learning Library (Spark MLlib), TensorFlow, and Microsoft Azure Machine Learning Studio (Azure ML). Apache Spark ML library provides a platform which generally uses Machine learning algorithms for regression, classification, and clustering (Quddus, 2018).

Literature by Pérez-Chacón et al. (2016) showed k-means algorithm in Spark MLlib was used to observe the electric energy consumption behaviour in a big time series-based data. The results showed that the software managed to discover a day-based consumption pattern with low computing power for big data sizing up to 3 years (2011, 2012 and 2013) for two buildings of a public university. Another type of platform is TensorFlow, which is an open-source library, that is focuses more on deep learning and reinforcement learning technique (Ramsundar and Zadeh, 2018). Under Apache 2.0 open-source licensing, TensorFlow development was then initiated by Google Brain team. The software's name, TensorFlow, basically explains its framework whereby it implements a data flow graph, consists of "tensors" (data batch which will be processed) and "flows" (data motion in the system) (Abadi et al., 2016). On the desktop, it is able to utilise both CPUs and GPUs resources such as in the study by Cai (2019), where the research implemented Convolutional Neural Networks (CNN) in TensorFlow. The computational platform which consists of Intel Core i7 5820 K and Nvidia GeForce GTX Titan X are both utilised to provide a learning-based power and runtime modeling.

The machine learning approach at the enterprise level has also emerged suddenly due to the introduction of a scalable data framework for handling big data. These companies would usually go for Microsoft Azure Machine Learning Studio (AzureML) as it utilise cloud-based predictive analytics, thus requires less investment in hardware for conducting analysis (Mateev, 2019). Such example of a company is British Petroleum (BP) whereby Azure Artificial Intelligent is utilised to improve their safety performance and work efficiency in terms of exploring potential new energy by generating useful model within lesser time (Microsoft Customer Stories, 2019). In addition to its functionality, AzureML supports Python and R as its external programming (AzureML Team, 2016). Moreover, Azure ML platform can provide machine learning service from the start until the generation of a predictive model and is able to continue with the next step, whether publishing or deploying the model to a website or other platforms (Qolomany et al., 2019).

3. Methodology

Based on the reviewed studies regarding the critical areas, this research conducts energy consumption prediction using a dataset previously collected from a commercial building located in Klang Valley Malaysia, from June 2018 until December 2018. The commercial building is equipped with IoT meters that are connected to the power inlet socket at two major tenants of the building. Each tenant is divided into two areas consisting of two IoT meters named tenant A1, A2, B1 and B2. Collected data per minute that mapped into Tenaga National Berhad (TNB) requirements was saved in an open-source web server. Collected data can be extracted manually from the online platform in the form of a CSV file.

The prediction method will employ 3 machine learning algorithms which are k-NN, SVM and ANN. Feature attributes for this prediction will use electrical power data consisting of power factor, voltage and current, in which the demand would be the targeted output. The prediction modeling will be conducted inside Microsoft Azure Machine Learning Studio (AzureML) utilising R programming language (Caret Package). Microsoft Azure has been chosen as a platform in this research based on the literature reviewed in the previous section. Before model training and testing, the raw data will initially be analysed and pre-processed to reduce the complexity of the model training and to manage any missing data. Finally, each model will be evaluated using validation metrics. Consequently, the energy consumption prediction framework will consist of four parts, which are:

Step 1 Normality testing of dataset
Step 2 Data pre-processing
Step 3 Model development (training)
Step 4 Model evaluation (test)

3.1. Step 1: normality testing of dataset

In this research data analysis, a normality testing of the dataset for each tenant was conducted to determine the dataset distribution. This process was orchestrated by identifying the skewness and kurtosis of the dataset. Normality testing is significant for model development as it usually assumes that the dataset was normally distributed. Based on the background research by Mishra et al. (2019), it was found that normality testing is ignorable if the sample size exceeds 100. However, understanding of the dataset distribution could provide a consequential analysis for the result of the prediction. On the ground of statistical analysis, skewness is defined as the measure of irregular probability distribution around the mean value whereas kurtosis is a quantification of the distribution peakness. The formula for skewness and kurtosis is as shown in equations (1) and (2), respectively

$$\text{Skewness,} \quad S = \sum_{i=1}^{N} (x_i - \mu)^3 \tag{1}$$

$$\text{Kurtosis,} \quad K = \sum_{i=1}^{N} (x_i - \mu)^4 \tag{2}$$

where N is the total number of hours, x_i is power consumption, and i is hour of the day.

For this analysis, AzureML Summarize Data module was utilised to present the normality testing numerically and graphically. This module in AzureML is used to create a set of standard statistical measures that describe each column in the input table.

3.2. Step 2: data pre-processing

The preliminary process in machine learning includes data pre-processing for the preparation of data, and it usually consumes much time and computational power. This process is required as the dataset usually consists of missing value and an inconsistent scale of value between features (Fontama et al., 2014). In this research, the data was pre-processed using mechanics of imputation of missing data and standardisation in which the former utilised Azure ML proprietary Clean Missing Data module while the latter was done using Caret R Package. For Clean Missing Data module in AzureML, it was used to remove, replace, or infer missing values. This module supports multiple types of operations for cleaning missing values including replacing missing values with a placeholder, mean, or other value; completely removing rows and columns that have missing values; or inferring values based on statistical methods.

Before proceeding with cleaning missing data, their mechanism was recognised by identifying the correlation between the tendency of the data to be missing and the data value itself. After the mechanism was identified, the methodology was continued with imputation using Probabilistic Principal Components Analysis (PPCA) method under Azure ML Clean Missing Data module. PPCA is a Maximum Likelihood Estimate-based method which applies the Expectation-Maximization (EM) algorithm to determine the value of lost data. PPCA was derived initially from Principal Component Analysis (PCA) method, which was used for dimensionality reduction or also known as compression of data (Hegde et al., 2019). During the process, the PCA method managed to minimize the restructuring error on variance by decreasing the Euclidean distance between the initial data and the conjectured data points. This particular advantage of PCA was utilised for the imputation of lost data by initially approximating the dispersion of the compressed data based on the known data. Then, the missing data was restructured based on the compressed details as predicted data points.

The imputation method was also evaluated to determine its performance. The resultant cleaned data was then further pre-processed using standardisation. Standardisation or also known as Z-score normalisation is a transformation to change the observed data to have characteristics of standard normal distribution in which the mean is 0, and the standard deviation is 1. This transformed the data to be equally distributed above and below the mean value by using the formula in equation (3).

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{3}$$

where μ is mean and σ is standard deviation.

The standardisation process under Caret Package consists of 2 steps which are centring and scaling. The centring transformation computes the mean for a feature and subtracts it from each data point of the feature. On the other hand, the scale transformation computes the standard deviation for a feature and divides the output from centring transformation with the standard deviation.

3.3. Step 3: model development (training)

This research used a supervised machine learning methodology to predict energy consumption. After data was prepared, it was then inputted into the learning algorithm. Different feature combinations were fed into the algorithm to generate a candidate for the predictive model. Before using the data to create and train the model, data partitioning was done to separate the data into two groups – a training group and a testing group.

The predictive modeling for this research used a classification method to predict discrete variables instead of regressive prediction. As Azure ML does not have k-Nearest Neighbour and Artificial Neural Network for classification, the modeling function in Caret R package was utilised for all prediction to ensure uniform execution. Three types of machine learning algorithm were used for this research which were Artificial Neural Network (ANN-MLP), k-Nearest Neighbour (k-NN), and Support Vector Machine (SVM-RBF). Fig. 1 below shows the process after the data preparation until the generation of the predictive model.

