1-1-2019

Distribution Level Building Load Prediction Using Deep Learning

Abdulaziz S. Almalaq
University of Denver

Distribution Level Building Load Prediction Using Deep Learning

A Dissertation

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Abdulaziz S. Almalaq

August 2019

Advisors: Dr. Amin Khodaei and Dr. Jun Jason Zhang

Author: Abdulaziz S. Almalaq
Title: Distribution Level Building Load Prediction Using Deep Learning
Advisors: Dr. Amin Khodaei and Dr. Jun Jason Zhang
Degree Date: August 2019

Abstract

Load prediction in distribution grids is an important means to improve energy supply scheduling, reduce the production cost, and support emission reduction. Determining accurate load predictions has become more crucial than ever as electrical load patterns are becoming increasingly complicated due to the versatility of the load profiles, the heterogeneity of individual load consumptions, and the variability of consumer-owned energy resources. However, despite the increase of smart grids technologies and energy conservation research, many challenges remain for accurate load prediction using existing methods. This dissertation investigates how to improve the accuracy of load predictions at the distribution level using artificial intelligence (AI), and in particular deep learning (DL), which have already shown significant progress in various other disciplines.

Existing research that applies the DL for load predictions has shown improved performance compared to traditional models. The current research using conventional DL tends to be modeled based on the developer's knowledge. However, there is little evidence that researchers have yet addressed the issue of optimizing the DL parameters using evolutionary computations to find more accurate predictions. Additionally, there are still questions about hybridizing different DL methods, conducting parallel computation techniques, and investigating them on complex smart buildings. In addition, there are still questions about disaggregating the net metered load data into load and behind-the-meter generation associated with solar and electric vehicles (EV).

The focus of this dissertation is to improve the distribution level load predictions using DL. Five approaches are investigated in this dissertation to find more accu-

rate load predictions. The first approach investigates the prediction performance of different DL methods applied for energy consumption in buildings using univariate time series datasets, where their numerical results show the effectiveness of recursive artificial neural networks (RNN). The second approach studies optimizing time window lags and network's hidden neurons of an RNN method, which is the Long Short-Term Memory, using the Genetic Algorithms, to find more accurate energy consumption forecasting in buildings using univariate time series datasets. The third approach considers multivariate time series and operational parameters of practical data to train a hybrid DL model. The fourth approach investigates parallel computing and big data analysis of different practical buildings at the DU campus to improve energy forecasting accuracies. Lastly, a hybrid DL model is used to disaggregate residential building load and behind-the-meter energy loads, including solar and EV.

Acknowledgements

It is a privilege for me to be a student under the supervision of Dr. Amin Khodaei, who is the leader of the KLAB. I am glad that I have joined the fantastic team based on his recommendation lately. Also, I am very thankful for his advising, support, and encouragements. His guidance helped me to overcome struggles during my research. This dissertation would not have been possible without his support and guidance. Also, I would like to thank my co-advisor, Dr. Jun Zhang, for his help and advice during my Ph.D. studies. His support helped me to search in deep learning and conduct artificial intelligence applications.

In addition, I would express my gratitude to my Ph.D. committee members Dr. Ali Besharat, Dr. David Gao, and Dr. Mohammad Matin, for their precious time spent to review my dissertation and for their positive feedbacks. Also, I would like to thank Dr. George Edwards for his help many times during research.

After all, I would like to express my genuine appreciation to my beloved family for their continuous support throughout my life. In particular, I am very thankful to my parents (generous father Mr. Saleh and great mother Ms. Fatemah) for their decent support, constant encouragement, and everything beautiful in my life. My special thanks go to my lovely wife Ms. Shahad, for her continuous support and motivations, especially during my rough times, and for every pleasant time. Also, my acknowledgments go to all my sisters, who always prayed for me to achieve my goals, and to my sweet kids (Yasmin and Saleh).

As importantly, I would like to thank all my friends, colleagues, and lab mates for their collaboration, and for the good times that we have had in the last few years. Last but not least, I would like to thank the University of Hail, which granted my scholarship for continuing my education.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Energy consumption in the building sector (residential and commercial) accounts for 40% of total energy production in the United States [1], and this percentage is increasing based on national trends. Thus, smart grids need reliable systems that have intelligent features to monitor, learn, predict, and make decisions at the distribution level to dynamically adapt to changes on demand.

Load prediction of future energy consumption can be perceived as load forecasting and energy disaggregation. In the future, load prediction will be important for all energy market participants. The load prediction will play a key role in mitigating uncertainty in the future energy sector because it provides the information needed to plan and maintain the operation of the power system. In addition, it provides the ability to apply demand response to residential buildings, and the ability to have dynamic demand pricing. In the energy market, accurate load prediction will assist participants to increase their profits and decrease their losses by planning appropriate corrections in their systems. Hence, accurate load prediction in buildings will help system planner to make precise plans for future budgets and energy costs.

1.1 Smart Grid

The concept of smart grids is a modern power system infrastructure that aims to build robust, reliable, efficient grids and minimize the cost of production. Enhancing the grids with renewable energy resources, automated control, and communication technologies provides possible means of efficiency, reliability, and safety for smart grids. The objective of smart grids is to advance the use of information and communication technologies by investing in the bidirectional flow of power and data. Smart grids infrastructure is full of advanced sensing, communicating and computing abilities that work in an interoperable way in different power system parts, generation, and distribution [2]. The effectiveness of smart grids relies on three primary roles that can help maintain and manage the grids as follows:

1. Dynamic pricing
2. Demand-side management
3. Load forecasting

1.2 Electricity Load at Distribution Level and Buildings

The electrical load at the distribution level is oscillatory and subject to change because human activities follow daily, weekly and monthly event cycles. For instance, the load is generally higher in the daytime and early evening, but it is lower in the late evening and early morning. This means that every electrical appliance or light bulb that is switched on or off by customers can directly affect the electrical load seen on the distribution feeder. In general, customers buy electricity from providers to power their appliances. Therefore, the distribution system exists to deliver energy to customers in the form of electrical appliances and equipment, lighting, heating, and cooling as well as other demands in the commercial and industrial sectors. The distribution system of the smart grids must satisfy customer needs in order to deliver a high quality of service.

The building sector including residential and commercial accounts for a significant percentage of total energy production and is deemed as a major energy consumer globally [1], [3], [4]. Thus, it is important to manage energy efficiency, with objectives of energy conservation and environmental impact reduction. However, there are many restrictions for the energy reduction on the residential and commercial buildings because the priority of building management is to keep the indoor environment (lighting and temperature) within a comfort bracket. Therefore, the objective of energy conservation and environmental impact reduction turns into a trade-off between indoor comfort and energy reduction.

1.3 Load Prediction Models at Distribution Level

Load prediction is a technique usually used by energy suppliers to forecast future energy consumption to meet the load demand and supply balance in the generation, transmission, and distribution sectors. Generally, a prediction technique is used to forecast future load, electricity price, wind power, and solar power. Household owners also use it; building managers in the commercial sector or energy supervisors in the industrial sector apply it to meet their energy requirements and build their bidding strategies. Therefore, the load prediction strategy is indispensable for all active energy market players. We define the three main categories of forecasting and their objectives as follows [5] [6]:

- *Long-term load forecasting (LTLF)*: The time interval of this type of forecasting lies from five years to decades in the future. The LTLF application is mainly for the generation and transmission systems which aim to plan for the future electricity capacity.

- *Medium-term load forecasting (MTLF)*: The forecasting time interval of this type prevails from a month to five years. The purpose of the MTLF is essentially to plan for near future power plants and show the dynamics of the smart grid.

- *Short-term load forecasting (STLF)*: This type handles time horizons of a single hour up to a couple of weeks. The STLF is necessary for the scheduling of power plants. In addition, the applications of this type of forecasting include real-time control, energy transfer scheduling, economic dispatch, and demand response.

1.4 Common Evaluation Metrics of Prediction Models

For the prediction models, various evaluation criteria are utilized in the literature to evaluate the performance results. The first criterion is directly using the 30% testing dataset to examine the performance of the prediction model. The second criterion of model performance evaluation is the metrics calculation where the conventional methods are the root-mean-squared error (RMSE), the coefficient of variation (CV) of the RMSE which is called normalized RMSE (NRMSE), the mean absolute error (MAE), the and mean absolute percentage error (MAPE) defined as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - y_i)^2} \tag{1.4.1}$$

$$CV = \frac{RMSE}{\bar{x}} \times 100\% \tag{1.4.2}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|x_i - y_i| \tag{1.4.3}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\frac{|x_i - y_i|}{y_i} \times 100\% \tag{1.4.4}$$

where m represents the total number of data points in the time series, x_i is the real measured time series in the original scale of the dataset, y_i is the predicted output of the time series, and $\bar{x}$ is the average of the actual values of energy consumption. The third criterion of model evaluation is that the model is benchmarked with conventional prediction methods such as statistical, ML and DL models.

The last criterion to examine the performance of the prediction model is cross-validation which splits the dataset into k-fold subsets to estimate the general performance of the prediction model and gives an insight on how the model generalizes the independent variables throughout the datasets. The method repeats the process of splitting the dataset into training and testing portions for k-times where the size of the testing data remains fixed but moves through the original dataset and the remainder used as training dataset every fold as in Fig. 1.1.

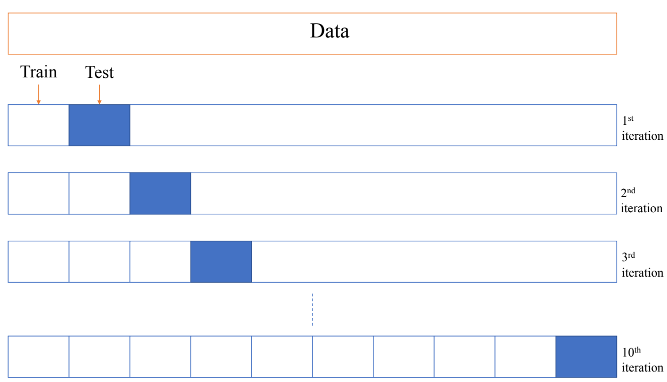


Figure 1.1: Cross validation method with kth folds.

Applying this method to the prediction model produces a robust averaged estimation of the prediction when each observation in the dataset is used for training and testing at each fold.

1.5 Existing Research on Load Prediction

Investigating the load prediction has gained wide attention, although it has been shown as a challenging problem due to the vast dependencies of the time series. Fortunately, the forecasting accuracy is promoted by the diversity of end-users behaviors. For building power consumption, daily routine, occupancy, and type of appliances are common factors that have a higher impact on load demand than underline features like temperature and weather which have the high correlation to the seasons and events. For instance, if a building has no occupancy for a particular time and the temperature was high, the power consumed by this building would not affect the overall electricity demand. Therefore, the power consumption load is dependable on the behavior of end-users more than correlated factors, and our proposed forecasting model investigates the seasonality, different timescales, and user behaviors.

1.5.1 Traditional Prediction Models

Most of the previous studies focus on predicting electricity demand on a system or a substation level, where this research emphasizes forecasting the load at end-users levels. There are two major conventional load forecasting approaches in the literature, which are physical based models and statistically based models, for example, Autoregressive Integrated Moving Average (ARIMA) in [7], [8], [9]. In the era of the AI, many applications in the power system gained from the intelligence of computation including the load forecasting application. There are many approaches to the AI used to automate, predict, classify and cluster power consumption applications. The most commonly used approach for load forecasting is machine learning (ML), which is a subfield of the AI, such as Artificial Neural Network (ANN) in [10], [11], [12], [13], [14], [15]. Moreover, another widely ML technique is Support Vector Machine (SVM) used to predict the power consumption at the building level in [16],

[17], [18], [19]. Decision Tree (DT) in [20], [21], [22] and k-nearest neighbor (kNN) in [23], [24] were applied for time series forecasting.

1.5.2 Deep Learning Methods Applied on Load Forecasting

Deep learning (DL), which is an advanced ML method by adding multi-hidden layers to the standard ML neural network, has gained a wide attention across a range of disciplines, for example, image processing, speech recognition, natural language processing, finance, and sequential problems due to state of the art results and precise forecasting accuracy.

The Autoencoder DL algorithm is used for STLF of the electricity price [5]. The author of this study proposed Stacked Denoising Autoencoders for short term forecasting. The forecasting approaches are designed for two models online forecasting and one day ahead forecasting. Stacked Denoising Autoencoder showed effective forecasting results especially for day ahead forecasting [25]. The results are compared with state of art forecasting methods such as classical neural network (NN), SVM, multivariate adaptive regression splines (MARS) and least absolute shrinkage and selection operator (Lasso) [25]. Another study has investigated Autoencoder and Long Short Term Memory (LSTM) for forecasting renewable energy power plants [26]. The study combined the Autoencoder and LSTM for forecasting and compared the results to the state of art approaches such Artificial Neural Network, LSTM and DBN [26]. The method is applied for 21 solar power plants when proposed method showed decreasing in the average of RMSE for training and testing results.

A recent study for household load forecasting has used a novel approach of Pooling Deep Recurrent Neural Network (PDRNN) [27]. The authors claimed that adding more layers to the neural network improves the forecasting performance [27]. This study has used a pool of inputs for a group of customers to increase the data diversity and volume [27]. The data of this study is collected from 920 smart metered customers in Ireland. The authors also used GPU in their hardware in order to accelerate the computational time by parallelizing the models. The results were compared to the literature load forecasting techniques such as ARIMA, SVR and DRNN [27]. This novel approach showed improvement results in RMSE when compared with; ARIMA by 19.5%, SVR by 13.1% and DRNN by 6.5% [27].

One of the common methods of DL which is been used in short term load forecasting is LSTM. This method is used for two approaches in building energy load such as standard LSTM and LSTM-based sequence to sequence (S2S) [28]. They are implemented on consumption dataset of residential load that is trained and tested with one hour and one minute resolution [28]. In this study, the standard LSTM performed successfully in one hour resolution, however, failed in one minute resolution [28]. For the second approach, both dataset resolutions performed well compared to results to the literature [28]. Another study investigated LSTM for residential load for short term forecasting has studied the usage of appliances for individual load forecasting [24]. The researchers showed their algorithm outperforms the state of art approaches of residential load forecasting [24].

Convolutional Neural Network (CNN) with k-mean clustering was applied in short term forecasting [29]. The K-mean algorithm is used on a large dataset to create clusters for training the CNN [29]. The study selected the data from August 2014 for summer data and from December 2014 for winter data to apply the method [29]. The comparison results in this study shows great improvement in terms of RMSE for CNN with k-means 0.2194 in summer experiment and 0.2399 in winter experiment. Neural network was applied to both experiment which resulted RMSE 0.2379 in summer and 0.2839 in winter. The author's also applied CNN only on the datasets in order to compare the results with the proposed method. CNN only resulted in lower accuracy in terms of RMSE 0.2502 in summer and RMSE 0.2614 in winter. The study concluded that CNN outperforms other methods with help of clustering techniques [29].

A study that has demonstrated great performance using DL in costumer's load electricity forecasting used Restricted Boltzmann Machine (RBM) method [30]. The training process in the study was driven by two cases pre-training process and Rectified Linear Unit without pre-training. DL approach is structured by using a heuristic method to determine the number of hidden neurons that are used in each hidden layer [30]. There were four hidden layers and 150 hidden neurons. A sigmoid function is used for RBM pre-training. A linear function is used for the prediction layer which is the output layer [30]. DL forecasting results are compared with other known methods in the literature for forecasting such sallow neural network (SNN), double seasonal Holt-Winters (DSHW) and ARIMA [30]. The results are compared with the previous work in the literature to verify the performance of the proposed approach of DL short term load forecasting. In addition, the results are verified by MAPE and relative RMSE [30]. This approach reduced relative RMSE by 22% in comparison with SSN and by 29% in comparison with DSHW [11].

Deep belief network (DBN) algorithm was proposed in a study for load short forecasting [31]. In this study, the authors aimed to improve the performance of DBN with load forecasting by using ensemble methods and emerge Support Vector Regression (SVR) [31]. The authors used three electricity load demand datasets and three regression datasets to apply their proposed method. The results show that the ensemble DL by combining DBN with SVR outperformed SVR, Feedforward NN, DBN and ensemble NN [31]. The prediction results for load demand of South Australia for the proposed method with RMSE 30.598, however, RMSE for SVR is 44.674 and RMSE for Feedforward NN is 38.8585.

A study that proposed Predictive Deep Boltzmann Machine (PDBM) for predicting wind speed was applied to wind energy [32]. The authors used raw wind speed data to forecast short term and long term wind speed. The algorithm was used to predict wind speed one hour ahead and one day ahead. The proposed method showed improved performance in comparison with autoregressive (AR), adaptive neuro-fuzzy inference system (ANFIS), and support vector machine (SVM) for support vector regression (SVR) [32]. The study was made for short term and long term forecasting. Short term forecasting is designed to forecast from 10 minutes ahead to 2 hours ahead. The result of 10 minutes ahead in terms of RMSE is 0.2951 for the proposed method as correspond to the RMSE for SVR is 0.6340. The proposed algorithm for one day ahead forecasting resulted RMSE 1.2926 which is an improvement over SVR that has RMSE 1.3678.

The following table summarize most of the reviewed papers in terms of the comparison between the proposed method in each paper and common benchmark method in the state of art. Table 2.2 shows RMSE percentage of reduction in comparison with the other method. Table 2.2, the reduction of RMSE in the method of ensemble DBN and SVR is remarkable in comparison with FNN. In addition, the method of PDRNN shows great reduction in comparison to ARIMA. Then, CNN with k-means showed also great percentage of reduction in RMSE.

Table 1.1: Shows percentage of reduction for RMSE in comparison between proposed method and benchmark state of art method.

Benchmark method	Proposed Method	RMSE % of reduction
MLP	Autoencoder and LSTM	5.51%
ARIMA	PDRNN	19.2%
NN	CNN	7.7%
CNN	CNN with k-means	12.3%
FNN	DBN and SVR	21.2%
SVR	PDBN	2.85%

1.5.3 Evolutionary Computation Applied on Prediction Models

In the last decade, many intelligent evolutionary computations based on optimization methods have been applied to the problem of energy consumption in buildings, e.g., Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Evolution Strategies (ES). These methods are types of metaheuristic optimization techniques that are nature inspired in mathematical optimization processes. In terms of forecasting chaotic time series, the PSO method improved the results of the ANN predictive model in [33] [34] [35]. For the problem of energy consumption prediction, PSO-ANN, and GA-ANN hybrid prediction methods applied with principal component analysis to select relevant input energy variables in [34]. The hybrid approaches resulted in better performance than regular ANN, where they had the same accuracy level. In addition, the GA was employed to improve Adaptive Network-based Fuzzy Inference Systems using two building datasets of Great building Energy Predictor Shootout and a library building in [36]. The optimization population-based research found the better performance of hybrid predictive models than regular ones. For the problem of time series, the ES was used to improve the ANN training models and converges faster to optimal solution [37].

1.5.4 Multivariate Time Series Prediction Models

Hence, the recent development of load prediction using appliance consumption data with the LSTM method is presented in [24]. The author concluded that by adding power consumption measurements of major appliances, the performance is improved since the residential electrical load is more related to residents' behaviors. Therefore, it was claimed that the predictive model was improved when more variables and parameters were added to an energy consumption dataset. In [38], the authors used multivariable inputs including outside temperature and operating parameters to predict the thermal load. The multivariable prediction model using Support Vector Regression performed better than the single variable prediction model. The multivariate regression (MR) was used for the prediction of residential energy consumption in [39]. The kNN and DT were implemented for multivariable time series (MTS) prediction in [40] and [41], respectively. In addition, Gated Recurrent Unit (GRU) model utilized with the MTS to predict missing data in [42] and achieved the best performance compared with conventional prediction models.

1.5.5 Energy disaggregation and prediction models

Tremendous research has been carried out to advance energy management systems and solve the problem of energy disaggregation for buildings since the 1980s. One of the initial studies was investigated by Hart in [43] using the event-based method which classifies the change and difference of steady-state and transient state consumptions. In this analysis, the author investigated household load appliances using edge detection of different consumption states. Moreover, in [44], the authors utilized edge detection based on analyzing the variations in active and reactive power. In general, the edge detection technique compares the variations and detects the threshold of the on/off states at the observed voltage, current, active power, reactive power, and power factor; the event is defined when the load exceeds the predefined set time [45]. Another event-based analysis using harmonic pattern recognition method, which provides extra information to the appliance load signatures, was utilized for the NILM in a commercial building in [46]. In this analysis, The authors applied their method using odd harmonic current patterns and phase difference related to the fundamental voltage.

Nonevent-based methods learn the consumption events automatically without the need to breakdown events in the consumption waveforms. These methods mostly use temporal graphical models like Hidden Markov models (HMM) in [47]. In this investigation, the authors used a factorial HMM technique for nonevent energy disaggregation, but this technique fails in local minima easily. In addition, the HMM technique was utilized for multi-state appliances modeling in [48] and [49]. The disadvantage of this approach for multi-state disaggregation is that fails with continuously varying loads, e.g., laptops [50].

