Deep-Learning Framework for Estimating Behind the Meter Solar Generation and Electric Vehicle Penetration Level and Time-of-Use

By

Mohamed Hassan Abdalla

A thesis

presented to the University Of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

Author Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The continual increase in the adoption of rooftop solar/photovoltaic (PV) generation and electric vehicles (EVs) presents challenges, as well as opportunities, in distribution power systems. Without monitoring or control, the addition of PV generation and EV charging to distribution power systems can result in power stability, as well as power congestion issues. In this research, a deep-learning framework is presented in order to monitor and estimate the penetration level of PV generation and EV charging in distribution power systems. The proposed framework is also developed to predict the time-of-use of EV charging in order to enable scheduling for demand response programs. Additionally, the framework presented in this research is capable of estimating the generated solar power behind the meter for improving distribution system operational planning as well as power procurement plans. The framework identifies the houses that include PVs or EVs and monitor their behind the meter solar generation as well as the time-of-use of EVs, through the use of only existing smart meter data, and it can also be scaled to include other flexible appliances of interest. In order to improve the overall performance of the inference system and mitigate error propagation, the framework exploits various customized sub-models that are specifically built for each sub-target. In this research, the framework was evaluated using real smart meter data from Pecan Street Dataport and achieved a promising 93-98% F-score across all its sub-models, which proves the feasibility and scalability of our approach.

Acknowledgments

First and foremost, all thanks and gratitude is to Allah, who is showering us with His countless blessings.

I would also like to express my heartfelt gratitude and appreciation to my supervisor, Dr. Ramadan, for his continuous support, guidance and encouragement throughout my master's program in both academics and non-academic matters.

I would also like to express my gratitude to Abdulla Al Ghurair Foundation for Education (AGFE) for giving me this opportunity to pursue my master's degree and for their generous support.

My deepest thanks and endless appreciation to my parents, brothers and sister for their continuous support, love, care and encouragement.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

PV	Photovoltaic
EV	Electric Vehicle
GHG	Greenhouse Gas
DR	Demand Response
ToU	Time-of-Use
PAR	Peak-to-Average Ratio
RNN	Recurrent Neural Network
BRNN	Bidirectional Recurrent Neural Network
NILM	Non-intrusive Load Monitoring
VC	Vehicle Controller
HMM	Hidden-Markov Model
GRU	Gate Recurrent Unit
FIT	Feed-in-Tariff
ANN	Artificial Neural Network
ReLU	Rectified Linear Unit
GMDH	Group Method of Data Handling
CNN	Convolution Neural Network
BP	Back-Propagation
TDNN	Time-Delay Neural Networks
BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm
LSTM	Long Short-Term Memory
CECs	Constant Error Carousels Units
TTS	Text-to-Speech
BPTT	Back-Propagation through Time
RUS	Random Under-Sampling
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
ROC	Receiver Operating Characteristics
AUC	Area Under the Curve
MSE	Mean Square Error
R^2	Coefficient of Determination

بسم الله الرحمن الرحيم

اللهم صلي و سلم على نبينا محمد

الحمد لله الذي بنعمته تتم الصالحات

Chapter 1

Introduction

1.1 Preamble

The adoption of photovoltaic (PV) generation and electric vehicle (EV) penetration in our power system presents a chance to reduce carbon dioxide emissions and a step toward greener power consumption and generation. Now a day, advanced countries are subject to limit their CO2 emission per the rules of the Kyoto Protocol [1] [2]. Generally, with respect to internal combustion engine vehicles, EVs provide a reduction of 17-30% in greenhouse gases (GHG) and a 19-23% CO2 reduction over their life cycle [3]. Additionally, if the charging energy source of EVs is a renewable energy source, e.g. PV generation, CO and CO2 emission caused by vehicles can be reduced by up to 85% [4] [5] Therefore, many countries view PV and EV adoption as a promising solution to reduce CO2 emission. As a result, such countries highly incentivize the implementation of rooftop PV generation as well as EV consumption with generous energy selling rates and added tax credits [2] [4] [5] [6] [7]. In turn, the number of homes with PV generation and/or EV consumption has been rapidly increasing [4] [6] [7] [8] [9] [10] [11] [12].