3.3.1. k-Nearest Neigbour (k-NN)

The first predictive model to forecast energy consumption used the k-Nearest Neighbour (k-NN) method. This machine learning method is frequently used due to its simple criteria and its forecasting capability on intricate non-linear pattern (Valgaev et al., 2016). It manages to provide prediction by determining similar instances between data points in feature space (González-Briones et al., 2019). For this research, the method was used to predict maximum demand by using voltage, current and power factor as the features as the resultant multiplication of the values will output the electrical power usage (kW). Based on the Euclidean distance function in equation (4), k-NN method was trained repeatedly up to the maximum tuning parameter (k-value) which equals to 49. The resultant model from the training with the lowest RMSE value was chosen for prediction.

$$\text{Euclidian distance function} = \sqrt{\sum_{i=1}^{k} (xi - yi)2} \tag{4}$$

3.3.2. Support Vector Machine (SVM)

In this research, the Support Vector Machine (SVM) was used with Radial Basis Function (RBF) as its kernel function. This methodology is usually known as a maximum margin classifier and is utilised to tackle problems regarding classification and regression for a large dataset (Ben-Hur et al., 2008). There are several kernel selections available for SVM method. In this study, Radial Basis Function (RBF) as shown in equation (5), was chosen due to the broad and non-linear characteristics of the dataset.

$$K_{RBF}(x, x') = \exp\left[-\gamma x - x'^{\,2}\right] \tag{5}$$

where γ is a gamma parameter to determine the spread distribution of the kernel and $x - x'$ is the Euclidean distance between the set of points.

Equation (5) has a corresponding definition as represented in equation (6) using sigma parameter (Liu et al., 2010).

$$K_{RBF}(x, x') = \exp\left[-\frac{x - x'^{\,2}}{2\sigma^2}\right] \tag{6}$$

There are 2 tuning parameters for SVM-RBF which are kernel parameter sigma (σ) and cost parameter (C), that were adjusted for repeated training. The sigma value plays an important role in getting a good fit model to the data. Cost parameter is the penalty limit if the data point is misclassified or oversteps maximum margin.

3.3.3. Artificial Nneural Network (ANN)

The third methodology in this research for energy consumption prediction was Artificial Neural Network (ANN). The advantages of using ANN such as its capability to learn complex behaviour, makes it widely used for predictions and pattern recognition (Karunathilake and Nagahamulla, 2017). ANN model structure consists of a formation of interconnected neurons that have three main layers; input layer, hidden layer, and output layer. By comparing the initial output with the desired output, adjustment of the synaptic weight of each link that connects between the neurons was made until the difference is minimal (minimising Sum Squared Error (SSE)); this would provide regularization for the model

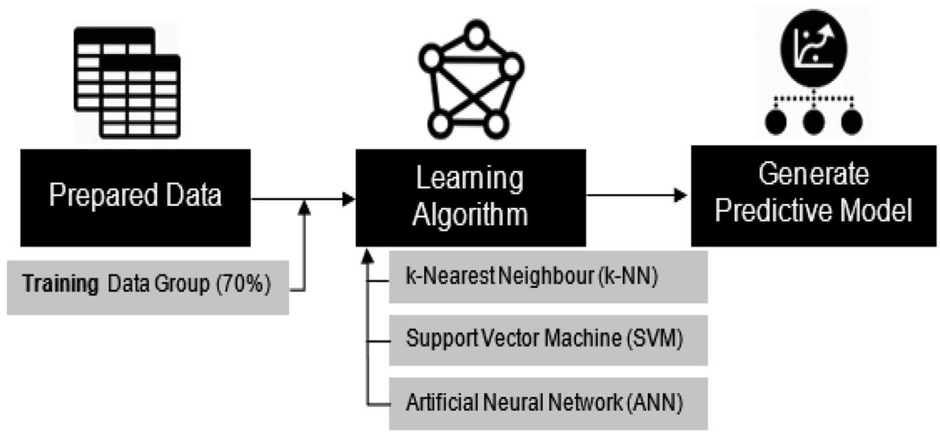


Fig. 1. Process of generating predictive model after data preparation.

(Liu et al., 2019). The weight is the representation of the priority or importance of the neuron input. For this research, a Multilayer Perceptron Model (MLP) type of ANN structure with error back propagation learning algorithm was used for its network solution structure. In the hidden layer, a suitable non-linear transfer function was used to compute the information accepted by the input layer. The ANN model is as shown in equation (7).

$$y_t = \alpha_0 + \sum_{j=1}^{n} \alpha_j f\left(\sum_{i=1}^{m} \beta_{ij} y_{t-i} + \beta_{0j}\right) + \varepsilon_t \tag{7}$$

where m is the number of input nodes, n is the number of hidden nodes, f is the Sigmoid Transfer function, $\{\alpha_j,\ j = 0, 1, \dots n\}$ is the vector of weights from the hidden layer to the output layer and $\{\beta_{ij},\ i = 0, 1, \dots m;\ j = 0, 1, \dots n\}$ is the weight from the input to the hidden nodes.

For this research, the hyperparameter tuned was the number of neurons per layer. This number of neurons denotes the width of the network and its latent space (Weissbart et al., 2019). Another penalizing parameter that was tuned and applied was weight decay. This parameter is a penalizing method to constrain the complexity of the model and to limit the growth of the model's weight parameter (Gnecco and Sanguineti, 2009).

3.4. Step 4: model evaluation (test)

Before inputting the data to the machine learning algorithm, the data was partitioned into two groups whereby 70% of the dataset was used for training and the other 30% was partitioned as testing data groups. The training groups of data were used to train each machine learning algorithm and generate a predictive model that could output value that matches with the recorded maximum demand data while the rest of the data was held back to be used to test the trained predictive model. The process is as illustrated in Fig. 2.

With AzureML, data partitioning for training and testing would not be a hassle and biased as it has built-in support for data division. The partitioning process was straightforward in which selection was made randomly. This process prevented overfitting, which could cause either underestimation or overestimation of the maximum demand value.

During model training, several models were created with different tuning parameters for each method, in which k-value was adjusted for k-NN tuning; sigma and C parameter were modified for SVM-RBF tuning; and weight decay and hidden unit size were adjusted for ANN-MLP tuning. After the repeated tuning finished up to its respective maximum parameters, each model was evaluated based on Root Mean Square Error (RMSE), R-Squared (R^2) and Mean Average Error (MAE). The formula is as shown in equations (8)–(10), respectively, given that A_t is the actual recorded values of maximum demand data, and F_t is the predicted values. Although 3 evaluations were made, only RMSE result was acknowledged as the best model for each method.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (A_t - F_t)^2}{n}} \tag{8}$$

$$R^2 = \frac{\sum_i (A_t - F_t)^2}{\sum_i (A_t - \bar{A})^2}\ ;\ \bar{A} = \frac{1}{n} \sum_{t=1}^{n} A_t \tag{9}$$

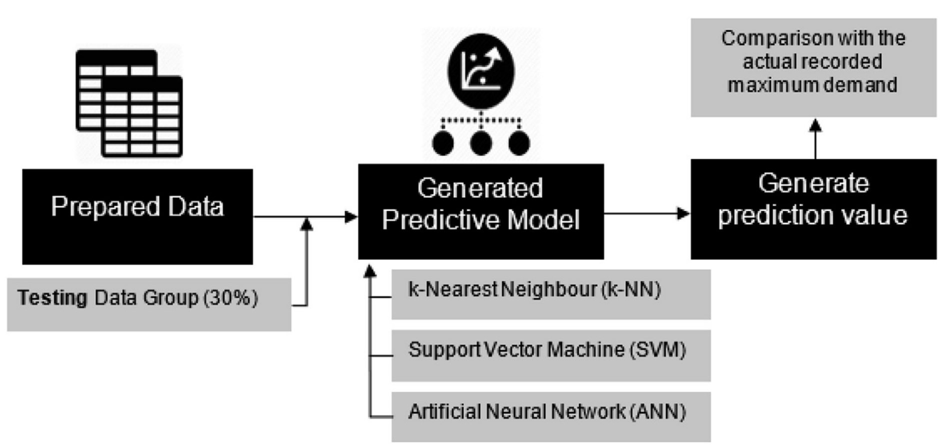


Fig. 2. Testing of the trained predictive model.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \quad (10)$$

After the predictive model using each machine learning algorithm was developed and prediction demand data was generated, they were then evaluated to determine their performance and accuracy. Three methods of evaluation were used which were Root Mean Square Error (RMSE), Normalised RMSE and Mean Absolute Percentage Error (MAPE). The formula for RMSE is as shown in equation (8) and MAPE in equation (11). Comparison of performance of the methods to different tenants was made by using a normalised RMSE or also known as Coefficient of Variation RMSE (CV RMSE). This metric removes the scale dependent of RMSE (Botchkarev, 2019).