The learning models of the energy disaggregation problem in the literature use a combination of non-supervised and supervised techniques. The non-supervised modeling, which does not require prior training with labeled data, was presented in [51], [52] and [53], and it is a desirable learning method, but it is more difficult to obtain results compared to supervised learning techniques. The supervised modeling technique, which requires a ground truth of input and output data for each class label in order to predict new unseen data, was presented in [54]. Therefore, it learns from the aggregated data and the labels of individual appliances data. Furthermore, it is used to assign the defined consumption sources to detected events and classify them accordingly in order to predict the energy consumption of individual appliances [54].

A variety of machine learning methods have been used to solve the classification problem in the energy disaggregation. The classification method technique using Support Vector Machine (SVM) was utilized in [55], [56], [57], and [58]. The clustering and classification techniques of k- Nearest Neighbors (k-NN) method was used quite frequently in [59], [60], [61], [62] and [63]. Moreover, Artificial neural networks (ANN) was applied to the energy disaggregation in [64], [65], [66], [67], and [68].

Recently, DL methods have been utilized to enhance the energy disaggregation in buildings. The classification technique of CNN method was utilized to identify residential equipment in [69] and [70]. LSTM was applied to the NILM problem using the recurrent neural network in order to supervise power disaggregation in [54], [71], and [72]. The classification technique of GRU method was conducted on energy disaggregation in [73].

Constructive surveys, which have been carried out to investigate the state of the art research of the energy disaggregation problem, can be found in [74], [75], and [76]. Although the literature is extensive, lacking is an effective method to disaggregate residential loads when new types of distributed energy resources (DERs) such as solar photovoltaic (PV) and electric vehicle are used.

1.6 Research Motivation and Main Contributions

The objective of this dissertation is to develop a robust and accurate prediction model that can improve the uncertainties of load forecasting, energy consumption prediction, and individual energy disaggregation at the distribution level. The following are the motivations and primary objective of each approach in this dissertation.

Commonly, many hyper-parameters of the DL network, such as the number of hidden layers, the number of hidden neurons, activation function, etc., are influential factors in the energy prediction model. If the selected hyper-parameters of the predictive DL model are unsuccessful, the model performs poorly and will lead to local optimum results. In addition, the window size or time lags of the input variables play another significant role in terms of finding optimum prediction value. Selecting the right hyper-parameters and the optimized window size is an optimization process that improves the accuracy of the prediction model. In [77], a literature review shows that evolutionary computation concepts are used to improve ML algorithm prediction, such as ANN and Fuzzy logic. Thus, there is a need to be employed to the DL algorithms, such as for the LSTM since it has proven better prediction performance in the literature. The motive of optimization predictive DL model using GA, and the research objective is to find a global or near-global optimum prediction error in the problem of building's energy consumption prediction by searching in a population base of the LSTM hyper-parameters and window size.

In addition, most existing studies focused on predicting energy consumption by using historical energy data, and a few included major appliances to improve prediction performance. However, there are still questions regarding incorporating all load-related building operational parameters into energy consumption prediction. We attempt to solve the following questions. How should smart buildings be modeled using operational parameters? How to modify and update the model continuously? The primary motivation of this research is to create a new paradigm, that energy consumption prediction can be modeled by a large number of building's parameters using hybrid DL method (LSTM-GRU), and thus provides the feasibility of precise energy management.

Furthermore, there are still questions regarding hybridizing two significant DL methods (CNN-GRU) to improve forecasting accuracies, using big data for training the forecasting models and formulating the forecasting model for online prediction. Also, there are still questions regarding different time resolutions of load forecasting. In this research, we try to solve the following open questions. How should a robust power consumption forecasting be modeled with hybrid DL methods for STLF? How to modify and update the model continuously with parallel computation? Moreover, how accurate is the corresponding forecasting using big datasets and different timescales resolutions such as (minutes, hours, days, weekdays, and seasons)? The primary motivation of this research is to create a new paradigm, that power consumption can be modeled by a parallel forecasting model, and using big dataset for training, and thus provides the feasibility of real-time forecasting and large-scale complex systems.

Still, a real challenge is that the available energy disaggregation cannot adequately capture localized generation, which is an extremely pressing issue due to the growing penetration of behind-the-meter energy resources. Moreover, electric utilities have largely installed one smart meter for each customer to measure the net load, which masks the local generation. The primary motivation of this research is to create an effective framework of energy disaggregation for a residential prosumer that comprises different electrical loads. The proposed method employs the data collected from the smart meter and trains a hybrid DL model (CNN-LSTM) to classify and determine different electric loads and behind-the-meter generation.

This dissertation work contributed to the solution of accurate load predictions (load forecasting and energy disaggregation) at the building level by using the DL and AI technologies. The major contributions in this dissertation include:

1. A new optimization model using the GA-LSTM method that optimizes the objective function by the time window lags and the network's hidden neurons to find the precision prediction.

2. A complex predictive model of a smart building that incorporates all operational parameters and uses hybrid DL to predict energy consumption in buildings.

3. A parallel computational framework for load forecasting in buildings using AI technologies.

4. A new energy disaggregation paradigm in residential buildings that disaggregates different electric loads and behind-the-meter loads.

5. Hybrid DL predictive models that utilize encoder and decoder for energy predictions in buildings.

1.7 Research Outline

This dissertation is organized as follows:

Chapter 2 investigates the prediction performance of non-recursive and recursive ANNs models used for energy consumption forecasting in buildings. The prediction models use univariate historical energy datasets. Also, the chapter explores the evolutionary DL method used for energy consumption prediction in buildings. The predictive model uses univariate time series for training and testing, with the intent to find optimal time window lags and network's hidden neurons.

Chapter 3 investigates a complex system approach for energy consumption prediction. The model hybridizes two DL methods to find an accurate prediction using multivariate time series datasets for training and testing.

Chapter 4 studies the parallel computation of different buildings to improve load forecasting accuracy. The approach is conducted on practical datasets of a couple of buildings at the DU campus.

Chapter 5 investigates energy disaggregation of residential building and predicts different electrical loads and behind-the-meter loads.

Finally, Chapter 6 concludes all research done for this dissertation and proposes potential future research directions.

Chapter 2

Energy Consumption Forecasting in Buildings

2.1 Recursive and Non-Recursive ANNs in Energy Consumption Forecasting in Buildings

This section investigates the accuracy with which the recursive ANNs algorithms, LSTM and GRU forecasted energy consumption for individual consumers under STLF and MTLF forecasting conditions. The study examined one-day ahead prediction for STLF and one- month ahead for MTLF method. The experiments were run using two historical load datasets of aggregated energy consumption for residential and commercial buildings. The experiment was repeated using the non-recursive algorithms, RBFN and MLP for comparison.

2.1.1 Problem Formulation

The presented approach of energy consumption prediction is to estimate the energy consumption load using historical energy consumption data in individual buildings and building sectors for STLF and MTLF. Let m time steps be a sequence of historical data and $x_i \in \mathbb{R}$, hence, the energy consumption vector is as follows:

$$X = \{x_i = (x_0, x_1, x_2, ..., x_{m-1})\colon x_i \in \mathbb{R}\} \tag{2.1.1}$$

where x_i is the historical energy consumption at a time i where $i \in \{1, 2, ..., m\}$. The prediction vector using sliding window step is defined as:

$$Y = \{y_i = (y_m, y_{m+1}, y_{m+2}, ..., y_{m+n-1})\colon y_i \in \mathbb{R}\} \tag{2.1.2}$$

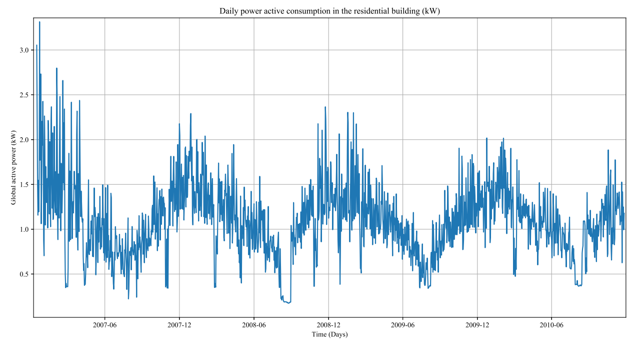
where Y is the predicted energy consumption vector and n is the window size. The prediction objective function is expressed as:

$$\arg\min \sum_{i=1}^{M} (x_i - y_i)^2 \quad \forall x \in X \tag{2.1.3}$$

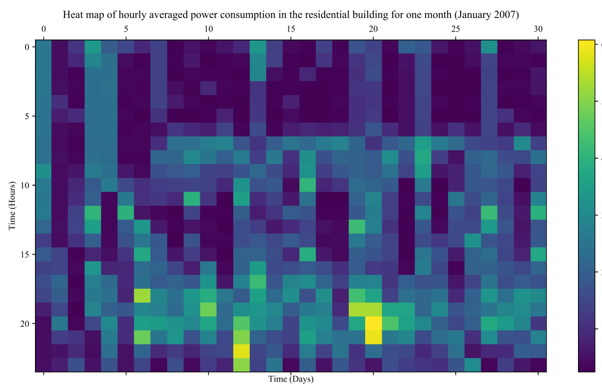
2.1.2 Datasets and Modeling Setup

Residential Building

The public dataset of a single residential building is named as individual household electric power consumption in [78]. The dataset consists of historical energy consumption in kW from December 2006 to November 2010 with one-minute resolution. The model in this research used only the active power consumption of the household from the dataset. The total number of samples in the dataset is more than 2 million time-steps. Fig. 2.1 (a) shows the variation of power consumption with different seasons and days and Fig. 2.1 (b) shows a heat map illustration of the averaged daily power consumption for one month. It is worth noting from the heat map that the residential building has a large volatility of consumption for each day during one month.



(a) Line graph.

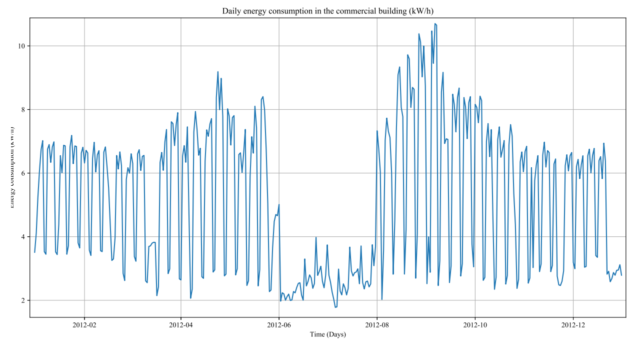


(b) Heat map.

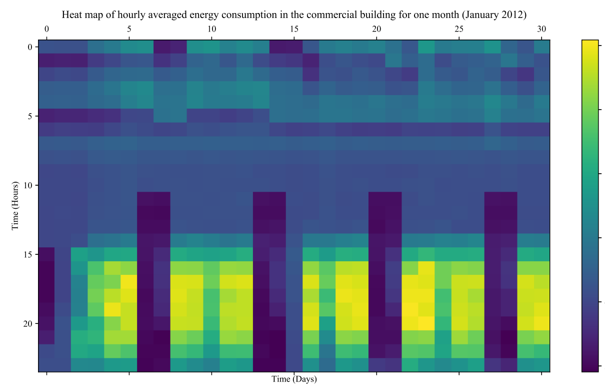
Figure 2.1: The daily average power consumption of residential building.

Commercial Building

The energy dataset of a single commercial building, which is a primary or secondary school in Denver, Colorado, USA, is randomly chosen from a list of publicly published commercial buildings datasets in [79] with the name 213.csv. The data contains energy consumption values in kW/h of one year in 2012 with five minutes resolution where the data size is 105408 time-steps. Fig. 2.2 (a) shows the line graph of daily averaged energy consumption and Fig. 2.2 (b) shows the heat map of averaged daily energy consumption for one month. From the heat map, the commercial building has a consistent high consumption during the working hours. However, the consumption is the lowest in the weekend days.



(a) Line graph.



(b) Heat map.

Figure 2.2: The daily average energy consumption of commercial building.

Modeling Setup

The univariate time series datasets were normalized and split into training and testing datasets for each model with 70% and 30%, respectively. The models of STLF and MTLF case studies have been implemented in Python with Keras package [80]. The two models were designed to predict one-time step for next day energy consumption in the STLF method and for next month energy consumption in the MTLF. The number of hidden neurons in each model for the non-recursive and recursive ANN is 10 hidden neurons with one hidden layer. The activation function was sigmoid for the MLP, LSTM and GRU, however the activation function was radial basis function for the RBFN. The loss function was mean square error and the number of epochs is 300.

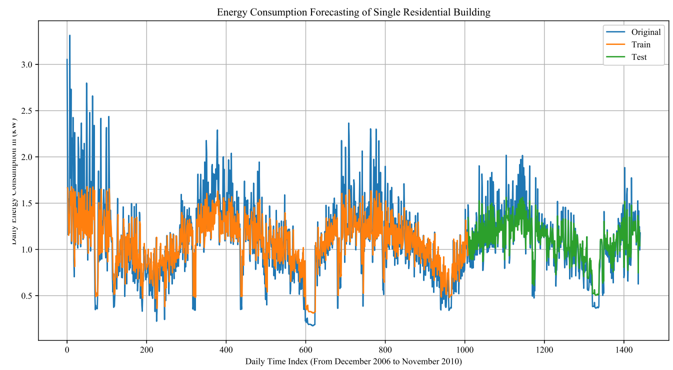
2.1.3 Prediction Results

The first case study, one-day ahead prediction was made using the non-recursive ANN models and recursive ANN models applied to single residential and commercial buildings datasets. The presented models were fed with one of the datasets, which was split for training and testing, to evaluate the performance of the one-day ahead prediction models. Tables 2.1 shows the results obtained fro different ANNs. The results found from testing prediction for one-day ahead accuracies indicate slight changes between different ANNs. The GRU showed the best performance in one-day forecasting. This indicates the effectiveness of the recursive ANN forecasting models in STLF. The prediction performance graphs for the GRU model for next day prediction is shown in Fig. 2.3, where they illustrate one-day predictions of training and testing processes with actual measurement of daily energy consumption in the two single buildings. From Fig. 2.3 (a) and Fig. 2.3 (b), it can be seen that the GRU follows the variation of the daily load in each dataset.

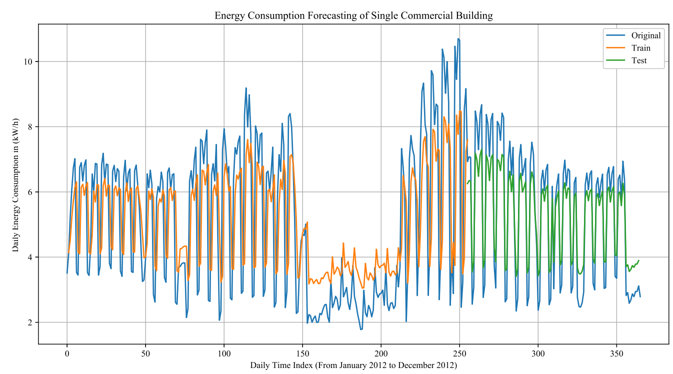
Table 2.1: The metrics evaluation for testing one-day ahead prediction in individual buildings.

-	Single Residential Building		Single Commercial Building	
Model	RMSE (kW)	CV (%)	RMSE (kW/h)	CV (%)
RBF	0.281	25.738 %	1.995	38.686%
MLP	0.277	25.354 %	1.895	36.618%
LSTM	0.266	24.327%	1.889	36.594%
GRU	0.247	24.308 %	1.885	36.587%

In the second case study, one-month ahead prediction was made using the non-recursive ANN models and recursive ANN models applied to residential and commercial buildings energy consumptions. The datasets were split for training and testing with the testing dataset used to evaluate the performance of the one-month looking ahead. Table 2.2 summarizes of the performances of the non-recursive and



(a) Residential Building



(b) Commercial Building

Figure 2.3: Graphs of one-day ahead energy consumption prediction for single buildings. a. Energy consumption prediction of single residential building. b. Energy consumption prediction of single commercial building.

recursive ANN models. The results obtained from testing shows small variation in the performances among different models. The GRU achieved the best prediction accuracy in the residential building, however, the LSTM has the best performance in the commercial building. This demonstrates that the recursive ANN models are robust forecasting in the MTLF.

Table 2.2: The metrics evaluation for testing one-month ahead prediction in individual buildings.

-	Single Residential Building		Single Commercial Building	
Model	RMSE (kW)	CV (%)	RMSE (kW/h)	CV (%)
RBF	0.179	16.417 %	1.877	36.410%
MLP	0.156	14.141 %	1.145	23.003%
LSTM	0.126	11.471 %	1.069	21.474%
GRU	0.126	11.468 %	1.082	21.503%

2.2 Evolutionary Deep Learning Based Energy Consumption Prediction for Buildings

Commonly, many hyper-parameters of the DL network, such as the number of hidden layers, the number of hidden neurons, activation function, etc., are influential factors in the energy prediction model. If the selected hyper-parameters of the predictive DL model are unsuccessful, the model performs poorly and will lead to local optimum results. In addition, the predictive window size or time lags of the input variables play another big role in terms of finding optimum prediction value. Selecting the right hyper-parameters and the fine window size is an optimization process that improves the accuracy of the prediction model. In [77], a literature review shows that the evolutionary computation concepts are used to improve ML algorithm prediction, such as ANN and Fuzzy logic. Thus, there is a need to be employed to the DL algorithms, such as for the LSTM since it has proven better prediction performance in the literature.

The modeling technique presented in this chapter is based on evolutionary DL method which utilizes the GA optimization method to improve the accuracy prediction levels of the LSTM method for the energy consumption in buildings. The proposed approach is compared with the results of conventional predictive models in the literature, e.g, ARIMA, Decision Tree, kNN, multilayer perceptron (MLP), which is a type of ANN with a potential of the deep neural network, and LSTM with different deep architectures. The optimization investigation is modeled by searching for the fine window size and the right number of hidden neurons. The GA-LSTM model is trained and tested with two different building datasets for residential and commercial buildings for very short-term prediction.

2.2.1 Problem Formulation

The energy consumption in a building is a time series problem that has a sequence of observations at time-space as $x_i = \{x_1, x_2, ...\}$ where each observation in $x_i \in \mathbb{R}$ corresponding to a particular time step i. The predicted time series is defined as $y_i \in \mathbb{R}$, which is the energy consumption prediction. The DL model is trained and tested as a supervised learning problem for future time step predictions, where a predictor function h predicts a next step energy consumption value yield as y_{i+1}. In general, the utilized sliding window method for multiple steps prediction (τ) is defined as:

$$y_{i+\tau} = h(x_{i+\tau}, x_{i-1+\tau}, ...x_{i-w+\tau}) \tag{2.2.1}$$

where w is the window size. If the window size $w = 1$, the prediction function will be $y_{i+1} = h(x_i)$.

The optimization technique used with objective function or the loss function is expressed as:

$$\arg\min \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_{i+\tau} - y_{i+\tau})^2} \quad \forall y \in y_i \tag{2.2.2}$$

$$\text{subject to.} \quad \underline{x}_{i-w+\tau} \leq x_{i-w+\tau} \leq \overline{x}_{i-w+\tau}, \tag{2.2.3}$$

where m represents the total number of data points in the time series, $x_{i+\tau}$ and $y_{i+\tau}$ are the real and the predicted energy consumption of future steps, respectively, and $\underline{x}_{i-w+\tau}$ and $\overline{x}_{i-w+\tau}$ are constraints of window size. The objective of the optimizer is to minimize the energy consumption prediction error with a sliding window and a number of hidden neurons in the DL network architecture. The solutions space is defined as $\mathbb{R}$ for the minimization fitness function. The task of the optimization problem is to find a solution $x^* \in \mathbb{R}$ such that:

$$h^* = h(x^*) \leq h(x) \quad \forall x \in x_i \tag{2.2.4}$$

where h^* is a global optimum fitness and x^* is the minimum location in the solutions space.

2.2.2 Modeling Setup

The proposed model in this research is utilized to optimize the prediction error of the LSTM as in Fig. 2.4. The hybrid model of the GA-LSTM is designed with a couple of hidden layers and an optimizable number of hidden neurons besides an optimizable window size. The optimization model schemes of GA-LSTM is shown in Fig. 2.5. The first step of the model is preprocessing the input dataset through normalization method as:

$$x_i' = \frac{x_i - min}{max - min} \tag{2.2.5}$$

where x_i is the original value of the input dataset, x_i' is the normalized value scaled to the range $[0, 1]$, max is the maximum value of the features, and min is the minimum value of the features. Normalizing the dataset features avoids the problem of dominating the large number ranges and helps the algorithm to perform accurately.