As the overall cost of residential PV installation continues to fall, the US residential solar market, for example, experienced a record of annual installation capacity [9]. Fig. 1.1 shows the annual residential solar PV installation capacity over the years of 2010-2019 according to [9]. At the same time, sales of EVs have topped 2.1 million globally in 2019, surpassing the annual sales record and boosting global EV stock in 2019 [12]. Fig. 1.2 shows the global EV stock over the years of 2010-2019 according to [12].

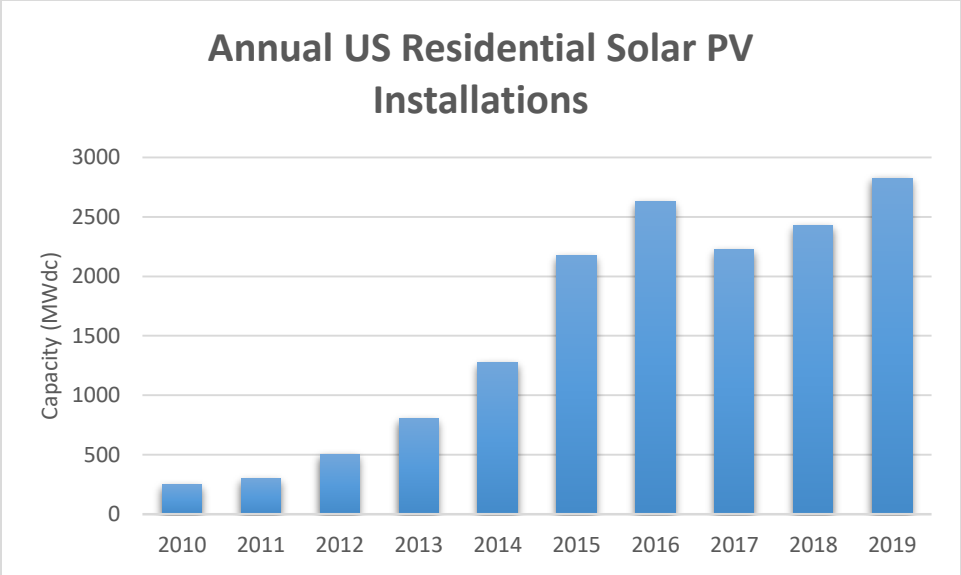


Fig. 1.1 Annual residential solar PV installation capacity [9]

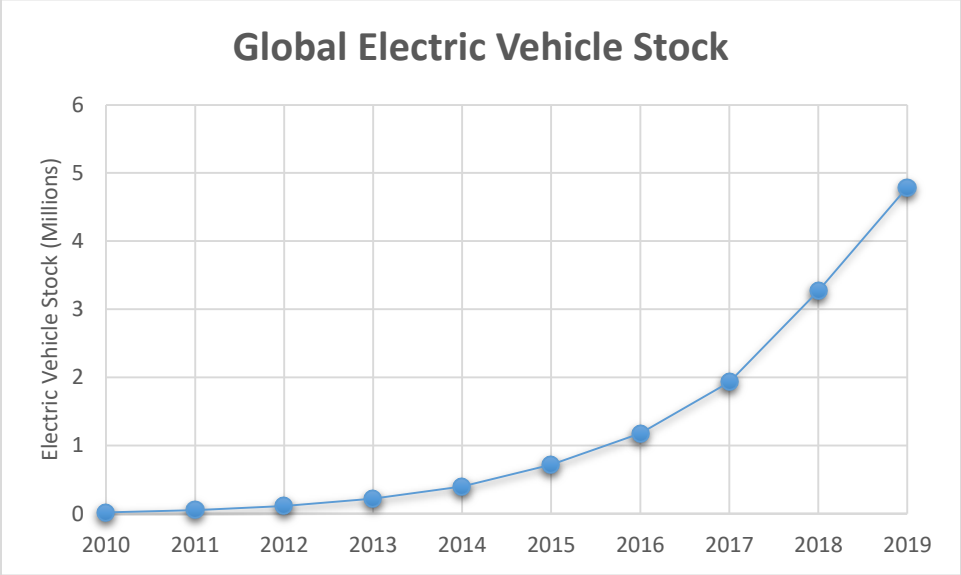


Fig. 1.2 Global EV stock over the years of 2010-2019 [12]