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2 \quad (11)$$

$$NRMSE = \frac{\sqrt{\frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}}}{Mean} \quad (12)$$

4. Results and discussion

The results of the experimentation were discussed in sections based on the steps of the predictions' framework. The pre-processing method and imputation of missing data using modules provided by both Microsoft Azure Machine Learning Studio and Caret Package were discussed in terms of procedure, effects, and importance. The findings regarding energy consumption prediction were reviewed for each tenant and performance comparison was provided for the prediction result of Artificial Neural Network (ANN), Support Vector Machine (SVM) and k-Nearest Neighbour (k-NN). Holistically, utilisation of Azure ML Studio as a forecasting medium was inspected to identify its reliability and compatibility to predict energy consumption.

4.1. Normality testing of dataset

For the normality test of the energy demand data, the skewness and kurtosis values were calculated using the aggregate data for each tenant, starting from June 2018 until December 2018. This assessment was intended to analyse whether the shape of data affects the performance of the developed predictive model. The generated data was compiled in Table 1 and Table 2, for skewness and kurtosis, respectively. Additionally, Fig. 3 shows the form of the dataset for a graphical assessment of normality. The skewness and kurtosis values were computed for all attributes, including the demand.

Based on Table 1 and Fig. 3, Tenant A1, A2, and B1 dataset was approximately symmetry and skewed with bimodal shape density. Nevertheless, for Tenant A1, the density was skewed left, as the long tail pointed to the left whereas for Tenant B1, the density was skewed right. Tenant A2 shows asymmetry normal whereby the tails between each end were approximately balanced. Different from Tenant B2, the distribution was highly skewed as the skewness was more than 1 at 1.267578. The density plot of Tenant B2 distribution shows that the density was also bimodal with right-skewed.

Table 1
Skewness for each tenant using aggregate data.

Tenants	Skewness			
	Power Factor	Current	Voltage	Demand
A1	−0.457296	−0.17541	−0.116392	−0.279034
A2	−1.564851	−0.105746	−0.054989	−0.182159
B1	−1.877798	−0.666165	−0.310973	0.282481
B2	1.735751	0.202681	−0.714503	1.267578

Table 2
Kurtosis level for each tenant using aggregate data.

Tenants	Kurtosis			
	Power Factor	Current	Voltage	Demand
A1	6.026131	−1.648018	−0.45877	−1.053977
A2	2.839891	−1.763198	−0.768862	−1.584641
B1	2.062267	−1.011822	0.469322	−0.126043
B2	3.472321	−1.403968	4.623413	1.824909

In terms of kurtosis, from Table 2, Tenant A1, A2 and B1 had excess kurtosis less than 0. This means that the distributions were platykurtic. This characteristic was also observed based on Fig. 3 (A1, A2, B1) whereby the probability density plot has a broader tail, and the peak centre was wider. Contrary to Tenant B2, the excess kurtosis was higher than 0 at 1.824909. This indicates that Tenant B2 distribution was leptokurtic and had a higher variance. From this normality testing, Tenant A1 and A2 had an approximately normal distribution. However, Tenant A1 has a mean value less than the median. Tenant B1 had also an approximately normal distribution but had slightly higher skewness and kurtosis compared to Tenant in department A. Tenant B2 dataset was highly skewed and had a higher variance in comparison with the other tenants.

4.2. Imputation of missing data

The study of missing data was conducted inside Azure ML studio whereby Summarize Data module was utilised to determine the amount of missing data and to reveal the rows which have missing data. The diagnosis of missing data was also conducted via observation on the value of other attributes in the same row. Table 3 shows a summary of the analysis.

Briefly, the total number of data for A1 is 4666, A2 is 4666, B1 is 4648 and B2 is 4648. Based on Table 3, it was noticed that the dataset only had a missing value for demand, which was the targeted output. It was also observed that Tenant B2 had the highest number of missing data. The identification of the missing data mechanism was made by referring to the power formula and the method of data generation. The voltage and current values were captured by the voltage and current sensors. On the other hand, the values for power demand and power factor were generated bases on analysing the magnitude and waveform for both voltage and current. The electrical power formula also shows that the power value was the multiplication result of the three attributes. This indicates that demand and power factor variables are dependent on voltage and current value. Observation in Table 3 shows that although power factor, current and voltage attributes had value, the respective demand output was missing. This deduces that the missing value was Missing Completely at Random (MCAR), in which the missingness of demand data does not depend on the observed attributes. The missing value for this data was then imputed using the PPCA method utilising the Clean Missing Data module in Azure ML Studio. Although data MCAR was negligible, the dataset was not ignored as it would discard valuable information. Understanding the mechanism of the missing data has provided proper guidance for the imputation process of missing data. This prevents the missing data from being imputed with the wrong value, which could generate more outliers. Continuing with missing data imputation, Probabilistic Principal Component Analysis was chosen as the configuration for the imputation method utilising AzureML Clean Missing Data module. Fulfilling the objective of this research, the cleaning method was evaluated based on Raw Bias (RB), Coverage Rate (CR) and R-Squared criteria. Table 4 shows the result of the evaluation for each tenant.

From the raw biased result, PPCA managed to produce imputation value with a low difference between the mean of the estimated value and

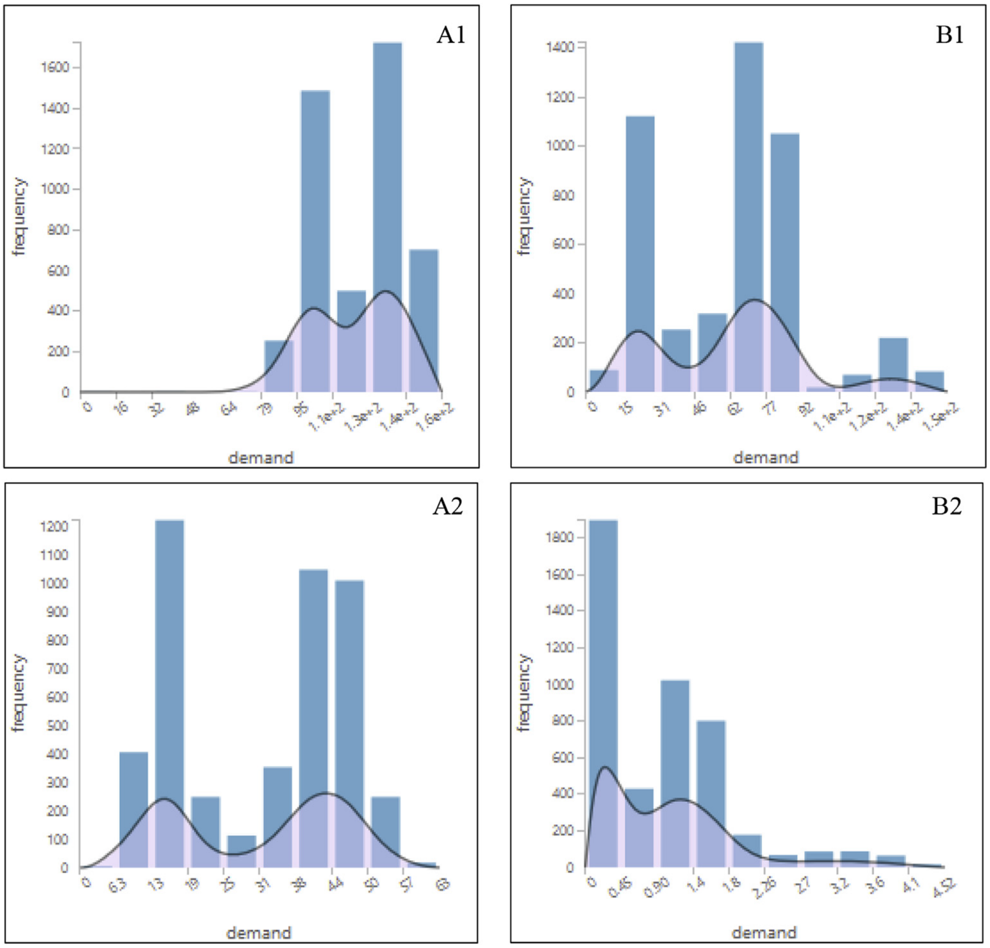


Fig. 3. Probability density for tenant A1, A2, B1 and B2.

Table 3
Number of missing data for all tenants.

Tenants	Number of missing data				Total number of missing data
	P. Factor	Current	Voltage	Demand	
A1	0	0	0	1	1
A2	0	0	0	3	3
B1	0	0	0	1	1
B2	0	0	0	171	171

Table 4
Evaluation of PPCA based imputation using AzureML Clean Missing Data Module.