The second step is to select the appropriate time lags or window size of the dataset observations and convert the data to a supervised learning form. Then, splitting the data into two main datasets of a training dataset and a testing dataset with the first 70% of the dataset and the last 30% of the dataset, respectively. To evaluate the performance of our proposed model properly, the training data is only utilized separately for the training process in the LSTM and the testing data is used for evaluating the predictive model. For instance, we utilized the first 33 months of residential building data with the one-minute resolution for training the proposed model and 14 months of data for the testing process. Similarly, we used 73785 time-steps of commercial building data for training and the rest is used for testing.

The fourth step is training the model with an initial window size and a number of hidden neurons in the first hidden layer. Then, testing the model by testing set with the selected window size and the number of hidden neurons is performed to calculate the prediction accuracy of the loss function using mean squared error, and the optimizer is stochastic gradient descent (SGD). The total number of epochs of all learning models is 300 epochs when one epoch is a complete pass through the training dataset. An illustration of the LSTM hyper-parameters hybrid with GA are demonstrated in Table 2.3. The window size, and the number of hidden neurons are used to construct a fitness function as in equation (2). The ending condition must be satisfied when the operation ends, otherwise, it will proceed and find a better solution in the next generation. When the condition is satisfied in the first LSTM model with one hidden layer, the model may need to be improved by adding a second hidden layer to the next LSTM model. The best window size and the number of hidden neurons in the first LSTM with one hidden layer will be held and added to the second LSTM model with two hidden layers. The GA process is done in the second LSTM model by only optimizing the number of hidden neurons in the second hidden layer at the second LSTM model.

The evolution base operation, e.g., GA as in Fig. 2.5, is a system to search for better solutions by using evolutionary concepts, including crossover, mutation and selection. Generating new chromosomes of window size and number hidden layers by integrating new behavior of the model to strengthen searching dynamics and improve the prediction accuracy. One of the important features of chromosomes in the GA is genotyping which is the binary coding of the features, and the phenotype refers to decoding parameters to variable values in order to be fed back to the model. The chosen parameters in our experiment, e.g., crossover probability P_{cx}, mutation probability P_M, number of generations M, size of population in each generation N, and the length of the chromosome l are represented in Table 2.4.

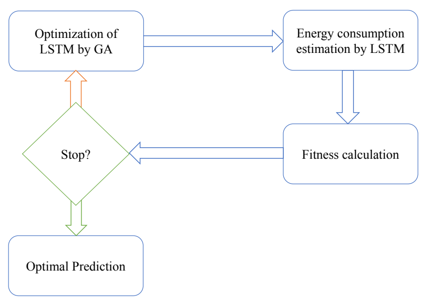


Figure 2.4: The evolutionary DL algorithm scheme.

Table 2.3: The LSTM model hyper-parameters.

Hyper-parameter	Selection
Number of hidden layers (N_l)	1-3
Number of hidden neurons in each layer (N_{np})	Optimizable with GA
Window size (N_t)	Optimizable with GA
Optimizer (opt)	SGD
Loss function	Mean squared error
Number of epochs (N_{ep})	300

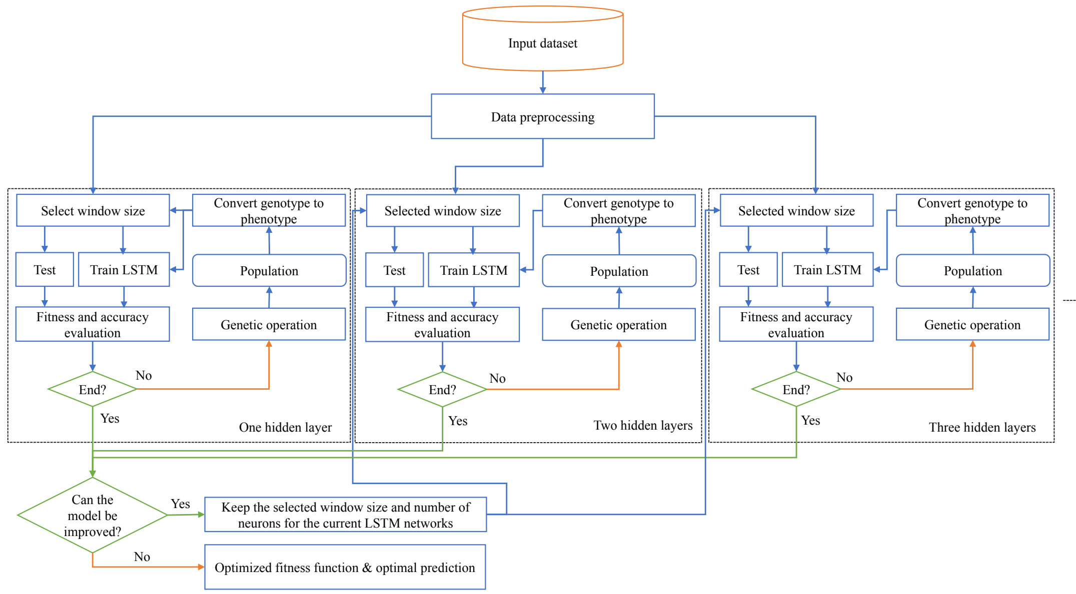


Figure 2.5: The GA-LSTM optimization architecture with three hidden layers.

Table 2.4: The GA model parameters.

Parameter	Selection
Crossover probability (P_{cx})	0.7
Mutation probability (P_M)	0.015
Selection	Tournament selection
Population Size (N)	20
Number of Generations (M)	20
Fitness Function	Root mean square error

2.2.3 Prediction Results

Finding the optimal or near optimal number of time lags and the number of hidden neurons in each layer in the LSTM network is a non-deterministic polynomial (NP) problem which is not easy to solve. The GA algorithm is a promising metaheuristic method which tends to solve such NP problems for good optimal solutions sometimes near to global optimum as found in these studies for time series lags [81] and [82]. Therefore, the number of time lags and the number of neurons are a potent combination of dependencies that affect the prediction process such as model overfitting problem and computation complexity. The selected range of window size or time lags in this experiment is (1-64) time lags and the range of number of hidden neurons in each layer is (1-1024) neurons. The results found in this section are solutions to the NP problem in each LSTM model.

Predicting Residential Building Power Consumption

Table 2.5 illustrates how the performance of the proposed GA-LSTM model compares with those conventional prediction models for the first case study in residential building power consumption. In the table, there are different architectures of regular DL models e.g., MLP-1 with one hidden layer and MLP-2 with two hidden layers. The obtained results show that the proposed model outperformed other models in metrics evaluations. From the table, we find that the two models MLP and LSTM performed in a similar way to the opposite of the proposed method, which overtook them significantly. It is noted that the prediction accuracies get worse when the networks get deeper because of the dependencies of the of the network hyper-parameters. In addition, the statistical model ARIMA and the kNN produced the worst prediction errors in comparison with other learning methods, however, the Decision Tree regression performed better than other conventional models and obtained prediction error close to the DL models. The conventional hybrid model GA-ANN performed better than all conventional methods and traditional DL methods for predicting residential energy consumption, however, the proposed approach outperformed the conventional hybrid model.

Table 2.6 shows the optimal parameters of GA-LSTM-1, GA-LSTM-2 and GA-LSTM-3 and the percentage of reduction in comparison with the LSTM models. We can see the window size is the same for all hidden layers because it is used as an input for the next hidden layer. It is worth noticing that the best percentage of reduction with the regular LSTM-1 model is 17.319 % in terms of RMSE value. In addition, the deeper networks performed good percentages of reduction in terms RMSE values.

Table 2.7 shows the 10-k fold results of the proposed model GA-LSTM-1 that achieved the best prediction from Table 2.5. The prediction error results in each fold are different because the training dataset (D_{tr}) size and testing dataset (D_{ts}) size are shuffled during the process of cross-validation and the final prediction error is averaged over the 10 folds. This validation process of the model increases the confidence of the prediction efficiency because the tested data is different and unseen during the training operation.

Table 2.5: The comparison with conventional methods over one minute resolution for the residential building.

Method	RMSE (kW)	CV (%)	MAE (kW)
ARIMA	0.264	24.170	0.095
Decision Tree	0.233	21.321	0.085
kNN	0.258	23.672	0.111
GA-ANN	0.223	20.158	0.072
MLP-1	0.232	20.934	0.083
MLP-2	0.231	20.844	0.081
MLP-3	0.231	20.844	0.079
LSTM-1	0.235	21.205	0.084
LSTM-2	0.233	21.025	0.084
LSTM-3	0.238	21.476	0.086
GA-LSTM-1	**0.1943**	**17.526**	**0.062**
GA-LSTM-2	0.217	19.581	0.071
GA-LSTM-3	0.225	20.303	0.074

Fig. 2.6 shows a prediction comparison of the residential active power consumption for very short term prediction. The comparison is made for all prediction models given in Table 2.5. From the graph, we can note that the proposed model

Table 2.6: The best parameters GA-LSTM models for the residential building and the percentage of reduction with benchmark LSTM.

Proposed Method			Benchmark	RMSE % of reduction
N_l	N_{np}	N_t	-	Percentage (%)
1	139	23	LSTM-1	**17.319**
2	139 & 43	23	LSTM-2	6.866
3	139 & 43 & 64	23	LSTM-3	5.462

Table 2.7: The 10-fold cross-validation results of GA-LSTM-1 for the first case study.

Fold No.	D_{tr}	D_{ts}	RMSE	CV (%)	MAE
1	188668	188659	0.221	20.238	0.082
2	377327	188659	0.237	21.703	0.085
3	565986	188659	0.220	20.146	0.082
4	754645	188659	0.212	19.413	0.071
5	943304	188659	0.219	20.054	0.073
6	1131963	188659	0.213	19.505	0.071
7	1320622	188659	0.203	18.589	0.069
8	1509281	188659	0.212	19.413	0.071
9	1697940	188659	0.202	18.498	0.069
10	1886599	188659	0.197	18.936	0.066
Mean	-	-	**0.213**	**19.560**	**0.074**
SD	-	-	0.012	1.057	0.007

is superior to the other two DL models benchmarked in this study i.e., MLP and LSTM. The GA-LSTM-1 was the best prediction line graph followed the original data line graph. It is worth noting that the GA-ANN is a skillful model that follows the proposed approach. We can see that the GA-LSTM outperform the models used to predict consumed energy.

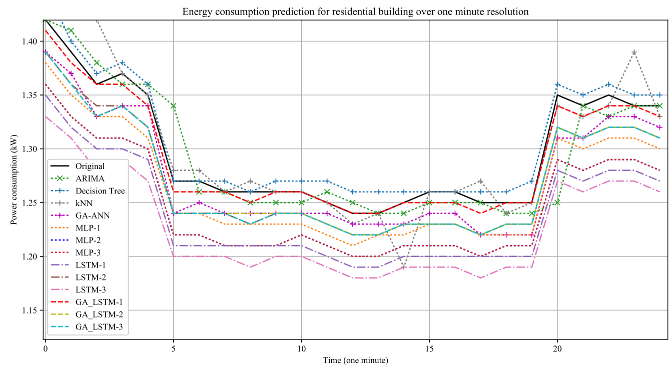


Figure 2.6: Prediction comparison between the proposed model with different conventional prediction models for very short term prediction.

Predicting Commercial Building Energy Consumption

The second case study is predicting commercial building energy consumption as in Table 2.8 which shows how the effectiveness of the proposed GA-LSTM model in comparison with those conventional prediction models. The results from the table show that the proposed method outperformed other methods in prediction accuracies, however, both MLP and LSTM results are close to each other. It is noticeable that the prediction accuracies failed with the deeper network in the conventional methods due to dependencies of the network hyper-parameters. As noted from the first case study and the second case study, the statistical model ARIMA and the kNN were the worst prediction errors in comparison with other learning methods and the Decision Tree regression obtained prediction error close to the DL models. Similarly, the conventional hybrid model GA-ANN obtained better predictions than conventional models and DL models for predicting commercial energy consumption, however, the proposed approach is a superior model to all compared methods.

The optimal parameters of GA-LSTM are given in Table 2.9 where the window size is fixed for all hidden layers because it is used as an input to the next hidden layer in the proposed method. From the table, the percentage of reduction comparison is illustrated and the best percentage is 10.669 % in comparison with LSTM-1. The other two deeper networks performed close to each other in their percentages of reduction.

The 10-k fold results of the best prediction GA-LSTM-1 from Table 2.8 are shown in Table 2.10. From the table, the shuffle operation of the 10-fold cross-validation produced different prediction errors due to the different size of training and testing in each fold. when the tested data is different in each fold and unseen during the training process, the validation technique promotes the certainty of the prediction efficiency of the proposed model.

Table 2.8: The comparison with conventional methods over five minutes resolution for the commercial building.

Method	RMSE (kW/h)	CV (%)	MAE (kW/h)
ARIMA	0.539	10.462	0.297
Decision Tree	0.482	9.353	0.273
kNN	0.544	10.561	0.326
GA-ANN	0.469	9.145	0.268
MLP-1	0.495	9.615	0.305
MLP-2	0.490	9.507	0.295
MLP-3	0.478	9.271	0.271
LSTM-1	0.478	9.283	0.276
LSTM-2	0.486	9.430	0.286
LSTM-3	0.480	9.312	0.276
GA-LSTM-1	**0.427**	**8.303**	**0.238**
GA-LSTM-2	0.451	8.755	0.256
GA-LSTM-3	0.449	8.716	0.263

The prediction performance in Fig. 2.7 shows a comparison between the proposed GA-LSTM and conventional methods of the commercial building for each prediction model. It is noticed from the graph that the proposed model performed better than the other models in this study and followed the original dataset for very

Table 2.9: The best parameters of GA-LSTM models for the commercial building and the percentage of reduction with benchmark LSTM.

Proposed Method			Benchmark	RMSE % of reduction
N_l	N_{np}	N_t	-	Percentage (%)
1	459	42	LSTM-1	**10.669**
2	459 & 187	23	LSTM-2	7.201
3	459 & 187 & 82	23	LSTM-3	6.458

short term prediction. The proposed GA-LSTM proofed its strength over the other compared methods.

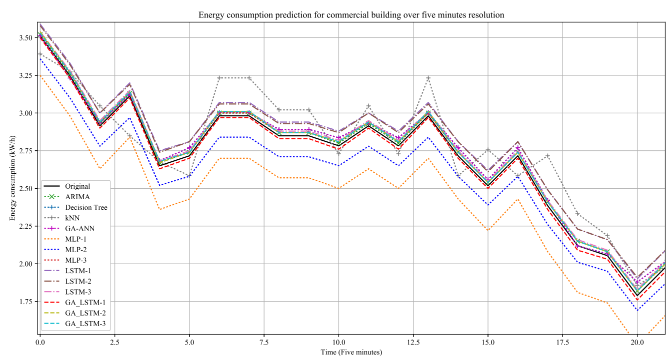


Figure 2.7: Prediction comparison between the proposed model with different conventional prediction models for very short term prediction.

Optimization Results Discussions

Hybridizing LSTM with GA produced more accurate prediction as seen from the tables and figures above. As the NP problem, it was not easy to find the best window size and number of hidden neurons in each layer because of the suitable combination of these parameters in each layer is a huge probabilistic task.

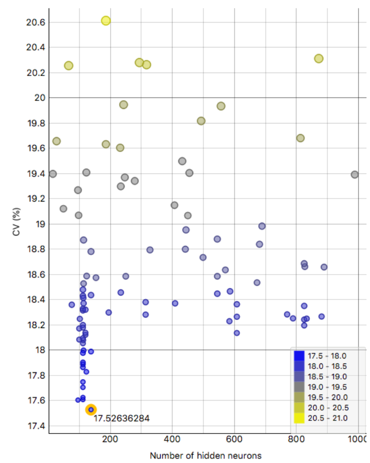
Fig. 2.8 (a) and (b) shows scatter plots of the best or survive offsprings in each generation at GA optimization problem of residential energy prediction, and comparisons between the number of hidden neurons and window size versus the CV score in percent. Fig. 2.8 (a) illustrates the performance of the GA-LSTM

Table 2.10: The 10-fold cross-validation results of GA-LSTM-1 for the second case study.

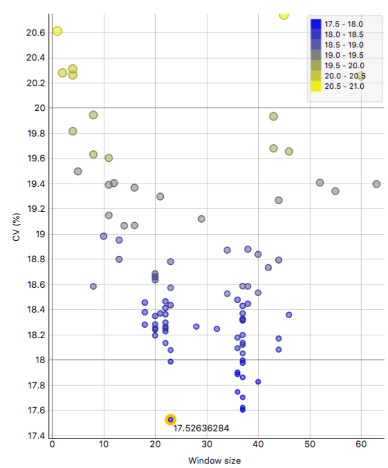
Fold No.	D_{tr}	D_{ts}	RMSE	CV (%)	MAE
1	9586	9582	0.199	3.859	0.147
2	19168	9582	0.194	3.765	0.111
3	28750	9582	0.384	7.456	0.217
4	38332	9582	0.460	8.940	0.251
5	47914	9582	0.401	7.785	0.251
6	57496	9582	0.647	12.570	0.373
7	67078	9582	0.763	14.806	0.431
8	76660	9582	0.617	11.985	0.354
9	86242	9582	0.357	6.935	0.221
10	95824	9582	0.291	5.653	0.198
Mean	-	-	**0.43**	**8.38**	**0.26**
SD	-	-	0.18	3.53	0.10

model while searching the best individual of hidden neurons which is 139 with 17.5% prediction accuracy. It is noticeable from the figure that the model converged with the number of neurons more than 100 and less than 150 neurons, however, the larger number failed to produce precise predictions. Similarly, Fig. 2.8 (b) presents the searching process of the proposed model to find best window size which is 23-time lags. From the figure, we can see that between 20 to 40 time lags the model performed the best results in comparison with smaller and larger time lags. Therefore, the GA-LSTM model converged to optimum results in the range of (100-150) neurons and the window size in the range of (20-40) time lags.

The scatter plots of the second case study in the commercial building are given in Fig. 2.9 (a) and (b). The scatter plot of the number of neurons versus the CV in Fig 2.9 (a) has a wider distribution than previous scatter plot of neurons in the residential building. There are a couple of local optimum individuals in the figure where the best offspring was 459 neurons with 8.3% prediction. Fig. 2.9 (b) shows



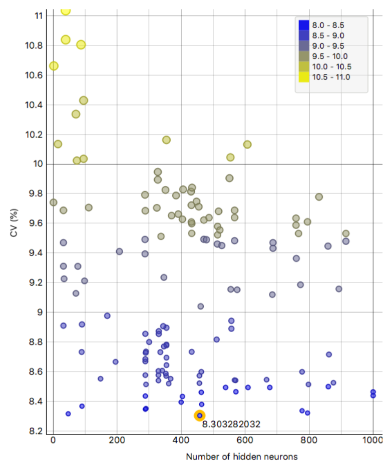
(a) Number of hidden neurons vs CV(%).



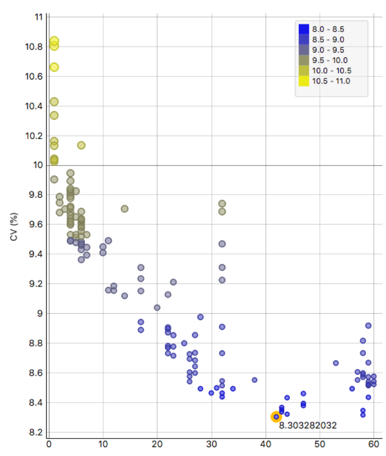
(b) Window size or time lags vs CV(%).

Figure 2.8: Scatter plots of window size and number of hidden neurons individuals in the GA optimization process for the residential energy prediction model.

the convergence results between 40 and 50-time lags where the smaller time lags are the worst prediction accuracy in the experiment. The best individual is 42 with CV 8.3%. Thus, the proposed model GA-LSTM led to optimum parameters of the number of hidden neurons and the window size in the commercial energy prediction.



(a) Number of hidden neurons vs CV(%).



(b) Window size or time lags vs CV(%).

Figure 2.9: Scatter plots of GA-LSTM optimization process for the commercial energy prediction model.

Chapter 3

A Complex System Approach for Smart Building Energy Consumption Prediction

3.1 Introduction

In modern buildings, advanced technology systems, such as building automation system, have been utilized to monitor real-time operations and enhance a tremendous amount of operational data to the building operators. However, there is a lack of integrated methods to handle that high operational information where the conventional methods with current building data analysis are neither effective nor efficient in predicting useful information from the massive data. Generally, it is challenging to predict building energy consumption precisely due to many influential factors correlated to energy consuming. Environmental parameters have high impacts on a building's electrical load, e.g., outdoor temperature, humidity, the day of the week, special events, etc.. Although environmental parameters are useful resources for energy consumption prediction, prediction using a large number of building's operational parameters, such as room temperature, major appliances and heating, ventilation, and air-conditioning (HVAC) system parameters, is a quite complicated problem, compared with prediction using only historical data. This huge challenge requires the building field to tackle the energy consumption prediction using the operational parameters.

The accurate prediction of energy usage at a specific time under many outside and inside conditions becomes the essential step. Still, with the aging and inappropriate use of the HVAC system and lighting system, the actual energy usage becomes unpredictable and infeasible for most of the existing energy estimation methodologies. Likewise, even if there are two buildings with the same building structure, the same HVAC and lighting system and the same environmental situation, the total energy consumption could be different from the two buildings. However, with the emerging of ACP theory, which consists of artificial societies, computational experiments, and parallel execution, gives us the practical architecture to solve the above problem. Combined with ACP theory, we can collect buildings' operational parameters using data acquisition and continuously train and update energy consumption prediction model using DL methodologies. Utilizing the ACP theory for energy consumption prediction ensures that the model is up to date with the current system condition and can predict precisely.

To ensure energy efficiency in a complex system such as buildings, planning and implementing precision energy consumption model and control measures are vital tools. Setting the optimal control measure requires associating all operational variables to energy consumption. Modeling an interactive, predictive model coupled with ACP concept provides a feasible solution of analysis and accurate prediction for precision energy consumption and control. ACP framework consists of three systems: (1) the physical system, which comprises the HVAC system, lighting system, and sensors, etc.; (2) the artificial system, which includes the data preprocessing and energy prediction model; (3) and communication system between physical and artificial system form the complex system. We implement the theory and framework to design the complex system and transfer the separated system into a correlated system, which can be executed by building managers to predict energy consumption of the specific building precisely under various control strategies and settings.

The primary objective of this research is to present a new approach to model a smart building's energy consumption using operational parameters. These parameters are MTS that include operational parameters and outside environment and used to model energy consumption by applying DL methods. Combined with ACP theory, this will provide new opportunities for analyzing building energy consumption, energy efficiency, and precision building control. The outcome of this research is an MTS predictive method embedded in ACP framework that can be applied to many other smart environment problems such as smart offices, smart homes, etc..

The modeling techniques presented in this chapter are based on hybrid DL algorithms. This chapter investigates the hybridization of the LSTM algorithm and the GRU algorithm using MTS inputs for supervised learning prediction. This case study seeks to predict the energy consumption in Daniel's College of Business (DCB) building at the University of Denver. To validate the proposed method, this investigation uses a real dataset from the DCB building that includes numerous environmental and operational parameters such as load profiles, outdoor temperatures, classrooms temperatures, and HVAC system parameters, etc., with a five-minute sampling interval. The model is designed for short-term load prediction which is a one-step-ahead prediction. The obtained results are compared with conventional prediction models by using traditional evaluation metrics.

3.2 Problem Formulation

The problem of predicting energy consumption using operational variables is an MTS processing problem. The MTS analysis technique is used for modeling and resolving important parameters sets. In this research, we solve the MTS challenge by implementing a hybrid DL approach combined with ACP theory to solve complex systems. Fig. 3.1 shows the technical scheme of the ACP theory.

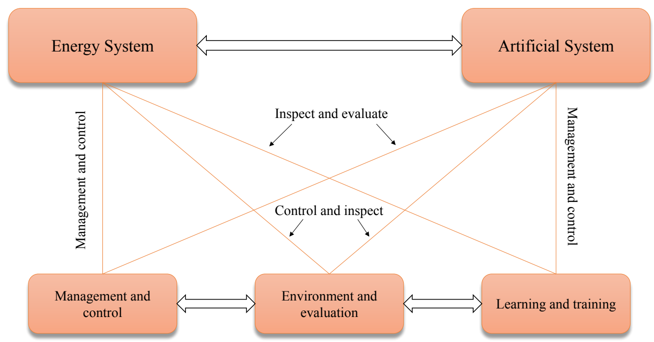


Figure 3.1: The ACP approach.

According to the ACP approach, to train the preliminary model for a complex system such as a smart building, the first step is that the sensors in the smart building will collect MTS variables and store enough data from the physical system as in Fig. 3.1, which is the energy system of the DCB building in this research. The stored information may include various types of data such as energy usage, humidity, temperature, weather conditions, room temperatures, and HVAC parameters, etc. Then, the collected data will serve as an input and training data in the artificial system. Once the model in the artificial system stage is trained, the model can predict energy usage under different control strategies and environmental factors. Therefore, building managers can refer to the trained artificial model to envision their future strategies. In addition, the artificial model may enhance the building's management to develop more efficient controlling strategies. The ACP approach ensures that the model in the artificial system is inspected and evaluated by the current data in the energy system. If an inaccurate prediction is detected, the ACP approach will retrain the model with a new dataset which has the status of variables and update the previous model. This mechanism guarantees that the artificial system is always up to date corresponding to the physical system.

Coupling the physical system with the artificial system provides a feasible means of parallel computations and intelligent systems. The physical system collects the operational variables of the building. Hence, the MTS vector collected from M-dimensional operational parameters and sensors can be defined as:

$$X = \{x_{i,m} = (x_{i,1}, x_{i,2}, \ldots, x_{i,M})\colon i \in \{1, 2, ..., N\}\} \tag{3.2.1}$$

where $x_{i,m} \in \mathbb{R}^M$ denotes the multivariable observations at a particular time step, e.g., $x_{n,m}$ is the value of the mth variable at time step n. The operational parameters observations will be transferred to the artificial system through the communication system. The artificial system consists of two major parts: data preprocessing and hybrid DL predictive model. The first step of the data preprocessing is normalizing the dataset features to eliminate the large deviation of instances, map the data vector X to a small ranges vector X', and help the learning algorithm to perform accurately. The value scale of the normalized input data is in the range $[0, 1]$ and can be defined as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.2.2}$$

where X is the MTS vector in the original values of the input dataset, X' is the normalized value scaled vector, X_{max} and X_{min} indicate the maximum value and minimum value of the features in X, respectively. The second step of the data preprocessing is preparing the dataset to supervised learning using a lag or sliding window method. The proposed hybrid DL model is trained and tested on supervised learning for the predictive model, and the prediction function is described as:

$$y'_{i+1} = h(x'_{i,m}, x'_{i-1,m}, \dots, x'_{i-w+1,m}) \tag{3.2.3}$$

where y' is the normalized energy consumption prediction value, h is the predictor function of the supervised learning, and w is the window size. In our experiment, the window size is one; therefore, the prediction function for the next step ahead is defined as:

$$y'_{i+1} = h(x'_{i,m}) \tag{3.2.4}$$

The third step of the data preprocessing is splitting the MTS operational parameters into training and testing sets. Let the training set be X_{tr} and the testing set be X_{ts} where $tr \in \{1, 2, \dots, p\}$ and $ts \in \{p+1, p+2 \dots, m\}$. To evaluate the predictive model properly, the training set is 70% of the total collected time observations m and the testing set is the last 30% of the observations.

The second part of the artificial system is our proposed predictive model which consists of encoder and decoder. The encoder is the LSTM model which is modeled to learn across sequential input datasets and extract the features. The decoder is the GRU model which learns from sequentially extracted features and predicts the output. When the predictive model trained all the training dataset X_{tr}, the testing dataset X_{ts} will be fed to the predictive model for prediction and testing the predictive model. The prediction accuracies are evaluated with a conventional evaluation metric. The technical details of our proposed hybrid DL method are presented in Section IV. The objective function of the proposed predictive model is expressed as:

$$\arg\min_{x,y} \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2 \quad \forall x \in X, \quad \forall y \in Y \tag{3.2.5}$$

where x_i is the actual energy consumption, y_i is the predicted energy consumption, and N is the total number of observations.

3.3 Datasets and Modeling Setup

3.3.1 Smart Building Data Description

The DCB building at the University of Denver was employed in this case study. The DCB is a six-story building with a total floor area of 110,536 square feet. All six floors of the building were instrumented with various sensors devices and data loggers. Different types of data were collected from the building: energy consumption, outdoor temperature conditions, indoor temperature data for each classroom or meeting room, and HVAC system parameters. The HVAC system in the DCB building consists of two air handling units (AHU), a chilled water system (CWS), a glycol fan coil system (GFCS), a hot water system (HWS) and a snowmelt system (SMS). The AHU systems include supply air temperature, return air temperature, and mixed air temperature. The CWS includes chilled water supply temperature and chilled water return temperature. The GFCS includes glycol water supply temperature and glycol water return temperature. The HWS includes hot water supply temperature and hot water return temperature. The SMS includes snowmelt supply temperature and snowmelt return temperature. Table 3.1 shows the types and number of operational variables collected from the DCB building. The total number of parameters used for energy prediction is 147, and they were collected with a five-minute interval for nine months. The MTS collected data is from February 2018 to October 2018. The data collected includes four different seasons (Winter, Spring, Summer, and Fall). Fig. 3.2 is the box plot that shows an example of eight operational parameters. The figure demonstrates the variation of the operational parameters in the building during the collected dataset. Fig. 3.3 shows the aggregate power consumption collected with five minutes resolution in the DCB for nine months.

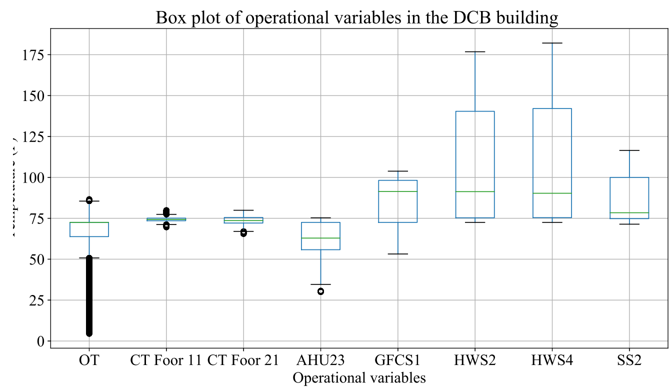


Figure 3.2: An example eight operational parameters in the DCB building.

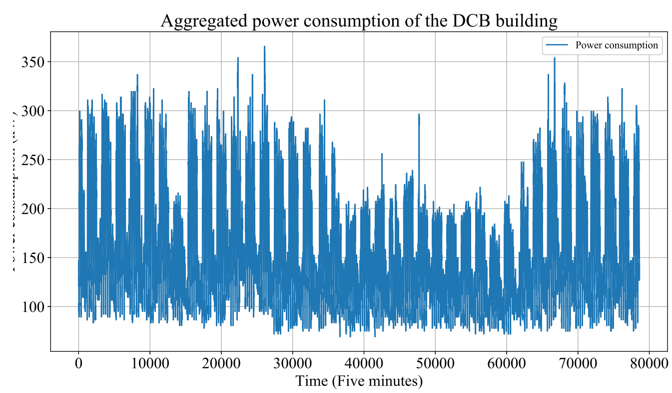


Figure 3.3: The aggregated power consumption of the DCB building.

3.3.2 Hybrid DL approach

Fig. 3.4 shows the technical scheme of the proposed hybrid DL approach which is part of the artificial system in the ACP scheme. The collected data from the DCB will be transferred to the artificial system which consists of the data preprocessing and the predictive model. There are three main steps in the preprocessing segment where the first depends on normalizing the original dataset, the second is preparing

Table 3.1: The type and number of the collected operational variables in the DCB building.

Type	Number	Unit
Outdoor temperature (OT)	1	^oF
Classrooms temperature (CT)	125	^oF
Chilled Water System (CWS)	5	^oF
Air Handling Unit (AHU) 1	4	^oF
Air Handling Unit (AHU) 2	4	^oF
Glycol Fan Coil System (GFCS)	2	^oF
Hot Water System (HWS)	3	^oF
Snowmelt System (SMS)	2	^oF
Aggregated power consumption	1	kW
Total number of the MTS operational variables	147	-

the input data to supervised learning form, and the third is splitting the normalized supervised dataset into two parts, the training set, and the testing set. To evaluate the performance of the proposed predictive model in the ACP theory properly, the testing set examines the trained model independently.

The predictive model is based on a coder and decoder which are the LSTM model and GRU model, respectively. The input features to the LSTM-GRU model are the record of operational parameters in the DCB building after the preprocessing analysis, and the output is the power consumption prediction for the step ahead which is the next 5 minutes in our experiment. It is unlike traditional LSTM or GRU models by hybridizing these two methods to improve the learning process. The first half is LSTM, which is utilized to extract the input features and encode them, and the second half is the GRU, which is used to analyze the extracted features from the LSTM and decodes them for energy consumption prediction. The process of feature extraction in the LSTM can sequentially capture the dynamics of the MTS dataset and determine complex time series events. The approach includes two layers of the LSTM to improve extracting the input features, and two layers of the GRU to analyze the collected extracted features and predict the output as shown in Fig. 3.4.

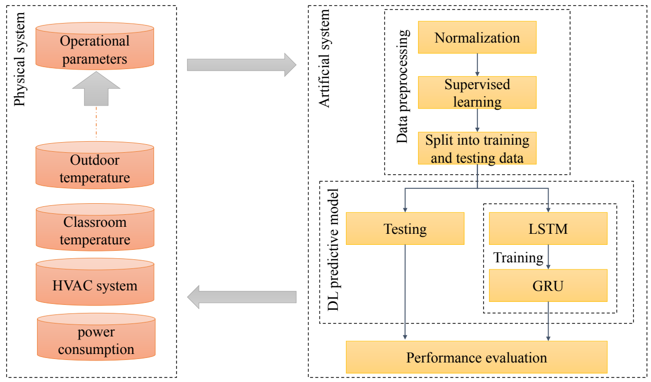


Figure 3.4: Hybrid DL predictive model scheme.

Commonly, the training process is applied with the BP algorithm to calculate the loss function and the gradient weights. The activation function used in our model is Rectified Linear Unit (ReLU) which is known as a ramp function and is widely applied for DL models in [30] [83] as:

$$ReLU(x) = \max(0, x) \tag{3.3.1}$$

where x is the input to a neuron. The total number of training epochs, which is a full pass of training through all the training dataset, is 300 epochs. The applied optimizer function is stochastic gradient descent, and the applied loss function is the mean square error.

3.4 Prediction Results

In this section, we utilized different evaluation criteria that exhibit the superiority of our proposed model. To test the performance comprehensively, the training data is set to seven months from February to August, and the testing data is two months from September to October. If the performance is not good enough, the system will retrain the hybrid DL model for the aforementioned ACP scheme. The first criterion discusses the enhancement of utilizing all MTS operational parameters to improve the energy consumption prediction in comparison with using only some of the operational parameters for training and testing. The second criterion is to compare our proposed predictive model with conventional prediction methods including MR, kNN, DT, MLP, LSTM, and GRU. The third criterion, we explore the advantage of the hybrid DL combined with the ACP theory compared with only hybrid DL prediction.

3.4.1 Different Operational Parameters

Considering different control operations for short-term use and high consumption, the system is modeled to predict energy consumption incorporating all operational parameters. To examine the enhancement of the use of operational parameters to our proposed predictive model, we compared the proposed model using all MTS parameters and using some of the operational parameters. This study, which conducts 5-minute-ahead prediction, explores the improvement of energy consumption prediction in the DCB using different types of operational parameters. According to the results in Table 3.2, the prediction performance enhanced with incorporating all MTS parameters. All MTS parameters have the best performance, and the AHU has the largest prediction errors. The HWS and CWS prediction have better performance than the AHU that has only four parameters. The results demonstrate that increasing the number of operational parameters enhances the prediction accuracy of the proposed model.

As shown in Fig. 3.5, the energy consumption prediction results of the MST parameters are shown in orange curves with triangles, and the original data is shown

Table 3.2: Performance of Different MTS Operational Parameters

Parameters	RMSE (kW)	CV (%)	MAE (kW)	MAPE (%)
AHU	7.565	4.700	5.690	3.876
CWS & HWS	7.515	4.669	5.616	3.763
All MTS	7.126	4.427	5.298	3.619

in blue curves. It is worth noting that the prediction for all MTS parameters is almost consistent with the original data and follow the original curves. The figure indicates the effectiveness of the proposed prediction approach.

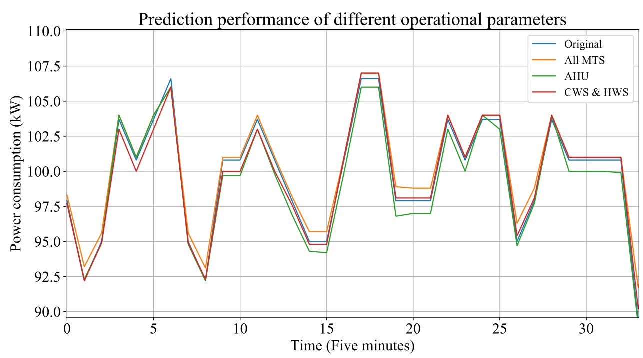


Figure 3.5: The prediction results of the compared prediction methods.

3.4.2 Compared with Conventional Prediction Methods

In [24] - [42], MR, kNN, DT, MLP, LSTM and GRU based multivariate and univariate predictions are presented. These methods were utilized in this research for comparison between the conventional models and the proposed model. The performances of the conducted 5-minutes-ahead predictions are shown in Table 3.3. According to the results in the table, the proposed approach has the best performance, and the DT has the largest prediction error. The traditional implementation of the GRU performed better than the traditional implementation of the LSTM. The results demonstrate the effectiveness of the proposed approach.

Table 3.3: Performance of different conventional prediction methods

Method	RMSE (kW)	CV (%)	MAE (kW)	MAPE (%)
MR	6.845	9.138	5.086	3.574
kNN	12.244	16.346	9.322	6.441
DT	12.603	16.825	8.904	6.266
MLP	9.971	6.195	7.825	5.154
LSTM	8.701	5.406	6.674	4.835
GRU	7.638	4.745	5.789	4.118
Proposed	7.126	4.427	5.298	3.619

Fig. 3.6 shows the energy consumption prediction of the proposed approach compared with different conventional methods. The proposed model has the best performance and follows the original curves of the energy consumption data. The figure indicates that the proposed approach is efficient and robust for energy consumption prediction.

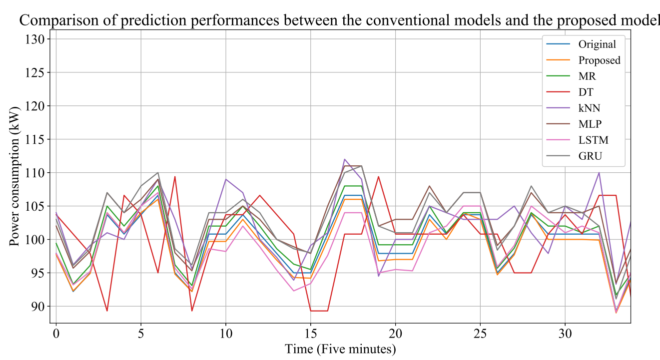


Figure 3.6: The prediction results of the compared prediction methods.

3.4.3 Hybrid DL Combined with the ACP Theory

The effectiveness of the strategy proposed in this research has been confirmed on combing the hybrid DL predictive model with the ACP using all operational parameters. The use of the real dataset allows us to compare the proposed approach with using only hybrid DL predictive model for training and testing. The reasoning behind the use of the ACP concept is that for hybrid DL predictive model, it can update and retrain the model if the collected data is corrupted while collecting the dataset and the prediction is inaccurate. Therefore, we compared the prediction performances of the proposed approach and the only hybrid DL model using all MTS parameters. The models were conducted for one month of training in February and tested for the last eight days in the month. The proposed approach was trained with the original values, and the only hybrid DL model was trained with the original values except for one day of corrupted data in the tenth of the month. According to the results in Table 3.4, the proposed approach outperformed the only hybrid DL model. The results demonstrate that the proposed approach has accurate and robust prediction performance. Fig. 3.7 shows the prediction performance of the proposed approach and hybrid DL model only. The results indicate that the proposed approach is a skillful predictive framework.

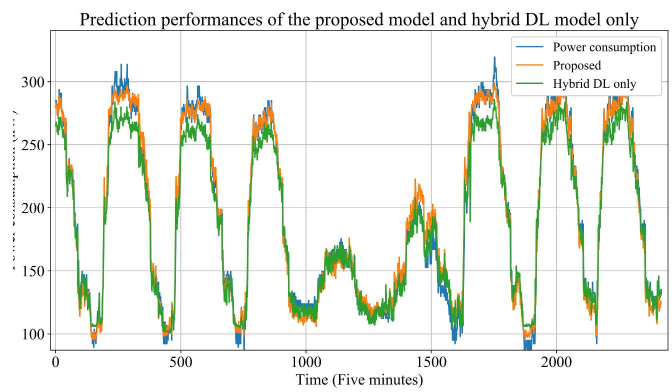


Figure 3.7: The prediction results of the hybrid DL combined with ACP and hybrid DL only.