Tenants	Raw Bias	Coverage Rate	R-Squared
A1	−0.02813237	0.9003215	0.8997098
A2	−0.006479131	0.9011366	0.9039039
B1	0.1121446	0.9014418	0.9051205
B2	−0.000560815	0.9014965	0.90587942

the mean of the actual value; this shows that the method was unbiased. Notably, the result for Tenant B2 shows a higher biased in comparison with the result for the other tenants. The coverage rate result shows that 90% of the imputed values fall within the confidence interval. Although it was less than 95%, the coverage rate did not fall below 90%, which in that case will be denoted as poor (Demirtas et al., 2008). The R-Squared result shows the estimated value has a good fit for the actual value.

4.3. Data pre-processing

For data pre-processing, it was observed from Fig. 4 that the original dataset contained different scale ranges between power factor with current, voltage and demand. The power factor has a value ranging from 0 (min) until 1 (max). However, the current, voltage and demand all have a higher range of scale. After the standardised transformation, the value managed to be scaled within the same range for all attributes and centralised to achieve a mean of 0 and a standard deviation of 1. This is the same for all tenants.

4.4. Performance evaluation and comparison

Subsequently, after model training and testing, the predictive model generated was compared in terms of performance between algorithms for each tenant. Initially, the result of the testing was observed by comparing the performance of the methods for individual tenants. The comparison table is as shown in Table 5.

From the performance evaluation in Table 5, the SVM method utilising Radial Basis Function kernel had the best performance for Tenant A1 and A2, in which the RMSE value was 4.7506789 and 3.5898263, respectively. Even though SVM had the best performance, the difference between RMSE value in comparison to k-NN method was minor. The MAPE result also indicates that SVM prediction had a lower absolute error percentage. For Tenant B1, k-NN was the best performing method in which the RMSE value was comparably smaller than the other method at 14.934312. However, its absolute error was higher than SVM absolute error. This event happened similarly to Tenant B2. The best performing algorithm was k-NN with 0.5439403 RMSE value, which was smaller

```
       pf              current            voltage           demand
 Min.   :0.8540   Min.   : 70.55   Min.   :226.7   Min.   : 33.65
 1st Qu.:0.9410   1st Qu.:100.92   1st Qu.:238.9   1st Qu.:104.19
 Median :0.9460   Median :132.22   Median :241.5   Median :129.95
 Mean   :0.9459   Mean   :122.36   Mean   :241.5   Mean   :123.36
 3rd Qu.:0.9510   3rd Qu.:141.83   3rd Qu.:244.2   3rd Qu.:140.35
 Max.   :0.9830   Max.   :160.21   Max.   :249.9   Max.   :158.97

Created from 4666 samples and 4 variables
Pre-processing:
  - centered (4)
  - scaled (4)
       pf                current              voltage               demand
 Min.   :-12.86809   Min.   :-2.4499   Min.   :-4.422776   Min.   :-4.7221
 1st Qu.: -0.68613   1st Qu.:-1.0137   1st Qu.:-0.773300   1st Qu.:-1.0092
 Median :  0.01398   Median : 0.4661   Median :-0.009282   Median : 0.3467
 Mean   :  0.00000   Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.000
 3rd Qu.:  0.71410   3rd Qu.: 0.9206   3rd Qu.: 0.822143   3rd Qu.: 0.8946
 Max.   :  5.19482   Max.   : 1.7899   Max.   : 2.516005   Max.   : 1.8747
```

Fig. 4. Summary of transform dataset for Tenant A1.

Table 5
Performance evaluation of test prediction for all tenants using trained SVM, k-NN and ANN model.

Tenant	Method	RMSE	NRMSE (%)	MAPE (%)
A1	k-NN	5.0025748	4.06	3.02
	SVM	4.7506789	3.85	2.76
	ANN	8.874015	7.19	5.02
A2	k-NN	3.6548885	11.46	9.98
	SVM	3.5898263	11.25	9.38
	ANN	4.540988	14.23	14.16
B1	k-NN	14.934312	23.87	15.43
	SVM	16.0690844	25.69	12.09
	ANN	20.63566	32.99	28.00
B2	k-NN	0.5439403	55.87	48.75
	SVM	0.5558279	57.09	43.97
	ANN	0.547152	56.20	60.62

compared to the RMSE result for SVM and ANN. However, the MAPE result for SVM was lower than the result for k-NN. As the forecast stated was in terms of the expectation of value, the square error method was a much better evaluation method (Tilmann, 2010). Thus, the k-NN method was denoted as the best performing algorithm for Tenant B1 and B2. Referring to the normalised RMSE result, it can be observed how each of the prediction methods performed differently under different datasets. The performance of every method deteriorated from Tenant A1 until Tenant A2, in which Tenant A2 had the worst performance for every method.

Under TNB tariff category, the tenants at both departmental lots A and B were categorised as Medium Voltage General Commercial (Tariff C1). This means that the monthly electricity charges were calculated by acquiring the maximum demand of the month and the kWh (Tenaga Nasional Berhad, 2006). For Tariff C1, the off and on-peak period was not applied to the billing process. Therefore, to predict energy billing thoroughly, the maximum energy demand and the kWh need to be determined. For maximum demand, forecasting an expectation of value can be done to predict the maximum demand of the month. However, for the energy consumption (kWh) prediction, the hourly predicted demand needs to be added up. As the average value shows the characteristics of the whole dataset, the average forecasted consumption was quantified and compared with the actual average consumption. The percentage of difference error between the forecasted and actual value would determine the best forecasting method. The comparison between actual and forecasted average consumption was tabulated and visualized in the form of a line graph, as shown in Fig. 5 until Fig. 8 in the Appendices.

Fig. 5 shows that the ANN forecasted average consumption had a stagnant peakness in which the forecasted average was almost the same

Table 6
Mean Absolute Percentage Error (MAPE) of forecasted method for all tenants.

Tenant	k-NN	SVM	ANN
A1	0.40	0.241318507	1.108522675
A2	0.942855477	0.666018364	1.841600488
B1	8.596963137	8.174497001	18.03425668
B2	24.61323638	17.78423714	29.0736946

for every month. The SVM forecasted average consumption, on the other hand, had better forecasted performance in which the line graph was much closer to the actual average consumption line graph. The k-NN forecast had a slightly larger error in comparison to the SVM forecasted result. Comparison between predicted and actual consumption for Tenant 2 in Fig. 6 shows that both predictions of average consumption made by k-NN and SVM method had better fit to the actual values than ANN method. In November, it can be observed that all methods forecasted a lower consumption for Tenant A2 than the actual consumption. In Fig. 7, the forecasted average consumption by k-NN and SVM was better than ANN. It was also perceived that all methods did not take into consideration the individual high consumption in the month of June. The performance of all methods in Tenant B1 was approximately similar to the performance of predicted consumption in Tenant B2. Identification of the best method to determine the average consumption on a monthly basis was made by calculating the MAPE value for every month. This calculated value was tabulated in Table 6, whereas Fig. 9 in the Appendices shows a bar graph that compares the percentage of different errors.

From Table 6 and Fig. 9, a conclusion can be made in which SVM forecasting is the best method to forecast monthly average consumption. The difference between the percentage of error by SVM method with that of k-NN was not significant. From the visual, the average consumption predicted by ANN method had the highest percentage in every tenant. This denotes that ANN method is inferior in comparison to both SVM and k-NN methods. Comparison of the model training time, as shown in Table 7, was conducted to determine the method with the lowest running time. From the tabulation, the SVM method took the longest time to train while k-NN method was the fastest to finish training. ANN method was faster compared to SVM but still took several hours to complete the training.

From the performance evaluation and comparison, SVM method was the best method to predict the individual peak energy demand for department lot A, while k-NN was the preferable choice for department lot B. In terms of average energy consumption, SVM method has proven to be the best method to predict the monthly mean value for energy consumption. However, high training time was required for SVM model training to achieve this high performance. This result further supports the "No Free Lunch" theorem by Wolpert (1996), which was discussed by

Table 7
Comparison of model training running time.