Table 3.4: Performance of The ACP with hybrid DL and hybrid DL only

Parameters	RMSE (kW)	CV (%)	MAE (kW)	MAPE (%)
Hybrid DL only	10.980	6.329	8.852	4.880
Proposed	10.344	5.581	7.925	4.797

Chapter 4

Parallel Power Consumption Forecasting for Buildings Based on Hybrid Deep Learning and Big Data

4.1 Introduction

Enhancing the STLF model with the Artificial intelligence (AI) to predict the power consumption provides real-time monitoring of power consumption, an accurate forecast, and a precision decision making when training big data and parallel buildings. Since most of the existing research of the STLF use small historical datasets of the electric consumption load and small computational capabilities, the forecasting accuracies need some improvements using bigger datasets and large computational frameworks. As a promising solution, the emerge of artificial societies, computational experiments, and parallel executions (ACP) theory [84], gives us the functional architecture to solve the above problem with problem modeling through the artificial societies, analyzing through computational experiments, and controlling through parallel executions [84]. Combining the forecasting model with ACP theory, we can implement a parallel predictive model to continuously train and update the energy prediction model and ensure that the model is up to date with the current system condition and can forecast precisely.

The research problem in this work is summed up with the use of recent DL methods and hybridize them to forecast the power consumption for the STLF. To be more specific, mixing two successful DL methods can produce a precision forecasting that is close to the original value. Thus, applying the hybridized DL methods to the power consumption problem can contribute to the improvement of power efficiency and grid reliability with obtained precision forecasting. Moreover, emerging the hybrid forecasting model with ACP concept can provide the primary paradigm for parallel big data forecasting. Consequently, the proposed framework can help utility providers and building managers to enhance their power consumption performances.

To the best of the authors' knowledge, the previous conventional studies of load forecasting focused on predicting energy consumption by using small historical datasets and applying offline training and testing without updating the models with the current states. However, there are still questions regarding hybridizing two significant DL methods to improve forecasting accuracies, using big data for training the forecasting models and formulating the forecasting model for online prediction. In this research, we try to solve the following open questions. How should a robust power consumption forecasting be modeled with hybrid DL methods for STLF? How to modify and update the model continuously with parallel computation? And how accurate is the corresponding forecasting using big datasets and different timescales resolutions such as (minutes, hours, days, weekdays, and seasons)?

The main motivation of this research is to create a new paradigm, that power consumption can be modeled by a parallel forecasting model, and using big dataset, and thus provides the feasibility of real-time forecasting and large-scale complex systems. And the primary objective is to present a new modeling approach to power consumption forecasting by hybridizing CNN and GRU methods coupled with ACP. This framework will provide new opportunities for analyzing power consumption, energy efficiency, and precision energy control. The outcome of this research can be applied to many other complex system problems such as smart offices, smart homes, smart campuses etc. Our major contributions include:

1. A hybridization technique to improve the performance of the DL forecasting model.
2. A new paradigm of online forecasting model using ACP concept and parallelizing computation.
3. Understanding forecasting factors that are correlated to the power consumption by investigating big data.

The modeling techniques presented in this research are based ACP theory for parallel computing and results updating and CNN-GRU for model forecasting in different timescales (5-minute ahead, 30-minute-ahead, 1-hour-ahead, 12-hour-ahead, 1-day-ahead, and 3-day-ahead). Moreover, the study investigates the forecasting accuracies for seasonality and events forecasting in comparison with the big cumulative dataset for training. This case study seeks to predict the power consumption in five buildings at the University of Denver. The proposed model is compared with conventional forecasting models. The performance of the models is evaluated with normalized root mean square error.

4.2 Problem Formulation

In this research, the presented approach of power consumption forecasting is to estimate the power consumption load using historical power consumption data for STLF. The proposed forecasting framework is coupled with ACP concept for parallelizing computation and real-time estimation. The ACP is connected with the artificial system of the forecasting model as in Fig. 3.1.

The physical system in the ACP concept is the energy system in our problem, where smart meters collect power consumption data and store them. These collected data will serve as an input to the artificial system, which is the hybrid forecasting model, for the training process. Once the artificial system is trained, the model can forecast power consumption in different timescales and events. The participants can refer to the trained artificial system for future control strategies. The ACP approach ensures that the model in the artificial system is updated with the current data. For instance, if the artificial system forecasts inaccurately, the ACP approach will start over to retrain the model with the updated dataset. This mechanism provides an up to date corresponding data to the physical system.

Coupling the physical system with the artificial system provides a feasible means of parallel computations and intelligent systems. Fig. 4.1 represents the flowchart scheme of the proposed model including the physical system and artificial system. The physical system as in the flowchart collects the operational variables of the building. The artificial system consists of two main parts: data preprocessing and hybrid DL predictive model. To achieve the most accurate prediction, the hybridization is designed to code and decode the MTS input features.

The flowchart showed in Fig. 4.1 represents the proposed model for the STLF. The first part in the diagram is the data preprocessing, which has two main steps, of the collected power consumption data. In the first step, the load data is preprocessed

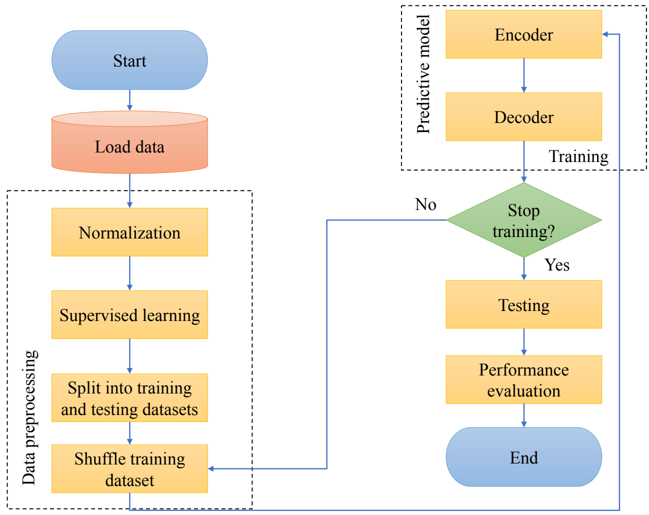


Figure 4.1: Flowchart of the proposed CNN-GRU model for the STLF.

with normalization method as:

$$X_i' = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{4.2.1}$$

where X_i is the original value of the input dataset, X_i' is the normalized value scaled to the range $[0, 1]$, X_{max} and X_{min} indicate the maximum value and minimum value of the features in $\mathbf{X}$, respectively. Normalizing the dataset features eliminates the large deviation of instances, maps the data vector $\mathbf{X}$ to a small ranges vector $\mathbf{X'}$, and helps the learning algorithm to perform accurately. Let m be a sequence of historical data and $X_i \in \mathbb{R}$; hence, the power consumption vector is as follows:

$$\mathbf{X} = \{X_0, X_1, \ldots, X_{m-1}\} \tag{4.2.2}$$

where X_i is the historical power consumption at a time i and $i \in \{1, 2, ..., m\}$. The mapped vector with normalized values is defined as follows:

$$\mathbf{X'} = \{X_0', X_1', \ldots, X_{m-1}'\} \tag{4.2.3}$$

In the second step of the data preprocessing is preparing the dataset to supervised learning. The proposed hybrid DL model is tested on supervised learning for the forecasting output as:

$$\mathbf{Y}' = f_s(\mathbf{X}') \tag{4.2.4}$$

where $\mathbf{Y}'$ is the normalized predicted power consumption of the time series, f_s is the supervised learning function, and the normalized output vector is defined as follows:

$$\mathbf{Y}' = \{Y'_m, Y'_{m+1}, \ldots, Y'_{m+n-1}\} \tag{4.2.5}$$

where Y'_{i_n} is the mapped predicted power consumption at a time i_n and $i_n \in \{m+1, m+2, ..., m+n\}$, n is the number of future time forecasting elements in the set $\mathbf{Y}'$. With the inverse function of the normalization preprocessing, the result $\mathbf{Y}$ is computed for the original scale values of the power consumption forecasting.

The objective function of the proposed predictive model is expressed as:

$$\underset{X,Y}{\arg\min} \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_{i_n})^2} \quad \forall X \in \mathbf{X}, \quad \forall Y \in \mathbf{Y} \tag{4.2.6}$$

The second main part of the flowchart is the predictive model which consists of encoder and decoder. The encoder is the CNN model which is modeled for one-dimensional convolutional layers to learn across sequential input datasets and extract the features. The decoder is the GRU model which learns from sequentially extracted features and predicts the output. The predictive model will go back to shuffle the training dataset for further learning until the training dataset is over. When the predictive model trained all the training dataset, the testing dataset will be served to the predictive model for forecasting and testing the predictive model. The prediction output will be evaluated with a conventional evaluation metric.

4.3 Modeling Setup

4.3.1 Modeling hybrid DL approach

Referring to the scheme of the proposed approach which is shown in Fig. 4.1, the model stands for two main parts including preprocessing and predictive model. As demonstrated in the problem formulation section, the data preprocessing segment is the first part to prepare the input features collected from the physical system in the ACP concept to the artificial system which is our proposed predictive model.

There are three main steps in the preprocessing segment where the first depends on normalizing the original dataset as in Eq. (4.2.1), the second is preparing the input data to supervised learning form as in Eq. (4.2.4), and the third is splitting the normalized supervised dataset into two parts, the training data, and the testing data. To evaluate the performance of the proposed model accurately, the training data is used for the training process of the approach, and it shuffles during the training process of the predictive model, and the testing data is used just for testing the forecasting process.

The predictive model is based on a coder and decoder which are the CNN model and GRU model, respectively. The input of the CNN-GRU is the record of power consumption dataset of a building after the preprocessing analysis, and the output is the power consumption forecasting for the next 5 minutes, next hour and the next day. It is unlike traditional CNN or GRU models by hybridizing these two methods to improve the learning process. The first half is CNN, which is utilized to extract the input features and encode them, and the second half is the GRU, which is used to analyze the extracted features from the CNN and decodes the features to predict the power consumption for the next period of time. The approach includes two layers of the one-dimensional CNN to improve extracting the input features, one layer of the one-dimensional pooling to collect the extracted features, and two layers of the GRU to analyze the collected extracted features and predict the output as shown in Fig. 4.2.

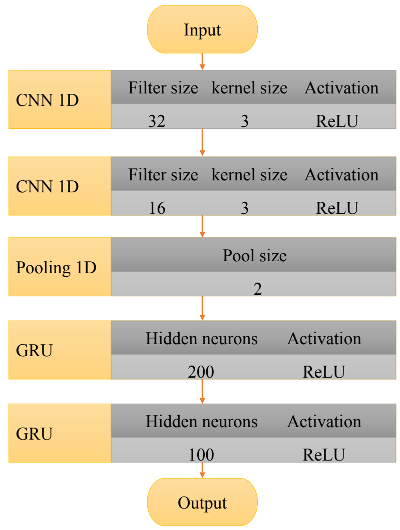


Figure 4.2: GRU block with reset gate and update gate.

Usually, the training process is applied with the BP algorithm to calculate the loss function and the gradient weights. The activation function used in our model is Rectified Linear Unit (ReLU), as in (12), which is known as a ramp function and is widely applied for DL models in [85, 83]. The total number of training epochs, which is a full pass of training through all the training dataset, is 300 epochs. The applied optimizer function is stochastic gradient descent, and the applied loss function is the mean square error.

4.3.2 Datasets

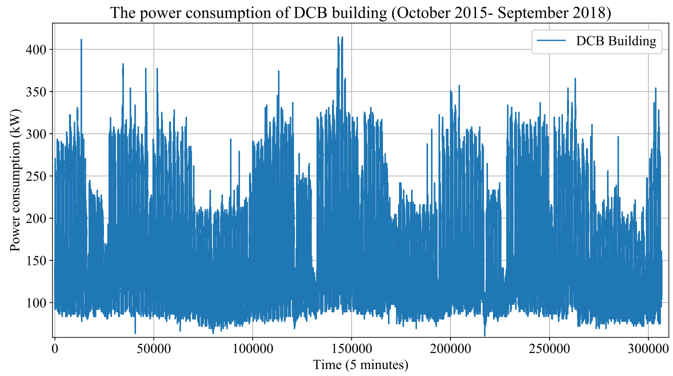
In this study, we provide a case study to demonstrate the proposed hybrid CNN-GRU for parallel distribution power consumption forecasting. From Fig.3.1 and Fig. 4.1, utilizing the parallel computation and using the technical scheme of the concept of ACP provides a feasible solution to achieve the objective of this study. The detailed technical description of the proposed predictive method was introduced last this section. The case study proposes the strategy of smart buildings power consumption interact in the University of Denver (DU) campus power grid. Five buildings from the DU campus are chosen for a demonstration of the proposed method including Daniel's College of Business (DCB), Newman Center for the Performing Arts (NCPA), Ricketson Law Building (RLB), Ritchie Wellness Center (RWC) and Sturm Hall Building (SHB). Table 4.1 provides detail information about these buildings in the campus.

Table 4.1: Five buildings details at the DU Campus.

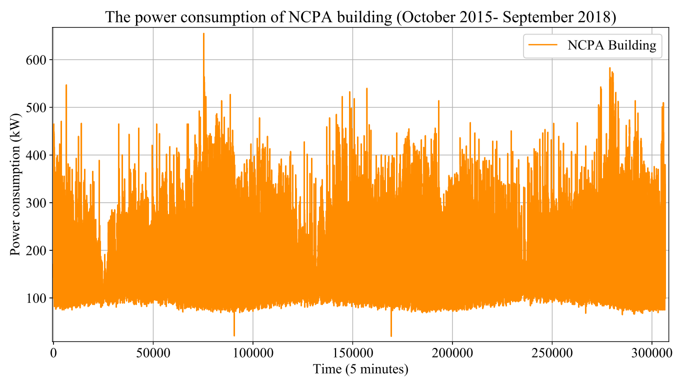
	Building type	Area (ft^2)	Floor
DCB	Academic	110536	6
NCPA	Performing arts	180000	5
RLB	Academic	181000	4
RWC	Fitness center	440000	4
SHB	Academic	245000	4

The buildings datasets consist of historical power consumption in kW from October 2015 to September 2018 with five-minute resolution, and the total number of attributes in each dataset is 315648 samples. Fig. 4.3 (a) and Fig. 4.3 (b) shows the line graphs of power consumption in DCB and NCPA buildings, respectively, and they illustrate the variations of consumptions during seasons and events during the three years of collected data.

Fig. 4.4 (c) and Fig. 4.4(d) display the heat maps of the RWC and SHB buildings, respectively, and illustrate the power consumptions over the daily and hourly consumption during the collected dataset. It is worth noting, the peak power consumption in the RWC building which is a wellness center for students is out of the normal peak ours in other buildings. Thus, the occupants of the wellness center they use the facility after working hours. However, the peak of the power consumption in the academic building SHB is within the normal peak hours.



(a) DCB building



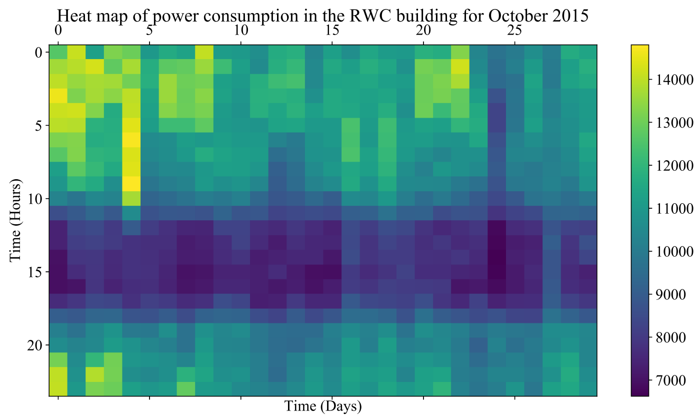
(b) NCPA building

Figure 4.3: The line graph of buildings power consumptions (DCB and NCPA) at the DU campus.

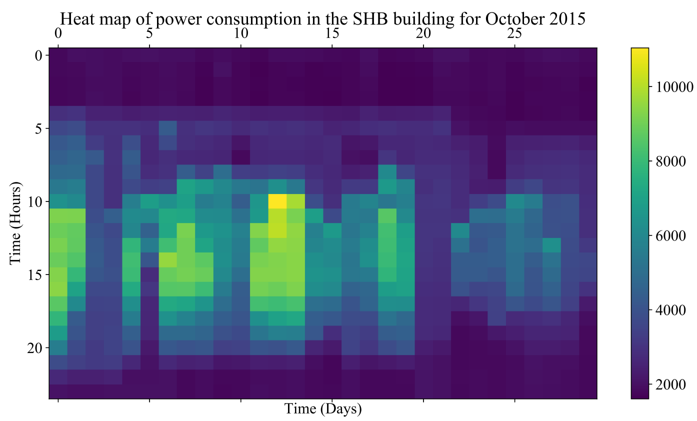
4.4 Prediction Results

The conducted datasets include three years of collected power consumption data with five-minute resolution for a couple of buildings at the DU campus, thus, they consist of twelve seasons which are three winter seasons (Dec. - Feb.), three spring seasons (Mar. - May.), three summer seasons (Jun. - Aug.), and three fall seasons (Sep. - Nov.).

To evaluate the forecasting performance results, we utilized 30% of the test dataset to examine the model after training with 70% of the dataset. The conven-



(a) RWC building



(b) SHB building

Figure 4.4: The heat map of buildings power consumptions (RWC and SHB) at the DU campus over October 2015.

tional metrics used to evaluate the predictive models are utilized to evaluate the forecasting in our experiments.

4.4.1 Numerical results and analysis

Different seasonal resolutions

Considering different time transients and short-term disturbances of seasonal effects, the proposed approach is conducted in different seasonal resolutions (winter, spring, fall, and summer). In our study, the forecasting of the seasonal resolutions is compared with a large-scale dataset that includes cumulative seasons datasets for each building to show the effect of the seasonal disturbances of the load consumption. In detail, the training dataset of each season is 70% of the historical power consumption with 5-minutes resolution. To evaluate the training model, the 30% dataset is employed to the CNN-GRU with sliding window for next 5-minutes ahead forecasting. For instance, first, the power consumption of seven months of summer seasons in DCB building is conducted as the training data to determine the best parameters and build the forecasting model. The remains two months are used as the test data to evaluate the model. However, the comparative study with big cumulative data is utilized to train twenty-six months of power consumption in the same building regardless of the seasons. Similarly, the remain ten months are used to test the forecasting model. This experiment examines the effect of big data utilized for training in predictive models.

Table 4.2: Performance of different seasons for 5-minute-ahead forecasting (NRMSE (%)).

	winter	spring	summer	fall	cumulative data
DCB	2.996	2.970	2.870	2.709	2.172
NCPA	5.987	5.942	6.845	6.510	5.533
RLB	1.613	1.841	1.666	1.945	1.560
RWC	2.175	2.289	2.929	2.093	1.954
SHB	3.737	4.748	4.306	4.478	3.307

Referring to the performance results demonstrated in Table 4.2, the 5-minute-ahead forecasting is examined for each building chosen in this study at the DU campus using NRMSE metric for performance evaluation. According to the presented results, different seasons forecasting and cumulative data forecasting are compared to demonstrate the ability to train big data. In general, the RLB has the best performance in comparing with other buildings. However, the NCPA building has the largest forecasting errors. It is worth noting, the CD forecasting in each building has performed the best among another seasonal forecasting experiments. The results confirm that big data training used in our hybrid CNN-GRU model is better than training for different seasonal resolution.

Different timescale resolutions

In different timescales forecasting, different time scales are considered for resolution effects such as 30-minute-ahead, 1-hour-ahead, 6-hour-ahead, and 1-day-ahead forecasting. Since the best performance of forecasting in Table 4.3 with 5-minute-ahead forecasting was cumulative data, the approach of different timescales is implemented on aggregate data in each building. The training data is 70% of the whole dataset for each timescale study, and the testing data is 30% of the dataset. The sliding window size depends on the timescale, for example, the sliding window of 1-hour-ahead forecasting is 1-hour size. It is worth noting, 5-minute-ahead forecasting has the best predictions in each building, and the performance worsens as the timescale resolutions increase. Moreover, the 1-day-ahead forecasting is better than the 6-hour-ahead forecasting in a couple of buildings because of the considerable variability of attributes in 6-hour resolution.

Table 4.3: Performance of the proposed method for different timescales ahead forecasting (NRMSE (%)).

	5-minute	30-minute	1-hour	6-hour	1-day
DCB	2.172	3.767	4.939	6.881	7.055
NCPA	5.533	5.998	5.933	6.440	6.474
RLB	1.560	2.644	3.287	5.212	4.728
RWC	1.954	3.154	3.485	4.702	4.065
SHB	3.307	9.850	10.098	11.470	10.652

Weekday power consumption forecasting

As shown Table 4.4, the weekday power consumption forecasting is investigated and compared with 1-day-ahead forecasting from the previous timescale resolution. Since most of these buildings are academic building and operates normally in the weekdays, the approach is conducted to investigate the effectiveness of training the model only on weekdays and excluding the weekends and national holidays. First, the weekdays power consumption values $\{X_0^{wd}, X_1^{wd}, \dots, X_{m-1}^{wd}\}$ are collected, where $wd = \{1, 2, 3, 4, 5\}$, which indicates Monday, Tuesday, Wednesday, Thursday and Friday. Then, the training set is 70% of the collected weekdays' data which is used to train the proposed method, and 30% is utilized to test the model. Table 4.4 shows that the weekdays forecasting did not improve the forecasting performance; therefore, the proposed hybrid DL model performs better with big data for training.

Table 4.4: Performance of the proposed method for weekdays forecasting (NRMSE (%)).

forecasting (data)	DCB	NCPA	RLB	RWC	SHB
1-day-ahead (weekday)	7.132	6.530	5.432	4.230	11.802
1-day-ahead (cumulative)	7.055	6.474	4.728	4.065	10.652

4.4.2 Compared with other methods

In Table 4.5, the performances of 5-minute-ahead forecasting is chosen from the different timescales forecasting. We utilize the NRMSE evaluation metric to benchmark the performance of our proposed model with the conventional forecasting methods proposed in the literature.

According to Table 4.5, our proposed model outperforms conventional models for power consumption forecasting in each building. It is worth noting; the ARIMA has performed the worst in a couple of buildings; however, the CNN performed better than other conventional models and then it is followed by the GRU model. It is worth notice, hybridizing these methods CNN and GRU improves the forecasting accuracy in each building experiment. The results prove the potential outperformance of our proposed predictive model which is hybridizing CNN with GRU.

Table 4.5: Performance of different methods for 5-minute-ahead forecasting (NRMSE (%)).

	ARIMA	MLP	LSTM	GRU	CNN	Proposed
DCB	4.121	4.084	3.849	3.600	3.471	2.172
NCPA	8.754	8.003	7.941	7.327	7.147	5.533
RLB	2.261	2.267	2.110	2.042	1.815	1.560
RWC	3.041	2.714	2.583	2.564	2.353	1.954
SHB	6.131	5.903	5.938	5.824	5.699	3.307

As shown Fig. 4.5 and Fig. 4.6, the forecasting result is shown in orange curves with triangles and the original data is shown with blue curves. It is worth noting, the forecasting curves are almost consistent with the with original curves except for several abrupt deviations points. This represents the effectiveness of the CNN-GRU forecasting method.

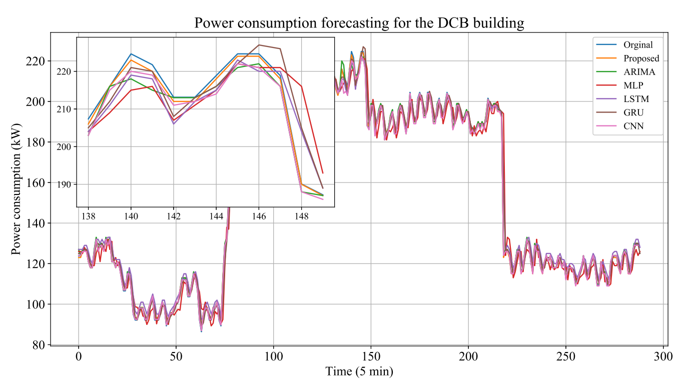


Figure 4.5: The performance of the power consumption forecasting in the DCB. The comparison between the proposed model and conventional models shows the outperformance of the proposed model.

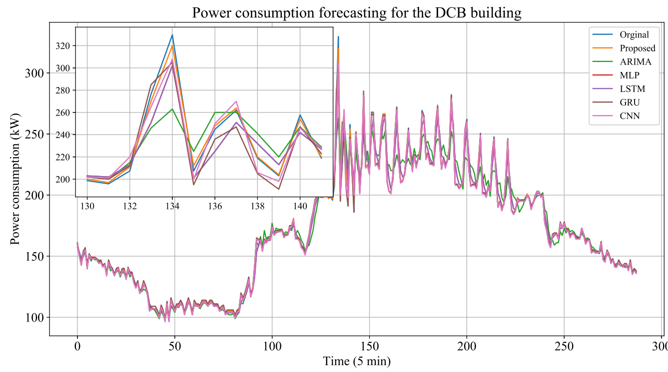


Figure 4.6: The performance of the power consumption forecasting in the NCPA building. The comparison between the proposed model and conventional models shows the outperformance of the proposed model.

4.4.3 Cross-validation and discussion

The time series cross-validation method, which splits the dataset sets into k-fold subsets to estimate the general performance of the forecasting model, gives an insight on how the model generalizes the independent variables throughout the datasets. The method repeats the process of splitting the dataset into training and testing portions for k-times where the size of the testing data remains fixed but moving through the original dataset and the remainder used as training dataset every fold.

Applying this method to the proposed model produces a robust averaged estimation of the forecasting when each observation in the dataset is used for training and testing at each fold. We utilized 10-fold cross-validation in our experiment for the best parameters of the proposed model in each case study of the residential and commercial buildings using time series cross-validator [86].

From the previous experiments of different seasonal resolutions, timescales and weekdays forecasting, we can notice that the performances of forecasting 5-minute-ahead using cumulative data are the best models for all buildings. Therefore, we implemented the cross-validation method for 5-minute-ahead forecasting for all buildings to sight the general models' performances. Fig. 4.7 shows the frequency distribution of the accuracy forecasting produced by the 10-k cross-validation method and Table 4.6 represents the resulted forecasting at each fold. These training datasets are shuffled along with the cross-validation process ten times. Thus, the mean and standard deviation (SD) of the total error forecasting are calculated. This technique increases the confidence of the proposed predictive model because the testing data is different and unseen during the training operation.

Table 4.6: The 10-fold cross-validation results of CNN-GRU for 5-minute-ahead forecasting (NRMSE (%)).

Fold No.	DCB	NCPA	RLB	RWC	SHB
1	2.768	4.272	1.504	2.116	4.552
2	2.931	4.588	1.561	1.957	3.643
3	2.767	5.464	1.589	2.315	3.293
4	2.696	5.607	1.780	2.119	4.113
5	2.728	5.371	1.606	2.121	3.760
6	2.729	5.442	1.633	2.166	4.565
7	2.857	5.415	1.570	2.156	4.220
8	2.759	5.389	1.562	2.017	4.296
9	2.711	5.281	1.564	2.111	4.749
10	2.670	5.947	1.536	2.175	4.292
Mean	**2.76**	**5.28**	**1.59**	**2.13**	**4.15**
SD	0.07	0.46	0.07	0.09	0.43

Comparing the average NRMSE for our proposed model with the standard prediction models obtained from the previous section will confirm the outperformance of our model. Referring to Table 4.5 and Table 4.6, the average error prediction of our proposed model is still better than all compared models for each building dataset. Fig. 4.8 represents the percentage of prediction improvement or the NRMSE reduction for each building dataset. Our proposed model improved forecasting accuracy in comparison with conventional methods. The highest improvement percentage is

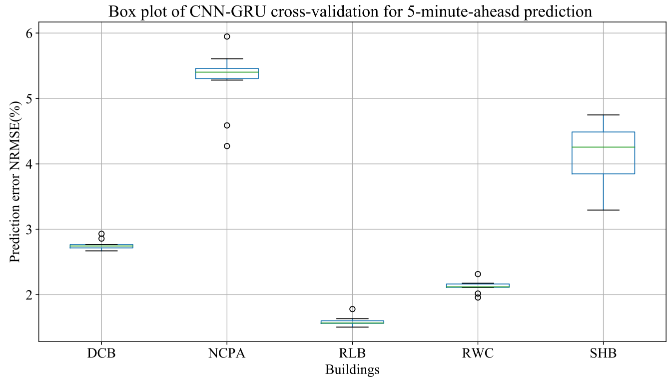


Figure 4.7: Box plot of NRMSE error prediction of cross-validation in each building.

almost 40% in comparison with ARIMA, and the lowest is about 9% in compare with CNN. These results confirm that our proposed model improved forecasting accuracy more than 9% and less than 40% in comparison with other forecasting methods. Therefore, we can conclude that the proposed method has gained a remarkable improved performance and percentage of reduction compared with conventional methods for all buildings in the study.

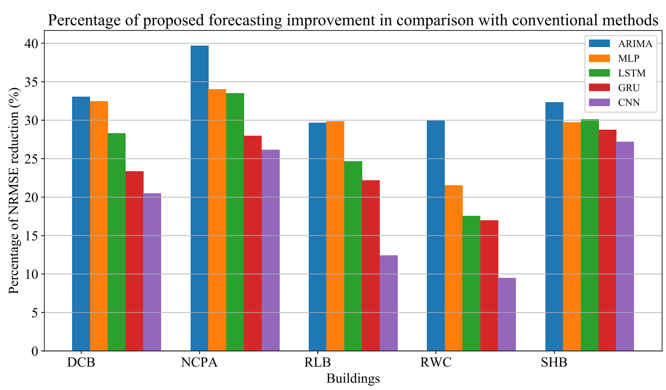


Figure 4.8: Percentage of NRMSE reduction in comparison with our proposed CNN-GRU model.

Chapter 5

Energy Disaggregation of Residential Prosumers

5.1 Introduction

Energy disaggregation is an efficient computational technique that is used for non-intrusive load monitoring (NILM) of individual loads and in particular residential appliances. It is unlike the more straightforward intrusive load monitoring (ILM) approach that is performed by placing sensors on appliance circuits. Indeed, the NILM is much cheaper than ILM, and it is an applicable method by preserving customers' privacy. In addition, The NILM is a sophisticated approach for utilities to provide customers with detail feedback that can provide feasible means of dynamic pricing and demand-side management. However, an existing challenge is that the available NILM methods cannot effectively capture localized generation which is an extremely pressing issue due to the growing penetration of behind-the-meter energy resources. Moreover, electric utilities have largely installed one smart meter for each customer to measure the net load, which masks the local generation.

A variety of NILM exists, including event-based methods which classify the change of steady-state and transient state consumptions [43], and nonevent-based methods that use temporal graphical models like Hidden Markov models [49]. Machine learning methods have also been used based on support vector machine [58], and deep learning [50], [69], [87]. Although the literature is extensive, lacking is an effective method to disaggregate residential loads when new types of distributed energy resources such as electric vehicle (EV) and solar photovoltaic (PV) are integrated. In this letter, an effective framework of energy disaggregation for a residential prosumer that comprises different electrical loads is proposed. The proposed method employs the data collected from smart meter and trains a hybrid deep learning model to classify and determine different electric loads and behind-the-meter generation.

5.2 Problem Formulation

Considering different appliance signatures in a residential building is an essential step to understand the customer behavior and load patterns. Consider the energy system in a modern residential building is integrated with PV and EV. The on/off state appliances are considered as type I, e.g., undimmable light bulbs. The multi-state appliances are considered as type II, e.g., washer and HVAC system. The PV and EV charging are considered as type III for continuously variable load.

In other words, the mathematical formulation using the real power of the net load, the aggregated load, solar generation and EV charging and discharging can be defined as:

$$P_t^n = P_t^a - P_t^s \pm P_t^e \tag{5.2.1}$$

where P_t^n is the measured net load by the smart meter, P_t^a is the aggregated prosumer's power, P_t^s is the behind-the-meter PV generation, P_t^e is the behind-the-

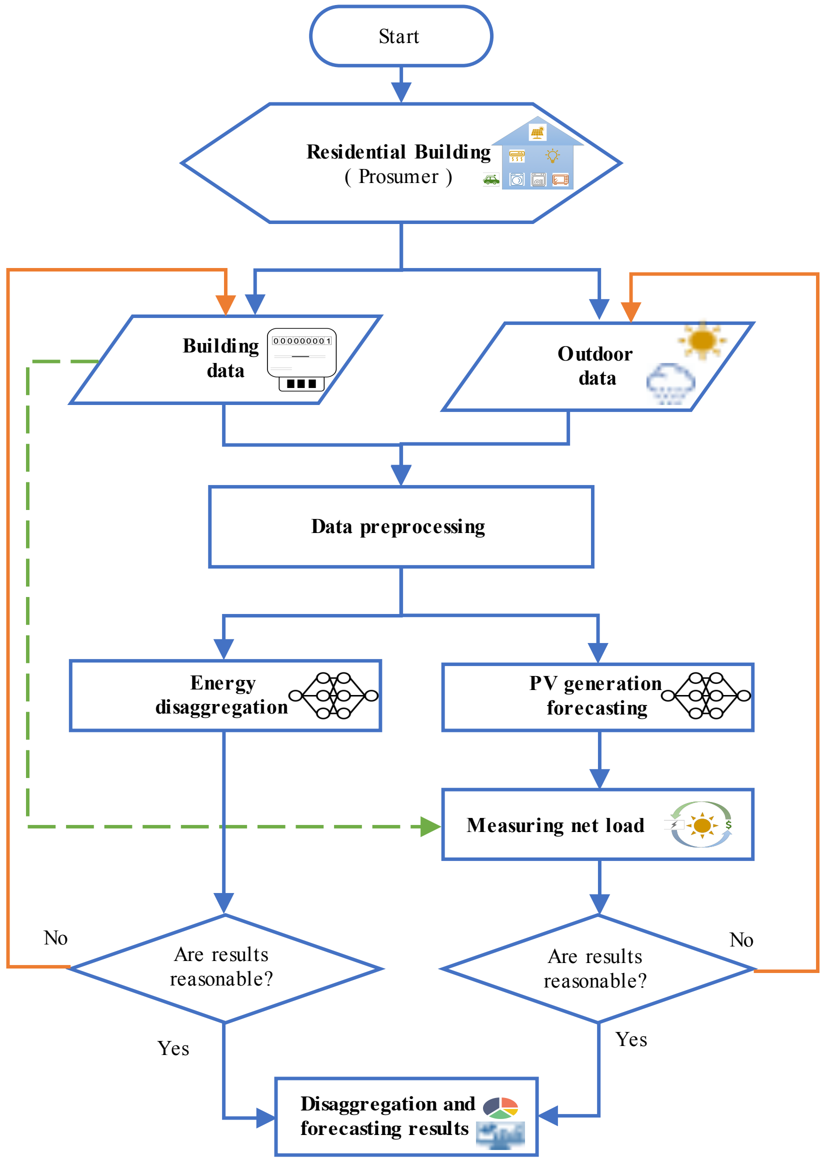


Figure 5.1: Flowchart of the energy disaggregation framework.

meter EV charging and discharging, and t is the index for time periods. The aggregated power is the summation of power usage of the individual appliances, including EV charging, and can be defined as follows

$$P_t^a = \sum_{i=1}^{n} p_{i,t}, \quad \forall p_i \in P^a \tag{5.2.2}$$

where $p_{i,t}$ is the power usage of individual appliance i at time t and n is the total number of appliances inside the building.

The proposed energy disaggregation framework is shown in Fig 5.1. The framework dataset comprises of weather data and building data. The preprocessing techniques are applied to the dataset by normalizing and transforming to supervised learning. The proposed disaggregation method consists of two main parts using a hybrid deep learning method that combines CNN model for sequential feature extraction and LSTM model for extracted feature analysis. The first part of the approach trains the model with 70% of the weather input data to forecast solar power, and the second half trains the model with 70% of the aggregated load data from the building to classify individual energy consumption sources. Once these two parts are trained, each model is tested with the last 30% of data accordingly.

Let T be a sequence of historical aggregated load data $P_t^a \in \mathbb{R}$,hence, the aggregated load vector can be defined as follows:

$$P_t^a = \{P_1^a, P_2^a \dots P_T^a\} \tag{5.2.3}$$

The proposed hybrid deep learning model is trained and tested on supervised learning for energy disaggregation and solar energy forecasting. The prediction output can be defined as follows:

$$\hat{P}_t = f_{sp}(P_t^a) \tag{5.2.4}$$

where f_{sp} is the supervised learning function, $\hat{P}_t$ is the predicted output for disaggregation and forecasting, and the prediction vector is defined as follows:

$$\hat{P}_t = \{\hat{P}_T, \hat{P}_{T+1} \dots, \hat{P}_{T+w-1}\} \tag{5.2.5}$$

where $\hat{P}_t$ the prediction vector which can be disaggregation prediction vector $\hat{P}_t^d$ and forecasting $\hat{P}_t^f$, and w is the supervised learning time window for disaggregation and forecasting elements. Once the PV forecasting vector $\hat{P}_t^f$ is predicted, measured net load vector is expressed as follows:

$$\hat{P}_t^{net} = P_t^n - \hat{P}_t^f \tag{5.2.6}$$

The objective function of the proposed hybrid deep learning model for disaggregation prediction is expressed as:

$$\arg\min \sqrt{\frac{1}{T}\sum_{t=1}^{T}(P_t^a - \hat{P}_t^d)^2} \tag{5.2.7}$$

The objective function of the proposed method for PV forecasting prediction is expressed as:

$$\arg\min \sqrt{\frac{1}{T}\sum_{t=1}^{T}(P_t^s - \hat{P}_t^f)^2} \tag{5.2.8}$$

5.3 Hybrid DL approach

The hybrid deep learning method consists of two learning steps, i.e., extracting sequence features from the input data with CNN model for one dimensional architecture and analyzing the extracted features with LSTM model.

The CNN operates somewhat like multilayer perceptron; however, the hidden layers in this method are convolutional layers that apply cross-correlation to the inputs. Generally, the time series data structure is one-dimensional grid at a time interval. Thus, time series applications utilize one-dimensional CNNs and it can be defined as:

$$S_t = (P^a * W)(t) = \sum_{\alpha=-\infty}^{\infty} X(\alpha)\, W(t-\alpha), \tag{5.3.1}$$

$$L_t = f(W_L \times S_t + b_L), \tag{5.3.2}$$

where P^a denotes aggregated power, W is the weighting function (kernel filter), α is the weighted average, and S is the convolutional output which is called feature map for the continuous time t. The $L(t)$ denotes the load classification and prediction outputs, $f(.)$ denotes the activation function, the W_L denotes the hidden to output weights and the b_L is the hidden to output bias vector.

The LSTM operates principally in the same way of the recurrent neural networks; however, it employs more gates for the recurrent neurons. The LSTM can be defined as:

$$i_t = g_1(W_{i,cr} \times L_t + W_{i,pr} \times \hat{P}_{t-1} + b_i), \tag{5.3.3}$$

$$f_t = g_1(W_{f,cr} \times L_t + W_{f,pr} \times \hat{P}_{t-1} + b_f), \tag{5.3.4}$$

$$o_t = g_1(W_{o,cr} \times L_t + W_{o,pr} \times \hat{P}_{t-1} + b_o), \tag{5.3.5}$$

$$U = g_2(W_{U,cr} \times L_t + W_{U,pr} \times \hat{P}_{t-1} + b_U), \tag{5.3.6}$$

$$C_t = f_t \times C_{t-1} + i_t \times U, \tag{5.3.7}$$

$$\hat{P}_t = o_t \times g_2(C_t), \tag{5.3.8}$$

where g_1 denotes the sigmoid function, g_2 denotes the hyperbolic tangents function, L_t is the input vector to the LSTM which is the extracted features and the output from the CNN, i_t is input gate, f_t is the forget gate, o_t is the output gate, U is the update signal, C_t is the state value at computation time t and $\hat{P}_t$ is the predicted output vector from the hybrid deep learning model. $W_{(.)}$ and $b_{(.)}$ are the weight matrices and bias vectors, respectively. The weights correspond to the current state are $W_{(.),cr}$ and the previous state are $W_{(.),pr}$.

5.4 Modeling and Results

The studied residential building is modeled using Building Energy Optimization (BEopt) software [88]. The building's specification was chosen as default in the Denver Int. AP location using the B10 benchmark option and weather option. The building floor area is 1265 sqft with two floors and one garage. In addition, the building consists of three beds and two baths. An EV with average electricity usage of 1998 kWh/year, and a PV with a size of 6 kW are considered. The designing process can supply related data of indoor/outdoor building characteristics, cooling and heating interactions, hot water energy consumption, appliance choices, and plug energy consumptions. The generated datasets are illustrated as in Fig. 5.2 for one year with one hour time resolution.