Tenant	k-NN	SVM	ANN
A1	37.916s	18 h 38 m 55.324s	4 h 39 m 30.035s
A2	28.978s	17 h 23 m 32.637s	5 h 14 m 22.311s
B1	34.350s	13 h 20 m 3.722s	5 h 13 m 9.418s
B2	34.5s	12 h 43 m 48.147s	6 h 34 m 48.972s

Stenudd (2010). No Free Lunch stated that many scenarios would determine whether a machine learning method would perform much better than the other. In this research, the scenario would be the dataset distribution and the targeted output (max demand or average consumption). Due to the small difference in performance result between k-NN and SVM methods, it can be concluded that both of these methods were excellent prediction methods. The choosing element would be whether the users want a slightly better accurate result or a faster training method.

4.5. Effect of skewness and kurtosis level on non-parametric classifier

Observing the performance evaluation of each test method in Table 5 shows that Tenant A1, A2, and B1 result was normal as the RMSE value was not unreasonably small and the absolute error between the forecasted demand to the actual demand was low. On the contrary, Tenant B2 prediction model for each method was considerably poor even by referring the normalised RMSE. From data analysis, it was determined that Tenant B2 dataset was highly skewed, and its kurtosis was leptokurtic, which means presence of high variance in the dataset. Correlating both observations, it can determine that the skewness and kurtosis of a dataset have an effect, especially towards a non-parametric classifier. A non-parametric classifier is an algorithm which falls under probabilistic density supervised classification (Kumar and Sahoo, 2012). This classification model was used if the density function was unknown and does not have a fixed size of parameters. Other than that, the model does not have an initial assumption of the probability density of the dataset (Sampat et al., 2005).

During the training and learning process of these algorithms, the parameters increase with the training set. In this research, the model that was used, which are k-Nearest Neighbour, Support Vector Machine with Radial Basis Function kernel, and Artificial Neural Network with Multilayer Perceptron model, is a non-parametric classifier. SVM was initially considered as parametric, but due to the introduction of RBF kernel, the model becomes nonparametric as RBF kernel matrix computes the distance between two pairs of data points (Vasile and Camps, 2013). Skewed and positive/negative kurtosis signifies that the dataset was not normally distributed. Several works of literature described how these properties influence the performance of a non-parametric classifier. Kurtosis level for a dataset has a significant effect on the performance of a nonparametric classifier. This was as indicated by Larasati et al. (2018) in which a leptokurtic type of distribution have a considerably higher chances to be misclassified as it have a higher dispersion of data from the mean value. Another study from the same researcher established the relationship between a skewed dataset and the accuracy of a non-parametric classifier (Larasati et al., 2019). The study concluded that a skewed data does affect the performance of the predictive model, especially the research focussed model, which was artificial neural network. A highly skewed data indicates that the data was unbalanced in which certain classes or output exist much frequently than the other output. Since non-parametric classifier model train by learning the pattern arrangement of the partitioned dataset, data that occurred less frequently was ignored. Therefore, the prediction model performed poorly for a highly skewed dataset. This explanation was further supported by Siddiqui and Ali (2016) in which the study was conducted to determine the performance of nonparametric on a skewed data. The study dictates that a skewed data would require a notable high learning effort for training, consideration for outliers and complicated computation. This requirement thus limits the performance of any non-parametric classifier on skewed data. Non-normal distributed data have shown to have great influence on the prediction model. Therefore, countermeasure needs to be taken to nullify this impact. Based on McCarthy et al. (2019), three techniques were suggested which are using transformation pre-processing for variance stabilizing; binning transformation which fraction the attributes into several group with a proper control of the weight at each range; and combination of both transformation method.

4.6. Analysis of Microsoft AzureML studio environment

This research has utilised Microsoft Azure Machine Learning Studio, which is a web service solution for the development of prediction model. Starting from data analysis until performance evaluation, AzureML has been successfully employed for the implementation of energy demand forecasting. In this section, the employment of AzureML was analysed in terms of 3 criteria which are: a) Distinguishing Features, II) System Overview, and III) Experimentation Overview. Thoroughly, this analysis would provide an insight on how reliable and capable AzureML studio for developing a prediction model.

4.6.1. Distinguishing features

One of the distinguishing features of AzureML was its ability to manoeuvre through a visualization workflow. The workflow that was conducted inside the environment was manipulated through a graphical drag and drop procedure. Other than that, parsing data for experiment was simply done by joining of modules. Additionally, the platform also supports script packages and algorithms written in external programming language, particularly R programming. In this research, R programming was heavily utilised to conduct prediction modeling such as data preprocessing and model training. The versioning feature of AzureML had helped to reduce the time for experimentation, in which each version of individual experiment information, for instance, parameter settings, was cached.

4.6.2. System Overview

Prediction model development inside AzureML has utilised four primary services which are the studio user interface, Experimentation Service (ES), Job Execution Service (JES) and Singe Node Runtime (SNR). The studio interface of AzureML provides important item for model development which include the availability of a significant number of modules and the ability to import and save user assets. In this research, the Summarize Data module, Clean Missing Data module and Execute R Script module were used. Under primary user interface is the ES which is the backend of AzureML. This service is the system that command the interaction between components that were present in the main interface, which include managing the events in the main interface such as data transformation and program execution. Other than that, the ES plays an important role as the repository for all the imported and saved assets, which in this research was the.CSV file containing energy consumption data.

JES act as the scheduler that performs the module execution. Responsibility of a scheduler was to track the task execution for each experiment. Inside the workspace of AzureML, modules used for experiment can be placed in parallel with other modules. When this module meets the requirements, such as selection of column to be manipulated, the ES runs the modules while JES scheduled each modules timing to be executed. In this research, cleaning of dataset need to be done before partition the dataset into testing and training. JES ensure that the cleaning process were executed first before the partitioning. The service that run the respective experiment is the Singe Node Runtime (SNR). SNR will receive the task parse by JES and execute the task. All this service works dependent to each other to ensure that the experiment works as desired.

4.6.3. Experimentation Overview

The experimentation of prediction model development was able to be done using the provided service by AzureML. This includes data import, data analysis, and data manipulation. However, under modeling, parameter tuning, and evaluation, the web service was unable to provide the required service. The algorithm for this research was k-Nearest Neighbour, Support Vector Machine with Radial Basis Function, and Artificial Neural Network with Multilayer perceptron model. AzureML does not provide this particular service, in which SVM model available was only made to provide binary and multiclass classification, not discrete value classification. Moreover, neural network provided was only available for regression and binary classification. Compensating for a low number of available models, AzureML provides Execute R Script module, which solves the problems as R package for modeling was able to be imported into the workspace. Albeit, using external R programming for prediction modeling requires the use of parameter tuning and performance evaluation by executing R codes. Thus, the parameter tuning, model scoring and evaluation module inside AzureML was not utilised. Collectively, AzureML platform provides a reliable and comprehensive environment for prediction model development, provided that the required machine learning algorithm for the development was available. Furthermore, AzureML provides conversion of an experiment into a web service, in which real-time data collection can be input into the workspace. Initially, the utilisation of this service was proposed. But due to Coronavirus 2019 (Covid-19) pandemic, the proposal was withheld, and prediction model development was done by using dataset that have been previously collected instead.

4.7. Comparison with previous study

Analysing the result of this study have shown significant discoveries in terms of the performance comparison between k-NN, SVM and ANN in forecasting maximum demand and average monthly demand. Furthermore, the process of prediction model development itself has displayed the capability of AzureML platform in terms of data pre-processing and computational power itself. From these discoveries, a comparative analysis was made with the previous study relating to machine learning.

One of the assumptions was made based on the research conducted by Fu et al. (2015), in which the next day electricity load was predicted using the Support Vector Machine algorithm. The input data used was weather predictions and hourly electricity loads from two days earlier. This dataset was separate into training and testing based on the cooling season of the area. Specification on the algorithm used was SVM with Radial Basis Function kernel, which was similar to the kernel used for the SVM model in this research. However, Fu et al. (2015) also tuned the epsilon parameter, ε, other than optimising the gamma, γ, and cost, C, parameter. The case study was also conducted with three different algorithms which are ARIMAX, Decision Tree and Artificial Neural Network, in which prediction of electricity load was made on four different systems, particularly air conditioning, lighting, power and other. The result of the research shows that the SVM method provided a lower error rate prediction for lighting and power, in which the dataset was determine as normally distributed due to a stable electricity load. However, for air conditioning and others, the SVM method has a large error as both the system electricity usage was more complex and stochastic as it was dependent on the occupants' behaviour. The result has a similar performance result when utilising SVM RBF method in this research. For Tenant A1 and Tenant A2, which have an approximately normal distribution, the best performing model was SVM. The SVM method also have better performance compared to ANN.