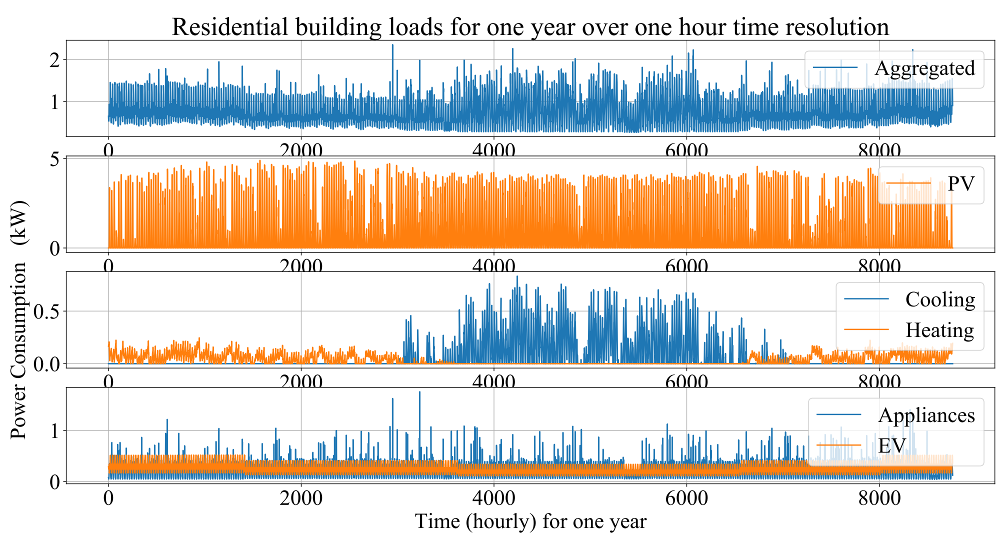


Figure 5.2: Shows the simulated residential building dataset for one year with one hour resolution including aggregated load from the smart meter, PV generation, cooling, heating, appliances and plug-in EV load.

The performance of the proposed method is evaluated using conventional metrics such as root mean square error (RMSE) and normalized RMSE (NRMSE). Considering different time scales for energy consumption in one day is vital to evaluate the model properly. The considered time scales or time window for energy disaggregation are one hour, three hours and six hours. Similarly, the considered time scales for prediction of solar generation are one hour, three hours and six hours. Generally, the two parts models are trained with 70% of the dataset and tested with 30% of the data accordingly.

Fig. 5.3. shows the performance of the proposed model for energy disaggregation in three time scales. The figure shows that the performance of the proposed method outperforms learning methods such as support vector regression (SVR), LSTM and CNN for all time scales. It is worth noting that the disaggregation for one hour time scale is the best performance in comparison with other time scales.

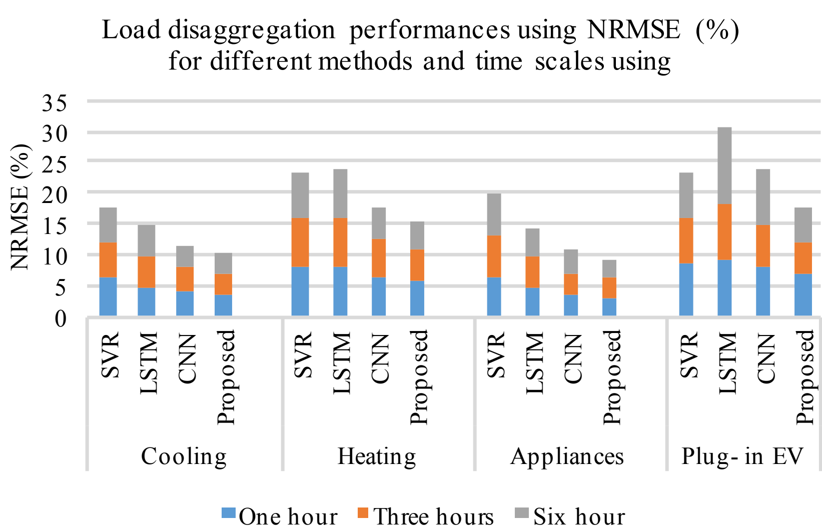


Figure 5.3: Disaggregation performance of different methods and time scales.

Fig. 5.4 and Table 5.1 shows the performance results of the proposed method for solar energy forecasting and net load prediction. The results are illustrated for two weeks of PV prediction and measured net load. The graph consists of the original values for the aggregated load, PV generation and net load. It is worth noticing that the forecasted PV generation follows the original values of the PV generation. In addition, the net power load follows the original simulated values.

Table 5.1: Solar energy forecasting performance using NRMSE (%) for different methods and time scales

Method	One hour	Three hours	Six hours
SVR	8.964	9.487	9.982
LSTM	6.956	7.255	7.940
CNNs	6.605	6.726	7.129
Proposed	6.392	6.569	6.656

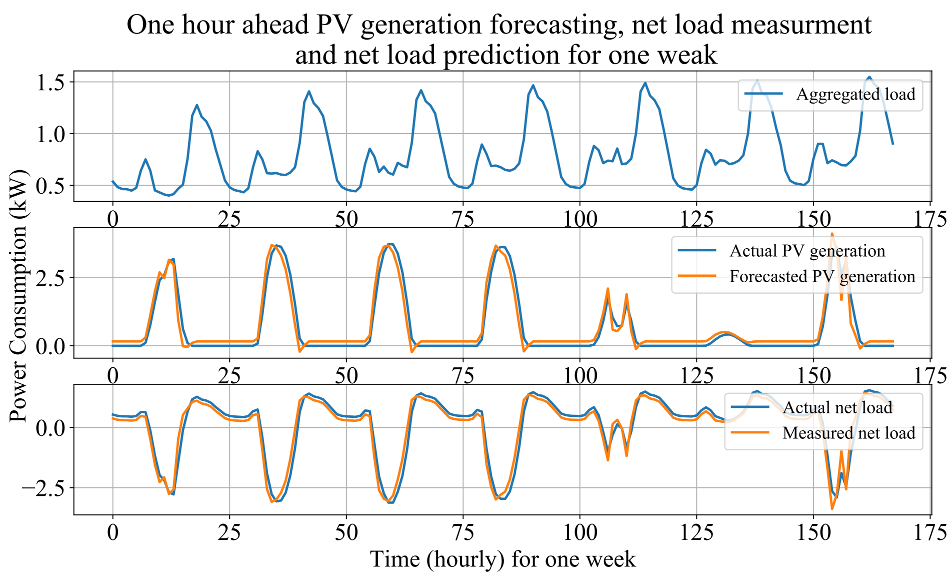


Figure 5.4: Solar energy forecasting and predicted net load using the proposed method.

Chapter 6

Conclusion and Future Research

6.1 Conclusion

This dissertation focuses on the use of the DL to improve the load prediction (load forecasting and energy disaggregation) at the distribution level through multiple proposed approaches. In general, the DL methods outperformed conventional prediction methods in all studies in this dissertation for load prediction. In addition, hybrid DL methods performed better than traditional DL methods, and the hybridizing showed the effectiveness of combining two successful DL methods. The following are the accordingly conclusions to each approach in each chapter.

An investigation of energy consumption forecasting for buildings was studied in Chapter 2. Firstly, recursive and non-recursive ANN was considered to be modeled for energy consumption forecasting in residential and commercial buildings. The numerical results showed that the recursive ANN including LSTM and GRU which is a type of the DL achieved the best performance. Then, an evolutionary-based development to the LSTM model to improve load forecasting accuracy in buildings was proposed. The proposed approach combined the GA with the LSTM method by evolving the window size, a number of hidden neurons and number hidden layers. The proposed model presented better performance than conventional prediction methods. In addition, the proposed method achieved 17.319% improvement in comparison with traditional LSTM for the residential building. Also, the proposed approach achieved 10.669% improvement for the commercial building. The reasoning behind the evolutionary learning concept is that for DL algorithms, it is faster and efficient to find the optimized window size and the optimized number of hidden neurons than to find them based the developer's knowledge and experimental trials. Therefore, the proposed approach showed the effectiveness of finding optimal prediction accuracy combining evolutionary computation with DL methods.

In chapter 3, a complex approach for a smart building was proposed and all operational parameters in the building were considered in the DL model for training and testing. Due to the complexity of smart building modeling using all the operational parameters, it was considered infeasible to conduct precision building analysis and control, until the emerge of ACP theory and modern artificial intelligence technology. A hybrid DL method (LSTM-GRU) was proposed to investigate building energy consumption prediction and the numerical results showed that all MTS operational parameters performed better than using few operational parameters. Also, the proposed method outperformed the benchmarked models. The analysis performed in the chapter showed that the hybrid DL model is a powerful artificial intelligence tool for modeling multivariable complex systems, and has the potential to be applied in different areas, e.g., smart building, smart office, smart home, and the smart city due to its outperformance in this chapter.

Chapter 4 investigated the accuracy of power consumption forecasting at distribution and building level and proposed the approach of hybrid DL method (CNN-GRU) to improve the forecasting accuracy and coupled with the ACP theory. The main contribution of this research can be summarized as an effective short-term power consumption forecasting with big data, and parallel computation was proposed for real buildings datasets at the DU. Since the research investigated the seasonality and different timescales effects, the study concluded that the bigger training dataset outperformed smaller dataset. Moreover, the performance of the proposed model was compared with conventional and the proposed approach had the best performance among other methods. The outcomes of the proposed approach can contribute to the fields of energy saving, smart grid planning, electricity bidding market, and demand response.

Finally, in chapter 5, a precision energy disaggregation based on hybrid DL method (CNN-LSTM) was proposed for residential building modeling using BEopt software. The disaggregation modeling included EV loads and PV generation which are considered as continuously varying load. The proposed method performed better than the conventional disaggregation method and obtained a high forecasting accuracy for solar energy and net load measurement. It can significantly help the energy suppliers to estimate the internal residential loads, appliances, plug-in EV charging, solar generation, and net load approximations. Thus, it can dramatically increase the certainty of demand response applications and dynamic pricing.

6.2 Future Research

In the future work, there are a couple of potential research areas that can be considered for load prediction at the distribution level. There are still questions about hybridizing other different DL methods such as autoencoder, RBM, DBN, and DBM. Also, hybridizing more than two successful DL methods would produce more accurate prediction, since hybridizing two successful DL methods performed better in this dissertation. Another direction, there is potential research of using reinforcement learning for control operation in buildings as an agent and environment framework. Also, there is still possible research of using ACP framework for energy consumption prediction at residential buildings.

Bibliography

[1] K. Amarasinghe, D. Wijayasekara, H. Carey, M. Manic, D. He, and W. P. Chen, "Artificial neural networks based thermal energy storage control for buildings," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Nov 2015, pp. 005 421–005 426.

[2] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov 2011.

[3] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1483–1488.

[4] L. Prez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394 – 398, 2008.

[5] A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 511–516.

[6] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352 – 1372, 2015.

[7] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 498–505, Aug 2001.

[8] M. T. Hagan and S. M. Behr, "The time series approach to short term load forecasting," *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 785–791, Aug 1987.

[9] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1014–1020, Aug 2003.

[10] M. Hayati and Y. Shirvany, "Artificial neural network approach for short term load forecasting for illam region," *World Academy of Science, Engineering and Technology*, vol. 28, pp. 280–284, 2007.

[11] N. Kandil, R. Wamkeue, M. Saad, and S. Georges, "An efficient approach for short term load forecasting using artificial neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 8, pp. 525–530, 2006.

[12] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.

[13] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on power systems*, vol. 16, no. 1, pp. 44–55, 2001.

[14] P. A. Gonzlez and J. M. Zamarreo, "Prediction of hourly energy consumption in buildings based on a feedback artificial neural network," *Energy and Buildings*, vol. 37, no. 6, pp. 595 – 601, 2005.

[15] G. Escriv-Escriv, C. lvarez Bel, C. Roldn-Blay, and M. Alczar-Ortega, "New artificial neural network prediction method for electrical consumption forecasting based on building end-uses," *Energy and Buildings*, vol. 43, no. 11, pp. 3112 – 3119, 2011.

[16] B.-J. Chen, M.-W. Chang *et al.*, "Load forecasting using support vector machines: A study on eunite competition 2001," *IEEE transactions on power systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[17] P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669–2688, 2005.

[18] B. Dong, C. Cao, and S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy and Buildings*, vol. 37, no. 5, pp. 545 – 553, 2005.

[19] Z. Zhu, Y. Sun, and H. Li, "Hybrid of emd and svms for short-term load forecasting," in *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*. IEEE, 2007, pp. 1044–1047.

[20] Q. Ding, "Long-term load forecast using decision tree method," in *2006 IEEE PES Power Systems Conference and Exposition*, Oct 2006, pp. 1541–1543.

[21] M. A. Al-Gunaid, M. V. Shcherbakov, D. A. Skorobogatchenko, A. G. Kravets, and V. A. Kamaev, "Forecasting energy consumption with the data reliability estimatimation in the management of hybrid energy system using fuzzy deci-

sion trees," in *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, July 2016, pp. 1–8.

[22] Y. yuan Chen, Y. Lv, Z. Li, and F. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 132–137.

[23] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, July 2016, pp. 1–5.

[24] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, 2017.

[25] L. Wang, Z. Zhang, and J. Chen, "Short-term electricity price forecasting with stacked denoising autoencoders," *IEEE Transactions on Power Systems*, 2016.

[26] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting;an approach using autoencoder and lstm neural networks," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on.* IEEE, 2016, pp. 002 858–002 865.

[27] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting–a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, 2017.

[28] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE.* IEEE, 2016, pp. 7046–7051.

[29] X. Dong, L. Qian, and L. Huang, "Short-term load forecasting in smart grid: A combined cnn and k-means clustering approach," in *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on.* IEEE, 2017, pp. 119–125.

[30] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," *Energies*, vol. 10, no. 1, p. 3, 2016.

[31] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on.* IEEE, 2014, pp. 1–6.

[32] C.-Y. Zhang, C. P. Chen, M. Gan, and L. Chen, "Predictive deep boltzmann machine for multiperiod wind speed forecasting," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1416–1425, 2015.

[33] L. Song, H. Qing, Y. Ying-ying, and L. Hao-ning, "Prediction for chaotic time series of optimized bp neural network based on modified pso," in *The 26th Chinese Control and Decision Conference (2014 CCDC)*, May 2014, pp. 697–702.

[34] H. Chenglei, L. Kangji, L. Guohai, and P. Lei, "Forecasting building energy consumption based on hybrid pso-ann prediction model," in *2015 34th Chinese Control Conference (CCC)*, July 2015, pp. 8243–8247.

[35] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ann) based model predictive control (mpc) and optimization of hvac systems: A state of the art review and case study of a residential hvac system," *Energy and Buildings*, vol. 141, pp. 96 – 113, 2017.

[36] K. Li, H. Su, and J. Chu, "Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: A comparative study," *Energy and Buildings*, vol. 43, no. 10, pp. 2893 – 2899, 2011.

[37] M. D. Sulistiyo, R. N. Dayawati, and Nurlasmaya, "Evolution strategies for weight optimization of artificial neural network in time series prediction," in *2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, Nov 2013, pp. 143–147.

[38] Z. Xuan, L. Qing-dian, L. Guo-qiang, Y. Jun-wei, Y. Jian-cheng, L. Lie-quan, and H. Wei, "Multi-variable time series forecasting for thermal load of air-conditioning system on svr," in *2015 34th Chinese Control Conference (CCC)*, July 2015, pp. 8276–8280.

[39] N. Fumo and M. R. Biswas, "Regression analysis for prediction of residential energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 47, pp. 332 – 343, 2015.

[40] F. H. Al-Qahtani and S. F. Crone, "Multivariate k-nearest neighbour regression for time series data a novel algorithm for forecasting uk electricity demand," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug 2013, pp. 1–8.

[41] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761 – 1768, 2007.

[42] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[43] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[44] F. Sultanem, "Using appliance signatures for monitoring residential loads at meter panel level," *IEEE Transactions on Power Delivery*, vol. 6, no. 4, pp. 1380–1385, Oct 1991.

[45] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," *IEEE Computer Applications in Power*, vol. 12, no. 4, pp. 47–51, Oct 1999.

[46] Y. Nakano and H. Murata, "Non-intrusive electric appliances load monitoring system using harmonic pattern recognition-trial application to commercial building," in *Int. Conf. Electrical Engineering, Hong Kong, China*, 2007.

[47] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.

[48] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, *Unsupervised Disaggregation of Low Frequency Power Measurements.* SIAM, 2011, pp. 747–758.

[49] T. Zia, D. Bruckner, and A. Zaidi, "A hidden markov model based procedure for identifying household electric loads," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, Nov 2011, pp. 3218–3223.

[50] S. Singh and A. Majumdar, "Deep sparse coding for nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4669–4678, Sep. 2018.

[51] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "An unsupervised training method for non-intrusive appliance load monitoring," *Artificial Intelligence*, vol. 217, pp. 1 – 19, 2014.

[52] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, “Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview,” in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, June 2015, pp. 1175–1180.

[53] F. Jazizadeh, B. Becerik-Gerber, M. Berges, and L. Soibelman, “An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems,” *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 311 – 326, 2014.

[54] L. Mauch and B. Yang, “A new approach for supervised power disaggregation by using a deep recurrent lstm network,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2015, pp. 63–67.

[55] C. Duarte, P. Delmar, K. W. Goossen, K. Barner, and E. Gomez-Luna, “Non-intrusive load monitoring based on switching voltage transients and wavelet transforms,” in *2012 Future of Instrumentation International Workshop (FIIW) Proceedings*, Oct 2012, pp. 1–4.

[56] Y. Lin and M. Tsai, “Applications of hierarchical support vector machines for identifying load operation in nonintrusive load monitoring systems,” in *2011 9th World Congress on Intelligent Control and Automation*, June 2011, pp. 688–693.

[57] L. Jiang, J. Li, S. Luo, S. West, and G. Platt, “Power load event detection and classification based on edge symbol analysis and support vector machine,” *Appl. Comp. Intell. Soft Comput.*, vol. 2012, pp. 27:27–27:27, Jan. 2012.

[58] L. Du, Y. Yang, D. He, R. G. Harley, T. G. Habetler, and B. Lu, “Support vector machine based methods for non-intrusive identification of miscellaneous

electric loads," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct 2012, pp. 4866–4871.

[59] M. Figueiredo, A. de Almeida, and B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (nilm) systems," *Neurocomputing*, vol. 96, pp. 66 – 73, 2012, adaptive and Natural Computing Algorithms.

[60] S. Rahimi, A. D. C. Chan, and R. A. Goubran, "Nonintrusive load monitoring of electrical devices in health smart homes," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, May 2012, pp. 2313–2316.

[61] E. Mocanu, P. H. Nguyen, and M. Gibescu, "Energy disaggregation for real-time building flexibility detection," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, July 2016, pp. 1–5.

[62] S. Giri, M. Bergs, and A. Rowe, "Towards automated appliance recognition using an emf sensor in nilm platforms," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 477 – 485, 2013.

[63] N. Batra, A. Singh, and K. Whitehouse, "Neighbourhood NILM: A big-data approach to household energy disaggregation," *CoRR*, vol. abs/1511.02900, 2015.

[64] Y. Lin and M. Tsai, "A novel feature extraction method for the development of nonintrusive load monitoring system based on bp-ann," in *2010 International Symposium on Computer, Communication, Control and Automation (3CA)*, vol. 2, May 2010, pp. 215–218.

[65] S. Biansoongnern and B. Plangklang, "Nonintrusive load monitoring (nilm) using an artificial neural network in embedded system with low sampling rate," in *2016 13th International Conference on Electrical Engineering/Electronics,*

Computer, Telecommunications and Information Technology (ECTI-CON), June 2016, pp. 1–4.

[66] F. Paradiso, F. Paganelli, A. Luchetta, D. Giuli, and P. Castrogiovanni, "Ann-based appliance recognition from low-frequency energy monitoring data," in *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, June 2013, pp. 1–6.

[67] H. Chang, K. Chen, Y. Tsai, and W. Lee, "A new measurement method for power signatures of nonintrusive demand monitoring and load identification," *IEEE Transactions on Industry Applications*, vol. 48, no. 2, pp. 764–771, March 2012.

[68] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, ser. BuildSys '15. New York, NY, USA: ACM, 2015, pp. 55–64.

[69] L. D. Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using vi trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32 – 36, 2018.

[70] D. de Paiva Penha and A. R. G. Castro, "Convolutional neural network applied to the identification of residential equipment in non-intrusive load monitoring systems," in *3rd International Conference on Artificial Intelligence and Applications*, 2017, pp. 11–21.

[71] A. F. Ebrahim and O. A. Mohammed, "Pre-processing of energy demand disaggregation based data mining techniques for household load demand forecasting," *Inventions*, vol. 3, no. 3, 2018.

[72] W. He and Y. Chai, "An empirical study on energy disaggregation via deep learning," *Advances in Intelligent Systems Research*, vol. 133, 2016.

[73] J. Kim, H. Kim *et al.*, "Classification performance using gated recurrent unit recurrent neural network on energy disaggregation," in *Machine Learning and Cybernetics (ICMLC), 2016 International Conference on*, vol. 1. IEEE, 2016, pp. 105–110.

[74] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem," *CoRR*, vol. abs/1703.00785, 2017.