ANN and SVM methods were significant in this study, as both methods are more complex compared to k-NN. In the previous research regarding the model accuracy analyses for building energy consumption prediction studied by Liu et al. (2019), the SVM method has better accuracy and higher complexity compared to ANN. But when kernel and model were applied, ANN with Multilayer perceptron was more complex than SVM Radial Basis Function. This was as concluded in the short-term load forecasting by El Khantach et al. (2019). In both of this research, the SVM method utilising RBF kernel was better in predicting energy demand.

With remarks on the performance analysis tabulated in Table 5, the RMSE difference between k-NN and SVM was small. Compared to the previous study done by González-Briones et al. (2019), the electricity consumption forecasting comparison between SVM and k-NN was considerably small. The research was conducted by learning the energy consumption in a different time granularity from daily, to yearly. The study also conducted prediction using Random Forest, Decision Tree, and Linear regression, in which Linear regression have similar accuracy with SVM at 0.857, while k-NN prediction accuracy was at 0.854, out of 1. Assessment of contrast between present and past research shows similarity in prediction result; nonetheless, this similarity was basically due to the fitting of the dataset to the prediction model assumption.

AzureML utilisation in this research was compared to Mateev (2019), whereby the AzureML platform was used to manage smart energy system and predictive analytics. Regarding predictive model development, the study utilised completely the solution provided by AzureML such as Cloud getaway, Data Storage and Prediction Layer. On account of the unavailable required algorithm in AzureML, the present study utilised the R programming inside AzureML instead for prediction model training, testing and evaluation. As AzureML perform model development in the server, the performance of the modeling does not dependent on the hardware system.

5. Conclusion

The course of this research has focused on developing an energy consumption predictive model for two commercial departments that have adopted the smart building ecosystem. The energy demand data collected from June 2018 until December 2018 has been analysed and pre-processed for training and testing of the predictive models. Utilising the Microsoft Azure Machine Learning studio (AzureML), statistical analysis of the data collected was made to determine the normality of the dataset. From this analysis, skewness and kurtosis values were acquired and established that all collected data was different in distribution characteristics. Continuing with the process of predictive model development, the data collected was pre-processed through the imputation of missing data using PPCA method and standardisation transformation. The pre-processing was successfully executed inside AzureML environment, in which the resultant processed data had a mean valued at 0 and a standard deviation of 1. An experimentation process was also made to determine the capability of the missing data imputation method. A sampled data with missing value was generated to evaluate the PPCA method, in which it produced a promising result with low raw bias and coverage rate of more than 90% of the actual value.

Focusing on the objective of this research, three supervised machine learning prediction methods namely k-Nearest Neighbour, Support Vector Machine with Radial Basis Function kernel, and Artificial Neural Network with Multilayer Perceptron model, were chosen as the algorithm for the predictive model. These methods were successfully compared in terms of their resultant structure and prediction performance. The consequence of the model training and testing shows that each method performed differently for every tenant. SVM method shows the most promising result, whereby it managed to be the best method for 2 tenants which were Tenant A1 and Tenant A2, with RMSE valued at 4.7506789 and 3.5898263, respectively. Furthermore, SVM result also shows a lower mean absolute error for Tenant B1 and Tenant B2 at 12.09 and 43.97, respectively, although k-NN had lower RMSE result for these two tenants. SVM predicted demand also had better accuracy when average consumption was calculated from the demand, in which it achieved a lower MAPE than the rest of the methods for all tenants. All in all, this SVM auspicious result comes with a price, whereby the model developed with the algorithm took 13–18 h to train. In contrast with k-

NN method that performed slightly worse than SVM, it only took maximum 40 s to train. For this research, Azure ML was employed for all processes and R programming was used extensively for data standardising, model training and testing and finally performance evaluation.

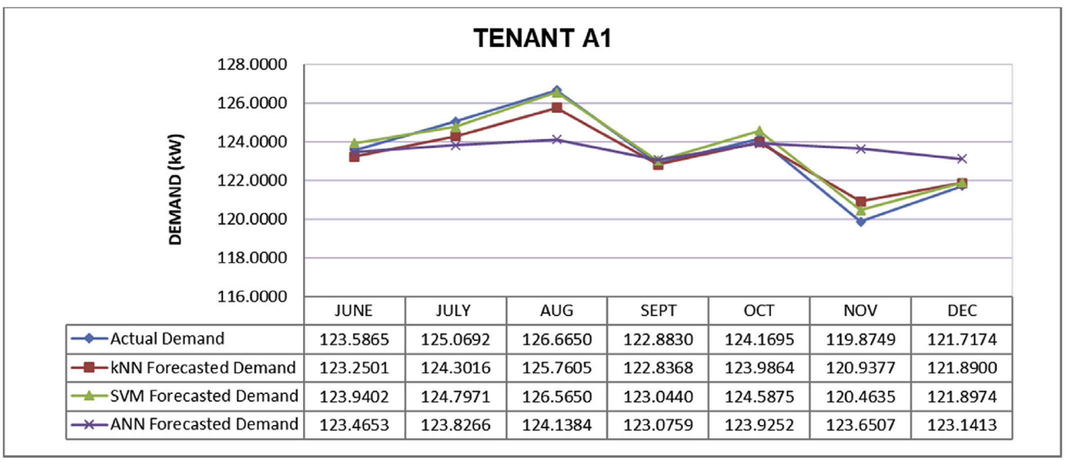
Cloud-based predictive model development has its advantage as it does not depend on the performance of the hardware it is running on. Moreover, it could manage to prevent from failing due to sudden hardware shutdown. However, AzureML particularly does not have the right algorithm needed for this research; as a result, R programming was used for the model training and testing and performance evaluation. Holistically, this research was triumphantly conducted and the objectives were achieved.

There are a few recommendations for future study that can be done to improve this study. Since the limitation of this study is the time taken to run the algorithm, hence, more powerful computer or platform should be used to run SVM algorithm. Secondly, more variables related and data should be collected as the input since there is limitation in this study on the data collection. Hybrid or ensemble methods can be proposed in the future study as well since it shows more accuracy than a single classifier. This study did not proposed hybrid classifier due to our focus is more on the platform than the method. Lastly, a comparison with another smart building could be added to differentiate the results obtained.

Declaration of competing interest

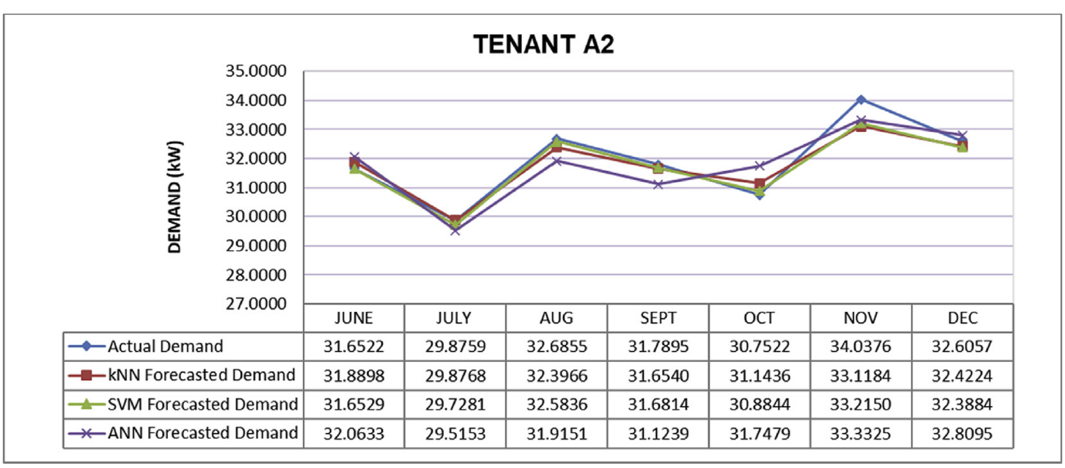
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendices.



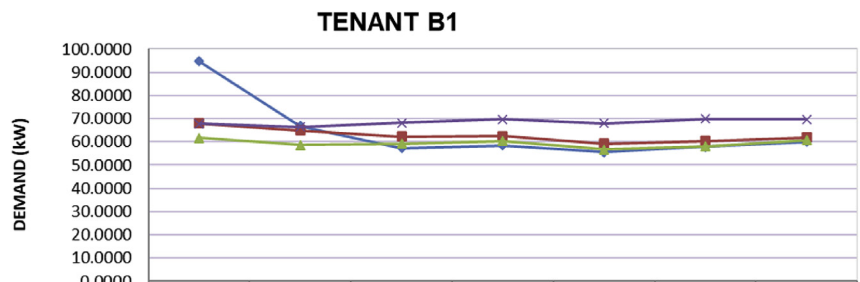
	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
Actual Demand	123.5865	125.0692	126.6650	122.8830	124.1695	119.8749	121.7174
kNN Forecasted Demand	123.2501	124.3016	125.7605	122.8368	123.9864	120.9377	121.8900
SVM Forecasted Demand	123.9402	124.7971	126.5650	123.0440	124.5875	120.4635	121.8974
ANN Forecasted Demand	123.4653	123.8266	124.1384	123.0759	123.9252	123.6507	123.1413

Fig. 5. Comparison between the actual and forecasted average demand for Tenant A1.



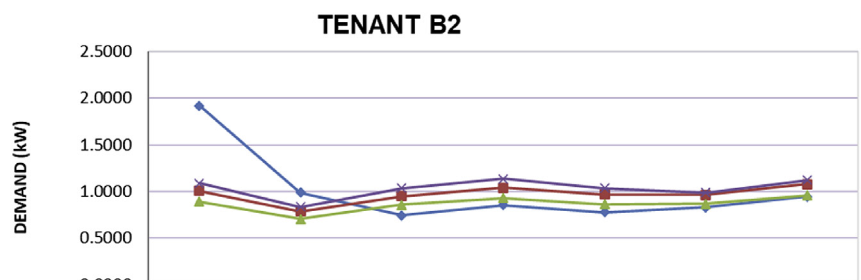
	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
Actual Demand	31.6522	29.8759	32.6855	31.7895	30.7522	34.0376	32.6057
kNN Forecasted Demand	31.8898	29.8768	32.3966	31.6540	31.1436	33.1184	32.4224
SVM Forecasted Demand	31.6529	29.7281	32.5836	31.6814	30.8844	33.2150	32.3884
ANN Forecasted Demand	32.0633	29.5153	31.9151	31.1239	31.7479	33.3325	32.8095

Fig. 6. Comparison between the actual and forecasted average demand for Tenant A2.



	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
Actual Demand	94.6897	66.8501	57.3975	58.3668	55.6314	57.9436	60.0092
kNN Forecasted Demand	67.9482	64.9945	62.3029	62.4319	59.2660	60.3343	61.8044
SVM Forecasted Demand	61.6907	58.5969	59.1627	60.3818	56.7882	58.1279	60.6696
ANN Forecasted Demand	67.8936	66.4288	68.1674	69.6951	68.0839	69.8800	69.7031

Fig. 7. Comparison between the actual and forecasted average demand for Tenant B1.



	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
Actual Demand	1.9174	0.9891	0.7421	0.8522	0.7782	0.8271	0.9437
kNN Forecasted Demand	1.0080	0.7848	0.9481	1.0378	0.9658	0.9630	1.0771
SVM Forecasted Demand	0.8928	0.7052	0.8570	0.9312	0.8643	0.8666	0.9602
ANN Forecasted Demand	1.0860	0.8328	1.0364	1.1356	1.0368	0.9862	1.1227

Fig. 8. Comparison between the actual and forecasted average demand for Tenant B2.

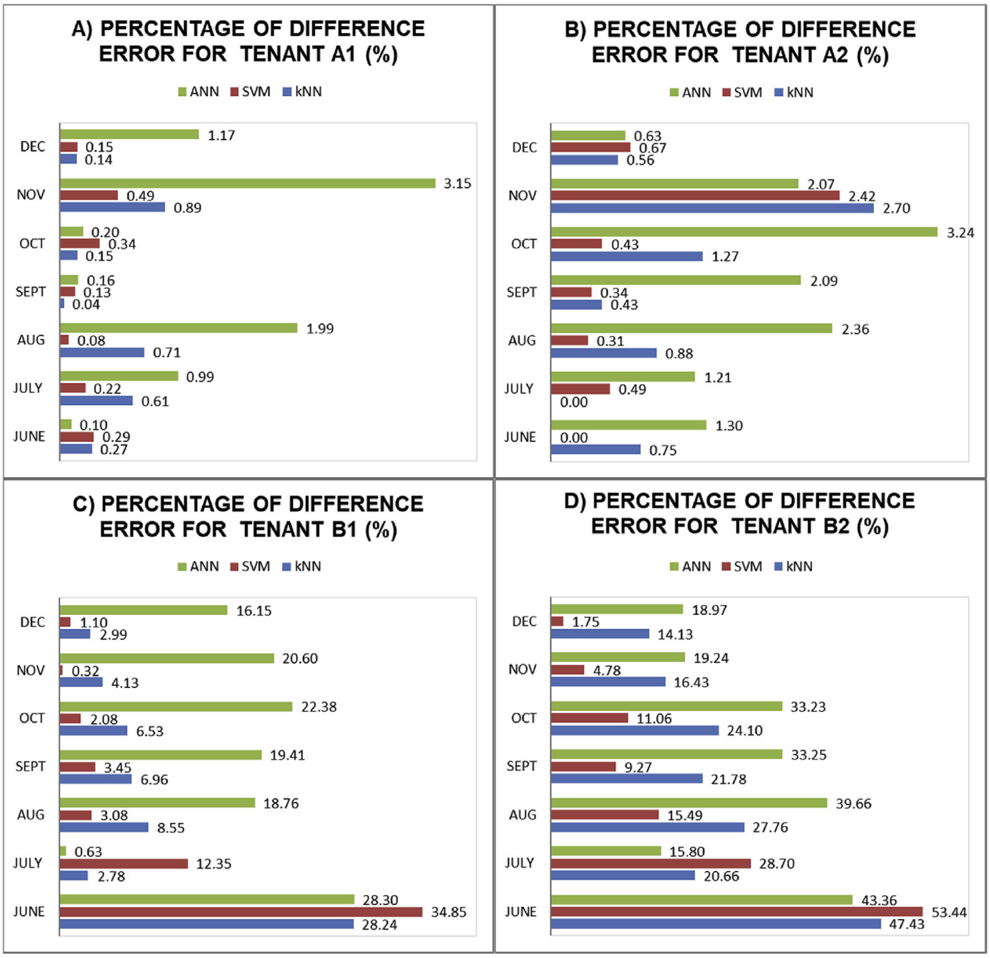


Fig. 9. Percentage of difference error between forecasted and actual average consumption.

References

Abadi, Martín, et al., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *ArXiv* abs/1603.04467.

Abdul Karim, S.A., Alwi, S., 2013. Electricity load forecasting in UTP using moving averages and exponential smoothing techniques. Appl. Math. Sci. 7, 4003–4014.

Ahmad, M.W., Mourshed, M., Mundow, D., Sisinni, M., Rezgui, Y., 2016. Building Energy Metering and Environmental Monitoring - a State-of-the-art Review and Directions for Future Research, 120, pp. 85–102.

Attewel, P., Monaghan, D., 2015. Data Mining for the Social Sciences: an Introduction. University of California Press, Oakland.

Azur, M.J., Stuart, E.A., Frangakis, C., Leaf, P.J., 2011. Multiple imputation by chained equation: what is it and how does it work? Int. J. Methods Psychiatr. Res. 49, 40–49.

AzureML Team, M., 2016. AzureML: anatomy of a machine learning service. In: JMLR: Workshop and Conference Proceedings.

Ben-Hur, A., Ong, C.S., Sonnenburg, S., Scholkopf, B., Ratsch, G., 2008. Support vector machines and kernels for computational biology. PLoS Comput. Biol. 4 (10), 1–10.

Berardi, U., 2015. Building energy consumption in US, EU, and BRIC countries. Procedia Eng. 118, 128–136.

Botchkarev, A., 2019. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: properties and typology. Interdiscipl. J. Inf. Knowl. Manag. 14, 45–76.

Cai, E., 2019. Learning-based Power and Runtime Modeling for Convolutional Neural Networks. Data Analysis Project.

Cheng, J., Kunz, T., 2009. A Survey on Smart Home Networking. Systems and Computer Engineering Technical Report, SCE-09-10. Carleton University.