[75] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[76] N. F. Esa, M. P. Abdullah, and M. Y. Hassan, "A review disaggregation method in non-intrusive appliance load monitoring," *Renewable and Sustainable Energy Reviews*, vol. 66, pp. 163 – 173, 2016.

[77] J. Zhang, Z. Zhan, Y. Lin, N. Chen, Y. Gong, J. Zhong, H. S. H. Chung, Y. Li, and Y. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, Nov 2011.

[78] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

[79] "Buildings datasets," 2012.

[80] F. Chollet *et al.*, "Keras," `https://github.com/fchollet/keras`, 2015.

[81] Z.-L. Sun, D.-S. Huang, C.-H. Zheng, and L. Shang, "Optimal selection of time lags for tdsep based on genetic algorithm," *Neurocomputing*, vol. 69, no. 7, pp.

884 – 887, 2006, new Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks.

[82] K. Lukoseviciute and M. Ragulskis, "Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems," *Neurocomputing*, vol. 73, no. 10, pp. 2077 – 2088, 2010, subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[83] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[84] F. Wang, "Toward a paradigm shift in social computing: The acp approach," *IEEE Intelligent Systems*, vol. 22, no. 5, pp. 65–67, Sept 2007.

[85] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," *Energies*, vol. 10, no. 1, 2017.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[87] S. Kumar, L. Hussain, S. Banarjee, and M. Reza, "Energy load forecasting using deep learning approach-lstm and gru in spark cluster," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, Jan 2018, pp. 1–4.

[88] C. Christensen, R. Anderson, S. Horowitz, A. Courtney, and J. Spencer, "Beopt(tm) software for building energy optimization: Features and capabilities," 8 2006.

[89] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, `http://www.deeplearningbook.org`.

[90] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[91] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[92] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[93] O. Kramer, *Machine learning for evolution strategies.* Springer, 2016, vol. 20.

[94] A. Almalaq and J. J. Zhang, "Evolutionary deep learning-based energy consumption prediction for buildings," *IEEE Access*, vol. 7, pp. 1520–1531, 2019.

[95] A. Almalaq, "Gated recurrent unit applied for energy consumption forecasting in building sectors," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, April 2018.

[96] A. Almalaq, J. Hao, J. J. Zhang, and F.-Y. Wang, "Parallel building: A complex system approach for smart building energy management," iEEE/CAA Journal of Automatica Sinica [Accepted].

[97] A. Almalaq and G. Edwards, "Comparison of recursive and non-recursive anns in energy consumption forecasting in buildings," 2019 IEEE Green Technologies Conference (GreenTech) [Accepted].

Appendix

A.1 Deep Learning Methods

A.1.1 Non-Recursive Artificial Neural Network (ANN)

The non-recursive ANN has direct connections from the input to the output where each neuron in the input layer is connected to the neurons in the hidden layers. The mathematical representation for non-recursive ANN is described as follows:

$$y_i = \zeta(W^T \times x_i + b) \tag{.0.1}$$

where $\zeta(.)$ denotes the activation function, T denotes transpose, x_i denotes inputs, y_i denotes the outputs, and W and b are the weight matrix and the bias vector, respectively. There are two common non-recursive ANN as follows:

Multilayer Perceptron (MLP)

Generally, the multilayer perceptrons (MLP) is an implementation of artificial neural networks (ANN) that has no feedback connections within the network [89].

The MLP may consist of three or more layers including an input, an output, and one or more hidden layers. Usually, the input layer is not counted as one layer; therefore, the graph representation of two-layer MLP is drawn as in Fig. A.1(a). The MLP is widely proposed for load forecasting in the literature, as in [90], [13], [91].

Typically, the MLP is used for supervised learning problems that can be solved with the back-propagation (BP) algorithm. There are two steps to compute the gradients. The first step is forward propagation, which propagates the initial information of the inputs up to the hidden units at each layer to produce the predicted values. The second step is BP, which computes the partial derivatives of the cost function with respect to the network.

Radial Basis Function Network (RBFN)

The RBFN measures similarities between the new input value and previous input values from the training dataset. The approximation result is the Euclidean distance between the input and its similarity.

A.1.2 Recursive Artificial Neural Network (RNN)

The recursive ANN has recurrent connections from the output to the input in the hidden layer. It is commonly applied to time series sequence because it has a memory state in its architecture that assists sequential data to be processed. The mathematical representation for the recursive ANN is defined as follows:

$$h_i = \zeta(W_h^T \times h_{i-1} + W_x^T \times x_i + b) \tag{.0.2}$$

$$y_i = W_y^T \times h_i \tag{.0.3}$$

where $\zeta(.)$ denotes the activation function, T denotes transpose, x_i denotes inputs, y_i denotes the outputs, h_i denotes the hidden state, h_{i-1} denotes the previous hidden state, W_x denotes the input to hidden weights, W_h denotes the recursive weights in the hidden layer, W_y denotes the hidden to output weights and b is the bias vector. There are two common types of the recursive ANN which have gating units as follows:

Long Short-Term Memory (LSTM)

The LSTM method is one type of the RNN that is designed to provide a longer-term memory and overcome the vanishing of gradient descent in the RNN. In the LSTM model, internal self-loops are used for storing information where there are five crucial elements in the computational graph such as input gate, forget gate, output gate, cell, and state output, as shown in the Fig. A.1(b). These gates operate as reading, writing, and erasing for cell memory states. The following equations show the mathematical representation of the LSTM model:

$$i_t = \sigma(x_i W_{i,n} + h_{(t-1)} W_{i,m} + b_i), \tag{.0.4}$$

$$f_t = \sigma(x_i W_{f,n} + h_{(t-1)} W_{f,m} + b_f), \tag{.0.5}$$

$$o_t = \sigma(x_i W_{o,n} + h_{(t-1)} W_{o,m} + b_o), \tag{.0.6}$$

$$U = \tanh(x_i W_{U,n} + h_{(t-1)} W_{U,m} + b_U), \tag{.0.7}$$

$$C_t = f_t \times C_{t-1} + i_t \times U, \tag{.0.8}$$

$$h_t = o_t \times \tanh(C_t), \tag{.0.9}$$

where σ denotes the sigmoid activation function, x_i is the input vector, i_t is the input of the input gate where the subscript means input, f_t is the input of the forget gate where the subscript means forget, o_t is the input of the output gate where the subscript means output, U is the update signal, C_t is the state value at the time t and h_t is the output of the LSTM cell. $W_{(.)}$ and $b_{(.)}$ are the weight matrices and bias vectors, respectively. The weights correspond to the current state values of a particular variable are denoted as $W_{(.),n}$ and previous state signal as $W_{(.),m}$. The memory state can be modified by the decision of the input gate using a sigmoid function with on/off state. If the value of the input gate is minimal and close to zero, there will be no change in the state cell memory.

Gated Recurrent Unit (GRU)

A recent approach for overcoming the vanishing gradient descent in the RNN is called the gated recurrent unit (GRU) algorithm [92]. It is similar to the LSTM architecture by having gating units, but with fewer gates and parameters than the LSTM. The GRU architecture consists only of two gates, which are a reset gate r_t and an update gate z_t as shown in Fig. A.1(c). It does not include external memory and output gate. The mathematical representation of the GRU is simpler than LSTM as follows:

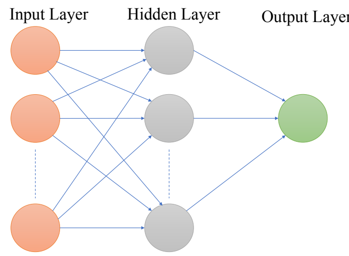
$$z_t = \sigma(x_i W_{z,n} + h_{(t-1)} W_{z,m} + b_z), \tag{.0.10}$$

$$r_t = \sigma(x_i W_{r,n} + h_{(t-1)} W_{r,m} + b_r), \tag{.0.11}$$

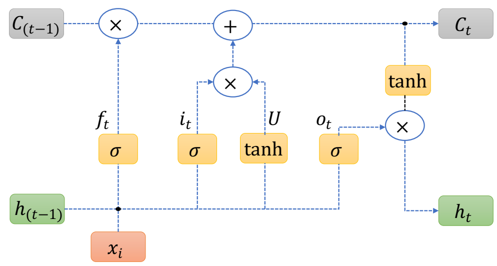
$$U = \tanh(x_i W_{U,n} + [r_t \odot h_{(t-1)}] W_{U,m} + b_U), \tag{.0.12}$$

$$h_t = (1 - z_t) \odot h_{(t-1)} + z_t \odot U, \tag{.0.13}$$

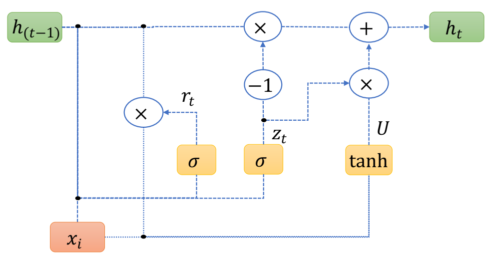
where σ denotes the sigmoid activation function, x_i is the input vector, h_t is the output vector, U is the update signal, and $\odot$ is element-wise multiplication. The weights correspond to the current state values of a particular variable, are denoted as $W_{(.),n}$ and previous state signal as $W_{(.),m}$. $W_{(.)}$ and $b_{(.)}$ are the weight matrices and bias vectors, respectively.



(a) MLP network architecture.



(b) LSTM cell block diagram.



(c) GRU cell block diagram.

Figure A.1: (a) The graph representation of a two layers MLP architecture. The representation includes one input layer for input variables, one hidden layer for hidden neurons, and one output layer for outcome neuron. (b) The block diagram of the LSTM cell. i_t, f_t, o_t and U are the input gate, the forget gate, output gate and the update signal, respectively. (c) The block diagram of the GRU cell. r_t, z_t and U are the reset gate, the update gate, and the update signal, respectively.

A.1.3 Convolutional Neural Network (CNN)

The CNN, which represents a type of DL algorithms, mimics the structure of human neurons and is applied widely in various applications including visual processing, video recognition, and natural language processing. This method is commonly used for processing grid data topology which includes a two-dimensional grid of pixels for image data construction [89]. However, the construction of the time series data is one-dimensional grid at a time interval, and we will apply the CNN for sequential time series. The mathematical convolution operation is employed in at least one of the CNN layers [89].

The convolution operation in signal processing is described generally in the following equation:

$$S(i) = (X * w)(i) = \sum X(a)\, w(i-a) \tag{.0.14}$$

where X is called input, w is the kernel filter, a is the weighted average and S is the convolutional output which is called feature map for the continuous time i.

Typically, two-dimensional CNN consists of three main stages that build the architecture of the network. The first stage has the convolutional layer, the second stage has the detector that is the rectified linear activation, and the third is the pooling function layer [89].

However, the one-dimensional CNN consists of two stages for the convolutional layer and pooling layer. Fig. A.2 shows an example of six inputs neurons, four convolutional neurons and two pooling neurons for one-dimensional CNN. The colors of the connections form three sets of the convolutional neurons that represent the kernels, and the same color represents sharing weights. The convolutional layer maps the input features, and the pooling layer extracts the important mapped features.

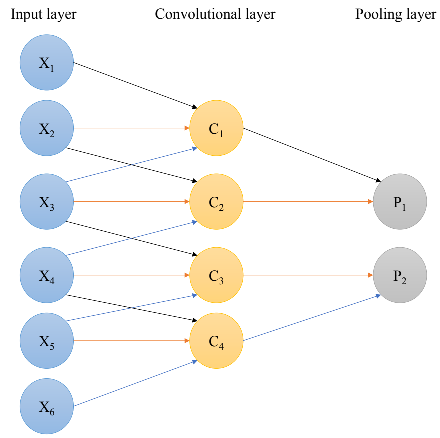


Figure A.2: The One-dimensional CNN example of six inputs with one convolutional layer and one pooling layer.

A.1.4 Other Deep Learning Methods

Autoencoder

Autoencoder is one of the feed forward neural network that is used to copy input neurons to output neurons by passing through a hidden layer or multiple hidden **h** layers as in stacked Autoencoders. [89].

The main parts of Autoencoder network architecture are based on encoder function $\mathbf{h} = f(x)$ and a decoder for reconstruction $\widehat{\mathbf{x}} = g(\mathbf{h})$. Therefore, the reconstructed output is $\widehat{\mathbf{x}} = g(f(x))$ which copies the data input. The mathematical representation of Autoencoder is described as follows:

$$\widehat{\mathbf{x}} = g(\mathbf{W}\mathbf{x} + \mathbf{b}) \qquad (.0.15)$$

where $\mathbf{x}$ is the input, $\mathbf{W}$ is the weights, $\mathbf{b}$ is the bias and g is the activation function which could be sigmoid or a rectified linear function. Figure A.3 shows the simple architecture of Autoencoder for three layers input, hidden and output layer. This learning algorithm is usually used for dimensionality reduction, feature learning or corrupted data reconstruction. The Autoencoder models that are used for these kinds of problems are known as Undercomplete Autoencoder, Sparse Autoencoder and Denoising Autoencoder, respectively [89].

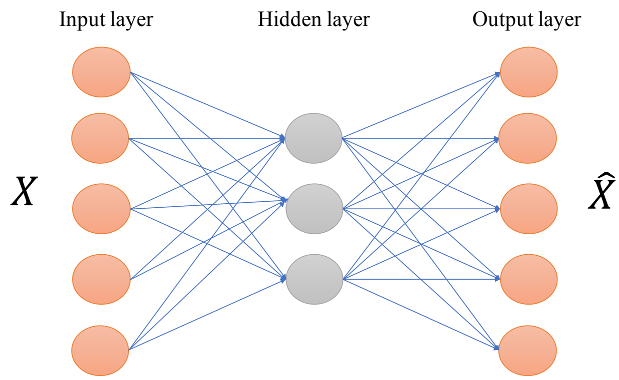


Figure A.3: Shows the architecture of the Autoencoder learning algorithm.

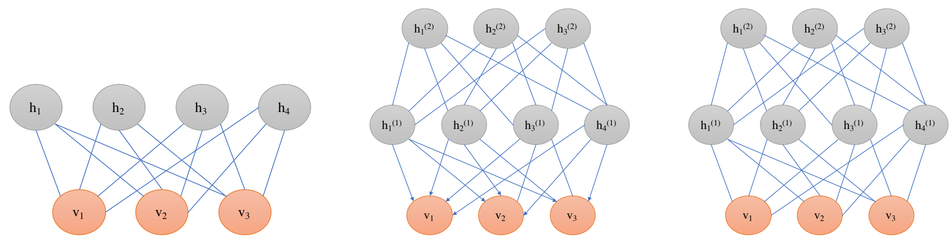
Restricted Boltzmann Machine (RBM)

RBM is one of the most famous Deep probabilistic models which are undirected probabilistic graphical models [89].

RBM has two main layers where the first contains visible inputs and the second layer contains hidden variables. Usually, RBM is stacked in order to make it deeper by building one on top of the other. Figure A.4a shows simple RBM for the undirected probabilistic graphical model for two layers.

Deep Belief Network (DBN)

DBN architecture has several layers of hidden units known as stacked RBM with multiple hidden layers that are trained using the Backpropagation algorithm [30]. Basically, the connection units in the DBN architecture is between each unit in a layer with each unit in the next layer, however, there are no intra connection for each layer units [89].



(a) RBM (b) DBN (c) DBM

Figure A.4: Shows different architecture of a. RBM with undirected connections between visible inputs and hidden variables. b. DBN with directed connections toward visible inputs and the others undirected connections for the hidden layers. c. DBM with undirected connections for one visible inputs layer and multi hidden layers.

Figure A.4b shows DBN for three layers configuration, two hidden and one visible layer where the top two layers are undirected, but, the connection between all other layers should be directed pointing towards the data layer [89]. Generally, a DBN is RBM with more hidden layers, however, RBM usually have only one hidden layer.

Deep Boltzmann Machine (DBM)

DBM neural network is like RBM architecture but with more hidden variables and layers than RBM. Also, DBM is unlike to DBN because DBM architecture has entirely undirected connections between variable within all layers, such a visible layer and multi hidden layers [89].

Figure A.4c shows the architecture of DBM neural network for three hidden layers and one visible layer.

A.2 Genetic Algorithms (GA)

The GA is a common nonlinear optimization algorithm which solves constrained and unconstrained optimization problems and provides an optimal or near-optimal solution through searching in a complex space. It is, found by Holland in 1975, an adaptive global optimization search based on natural selection of Darwinian analogy and genetic biology [93] and utilizes crossover and mutation probabilities to guide the search of an optimum solution (individual) in the fitness function. The GA is based on a population search where a set of candidate solutions (individuals) of the fitness function are obtained after a series of iterative computations. One of the advantages of the GA is less sensitive to initialization due to the nature of mutation and crossover probabilities, however, it is not the best method for online implementation due to its slow convergence in a complex space [93].

The individuals are composed of chromosomes, which are candidate solutions, based on the Darwinian principle of survival of the fitness value. The fitness function determines the living ability and living quality of each individual as depending on the evolutionary process of the GA.

There are three major operators of the evolutionary process in the GA, which are the crossover operator, the mutation operator, and the selection operator. These operators directly affect the fitness value searching process, and find the most optimum solution. Another strategy in the GA that pledges the convergence of the fitness value to the optimum is elitism selection which means copying the best individual in the generation to next generation [93]. Nevertheless, the chromosome length and crossover method, such as one-point crossover, two-point crossover, etc., are important techniques to find the optimum value in the efficient process.

The operation of crossover, which is the most important operation in the GA algorithm, is a random exchange of two chromosomes that are genotyped in a binary gene's base using one of the crossover methods as Fig. A.5. The mutation operation is the random alteration in one gene or more from 1 to 0 or vice versa. The selection operation is the process of selecting the highest fitness value among the population's individuals by using a selection method, e.g., the roulette wheel and tournament selection.

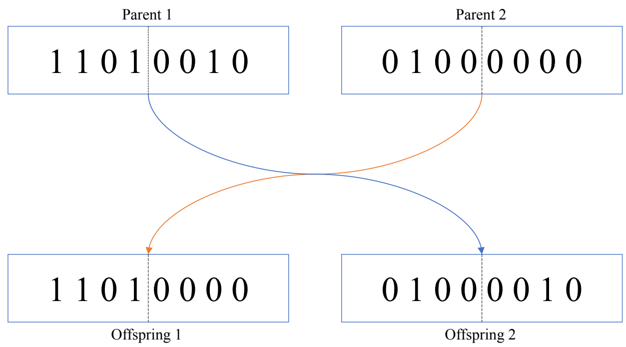


Figure A.5: One point crossover operation.

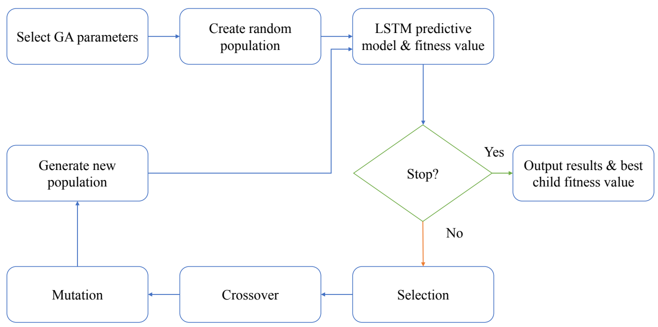


Figure A.6: The GA algorithm operation scheme.

Moreover, The population size and number of generation are important factors that influence computation complexity. If the population size, which implies the number of the solution in each generation, is too large, the GA algorithm will cost large computation quantity and the probability of plunged local optimum is low. If the population size is small, the algorithm complexity will be reduced and the likelihood of falling in a local optimum is high.

The convergence of the evolutionary process in the GA algorithm is found with iterative steps, where the termination criterion is pre-defined with the maximum number of iteration. Fig. A.6 shows an illustration of the GA iteration process and the basic process of the GA steps is as follows:

1. Generate initial population randomly.

2. Evaluate the fitness value of each individual in the population.

3. Perform the crossover operation.

4. Perform the mutation operation.

5. Perform the selection method.

6. Stop the GA algorithm if the termination criterion is satisfied, otherwise, return to number (2).

A.3 List of Publications

- A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 511–516

- A. Almalaq and J. J. Zhang, "Evolutionary deep learning-based energy consumption prediction for buildings," *IEEE Access*, vol. 7, pp. 1520–1531, 2019

- A. Almalaq, "Gated recurrent unit applied for energy consumption forecasting in building sectors," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, April 2018

- A. Almalaq, J. Hao, J. J. Zhang, and F.-Y. Wang, "Parallel building: A complex system approach for smart building energy management," iEEE/CAA Journal of Automatica Sinica [Accepted]

- A. Almalaq and G. Edwards, "Comparison of recursive and non-recursive anns in energy consumption forecasting in buildings," 2019 IEEE Green Technologies Conference (GreenTech) [Accepted]