Corgnati, S., Bednar, T., Jang, Y., Yoshino, H., M, F., S, D., N, D., M, S., Ghiaus, C., A, C., N, T., V, F., 2013. Total Energy Use in Buildings. Analysis and Evaluation Methods, Final Report Annex 53. Statistical Analysis and Prediction Methods.

Demirtas, H., Freels, S.A, Yucel, R.M, 2008. Plausibility of multivariate normality assumption when multiply imputing non-Gaussian continuous outcomes: a simulation assessment. J. Stat. Comput. Simulat. 78 (1), 69–84. https://doi.org/10.1080/10629360600903866.

El Khantach, A., Hamlich, M., Belbounaguia, N.E., 2019. Short-term load forecasting using machine learning and periodicity decomposition. AIMS Energy 7 (3), 382–394.

Fontama, V., Barga, R., Wee, H.T., 2014. Predictive Analysis with Microsoft Azure Machine Learning: Build and Deploy Actionable Solutions in Minutes, 1 ed. Apress.

Fu, Y., Li, Z., Zhang, H., Xu, P., 2015. Using support vector machine to predict next day electricity load of public buildings with sub-metering devices. Procedia Eng. 121, 1016–1022.

Gnecco, G., Sanguineti, M., 2009. The weight-decay technique in learning from data: an optimization point of view. Comput. Manag. Sci. 6, 53–79. https://doi.org/10.1007/s10287-008-0072-5.

González-Briones, A., Hernández, G., Corchado, J.M., Omatu, S., Mohamad, M.S., 2019. Machine learning models for electricity consumption forecasting: a review. In: IEEE Virtual-Ledgers-Tecnologías DLT/Blockchain y Cripto-IOT.

Hassan, J.S., Zin, R.M., Majid, M.Z.A., Balubaid, S., Hainin, M.R., 2014. Building energy consumption in Malaysia: an overview. Jurnal Teknologi 707 (7), 2180–3722.

Hegde, H., Shimpi, N., Panny, A., Glurich, I., Christie, P., Acharya, A., 2019. MICE vs PPCA: missing data imputation in healthcare. Inf. Med. Unlocked 17. https://doi.org/10.1016/j.imu.2019.100275.

Kang, H., 2013. The prevention and handling of the missing data. Kor. J. Anesthesiol. 64 (5), 402–406.

Karunathilake, S.L., Nagahamulla, H.R.K., 2017. Artificial neural networks for daily electricity demand predicitons of Sri Lanka. In: International Conference on Advances in ICT for Emerging Regions (ICTer).

Kumar, Y., Sahoo, G., 2012. Analysis of parametric & non parametric classifiers for classification technoque using WEKA. Int. J. Inf. Technol. Comput. Sci. 7, 43–49.

Landset, T., Khoshgoftaar, T.M., Richter, A.N., Hasanin, T., 2015. A survey of open source tools for machine learning with big data in the Hadoop ecosystem. J. Big Data 2, 24.

Larasati, A., Dwiastutik, A., Ramadhanti, D., Mahardika, A., 2018. The effect of kurtosis on the accuracy of artificial neural network predictive model. In: MATEC Web of Conferences, 204.

Larasati, A., Hajji, A.M., Dwiastuti, A., 2019. The relationship between data skweness and accuracy of artificial neural network predictive model. IOP Conf. Ser. Mater. Sci. Eng. 523.

Liu, Y., He, X., Xu, B., 2010. Evaluation and comparison of compactly supported radial basis function for kernel machine. In: IEEE International Conference on Intelligent Systems and Knowledge Engineering, Hangzhou.
Liu, Z., Wu, D., Liu, Y., Han, Z., Lun, L., Gao, J., Jin, G., Cao, G., 2019. Accuracy analyses and model comparison of machine learning adopted in building energy consumption prediction. Energy Explor. Exploit. 1–26.
Manly, C.A., Wells, R.S., 2015. Reporting the use of multiple imputation for missing data in higher education research. Res. High. Educ. 56 (4), 397–409.
Mateev, M., 2019. Management of smart energy systems with IoT-based predictive analytics for more efficient energy consumption. In: ATINER's Conference Paper Proceedings Series: COM2019-0163 , Athens.
Mazlan, N.L., Ramli, N.A., Awalin, J.L., Ismail, M.B., Kassim, A., Menon, A., 2020. A smart building energy management using Internet ofThings (IoT) and machine learning. Test Eng. Manag. 83, 8083–8090.
McCarthy, R.V., McCarthy, M.M., Ceccucci, W., Halawi, L., 2019. Applying Predictive Analytics: Finding Value in Data. Springer, Cham, Switzerland.
Microsoft Customer Stories https://customers.microsoft.com/en-us/story/bp-mining-oil-gas-azure-machine-learning (accessed 24 May 2019).
Mishra, P., Pandey, C.M., Singh, U., Gupta, A., Sahu, C., Keshri, A., 2019. Descriptive statistics and normality tests for statistical data. Ann. Card Anaesth. 22 (1), 67–72.
Newgard, C.D., Lewis, R.J., 2015. Missing data: how to best account for what is not known. J. Am. Med. Assoc. 314 (9), 940–941.
Newsham, G.R., Birt, B.J., 2010. Building-level occupancy data to improve ARIMA-based electricity use forecasts. In: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building (Build Sys '10), New York, NY, USA.
Nugroho, H., Surendro, K., 2019. Missing data problem in predictive analytics. In: 8th International Conference on Software and Computer Applications (ICSCA 2019), Penang, Malaysia.
Pérez-Chacón, R., Talavera-Llames, R.L., Martínez-Álvarez, F., Troncoso, A., 2016. Finding electric energy consumption patterns in big time series data. In: Distributed Computing and Artificial Intelligence, 13th International Conference.
Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J., Fong, A.C., 2019. Leveraging machine learning and big data for smart buildings: a comprehensive survey. IEEE Access 7, 90316–90356.
Quddus, J., 2018. Machine Learning with Apache Spark Quick Start Guide. Packt Publishing.
Ramsundar, B., Zadeh, R.B., 2018. TensorFlow for Deep Learning, 1 ed. O'Reilly Media, Sebastopol, California.
Sampat, M.P., Bovik, A.C., Aggarwal, J.K., Castleman, K.R., 2005. Supervised parametric and non parametric classification of chromosome images. Pattern Recogn. 38.
Schmitt, P., Mandel, J., Guedj, M., 2015. A comparison of six method s for missing data imputation. J. Biometrics Biostat. 6 (1).
Siddiqui, F., Ali, Q.M., 2016. Performance of non-parametric classifiers on highly skewed data. Global J. Pure Appl. Math. 12 (3), 1547–1565.
Stenudd, S., 2010. Using Machine Learning in the Adaptive Control of a Smart Environment, vol. 751. VTT PUBLICATIONS, Vuorimiehentie, p. 82.
Tenaga Nasional Berhad, 2006. Tenaga Nasional Berhad tariff book. https://www.tnb.com.my/residential/pricing-tariffs.
Tilmann, G., 2010. Making and evaluating point forecast. J. Am. Stat. Assoc. 106 (494), 746–762.
Valgaev, O., Kupzog, F., Schmeck, H., 2016. Low-voltage power demand forecasting using K-nearest neighbors approach. In: IEEE Innovative Smart Grid Technologies - Asia (ISGT - Asia), pp. 1019–1024.
Vasile, A.N., Camps, O., 2013. Hierarchical image geo-location on a world-wide scale. In: Advances in Visual Computing: 9th International Symposium (ISVC).
Wei, R., Wang, J., Gan, Q., Dang, X., Wang, H., 2019. Predicting electricity usage based on deep neural network. In: 2019 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Tianjin, China.
Weissbart, L., Picek, S., Batina, L., 2019. On the Performance of Multilayer Perceptron in Profiling Side-channel Analysis. Cryptology ePrint Archive. Report 2019/1476.
Wolpert, D.H., 1996. The lack of A priori distinctions between learning algorithm. Neural Comput. 1341–1390.
Xiangyu, Z., Qingqing, M., Tao, L., Lei, Z., Jianquan, Z., 2019. Linear regression electricity prediction method based on clustering of electric characteristics. In: ICBDT2019: Proceedings of the 2nd International Conference on Big Data Technologies, Jinan.
D. Xu, Z. Li, S. Yang, Z. Lu, H. Zhang, W. Chen, and Q. Xu, A classified identification deep-belief network for predicting electric-power load, 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) 1-6.