Aside from the environmental advantages of this increase, EVs charging and PV generation greatly reshape the residential consumption, and hence the net consumption. At high penetration levels, without accurate knowledge of the amount of EVs and PV generation in the system, the efficiency of power system operational planning will be reduced due to over/underestimation of the net load in the system throughout the day. Such poor planning can, in the case of underestimation of the net load, result in serious power congestion problems overloading the assets of the

distribution system and calling for high-cost distribution power assets upgrade. On the other hand, it could impair our ability to forecast the demand accurately and provide an adequate supply at the appropriate time, resulting in voltage fluctuation beyond acceptable limits and mal-functioning of voltage-regulation and protection devices [13] [14] [15] [16] [17].

For example, according to [2], a Nissan Leaf EV driving an average of 18,129 km/year consumes what corresponds to 74% of the average one-person residential electricity consumption per year in the US. As a result, uncontrollable EV charging could increase the peak demand significantly. An unplanned peak demand increase of such level would certainly overload distribution system assets reducing their life expectancy and require installations of higher capacity assets [13] [14] [15]. According to [2], to supply this uncontrollable EV charging, 82 of 1 GW additional nuclear power plant units would be required in the US for instance. However, as a flexible load, scheduled EV charging flattens the peak demand [2] [17] and in the same analysis [2] can afford 70% EV penetration without additional power plants in the US, hence, not requiring assets upgrades in the distribution power system.

On the supply side, while solar PV generation presents operational benefits for distribution system utilities in addition to the environmental merits [18] [19] [20], the invisibility of behind-the-meter solar generation impair the accurate forecasting of the net demand as well as efficient short-term operational power system planning. Additionally, if not properly monitored, increased residential behind-the-meter solar generation risks voltage fluctuation beyond acceptable limits, increased system losses and malfunctioning of protection devices [29] [21] [22] [23] [24].

For proper scheduling of flexible loads, such as EV charging, as well as accurate PV generation monitoring, the utilities must be able to observe the penetration level as well as the time-of-use of flexible loads. Additionally, proper monitoring of residential behind-the-meter solar generation is required.

1.2 Motivation

In light of the above, an accurate estimation of the amount of PV generation and EV penetration level in the system is crucial for efficient power system operational

planning as well as accurate demand forecasting. Additionally, to avoid possible power congestion and the need to upgrade existing power distribution assets, EV charging must be scheduled. To support the scheduling of flexible loads, such as EVs, as part of the movement toward intelligent and decentralized demand response (DR) operation with a high level of comfort for end-customers, one must be able to estimate ownership information of such flexible loads as well as their time-of-use (ToU), e.g. EVs time of charge [25], [26]. This estimated information can be used to balance the on-peak and off-peak demand more efficiently and minimize the peak-to-average ratio (PAR) [27], [28]. Estimation of the amount of PV generation and EV penetration in the system, as well as the EVs' time of charge or any other flexible load's ToU, could be achieved by either intrusive or non-intrusive load monitoring.

To achieve such estimation using intrusive load monitoring, additional power meters must be installed for every new EV and PV generation system. Although such method will yield accurate results, the additional hardware requirements make intrusive load monitoring unreasonably complicated and expensive. Unlike intrusive load monitoring, non-intrusive load monitoring requires no additional hardware but rather obtains the required estimations from the existing aggregated house consumption smart meters' measurement using inference models. Nevertheless, an accurate inference model with high performance typically requires prior behaviouristic information about each customer, appliance ownership information, high-resolution consumption data, or on-site training, which complicate the scalability of such approaches. Therefore, it seems that a model that doesn't require any prior information about the customers' life-style or their appliance ownership information, that is suitable to accurately predict PV generation and flexible loads' ToU from existing low-resolution smart meter data with no on-site training requirement is needed. The proposed model would facilitate the operational power distribution system planning and DR programs without any hardware requirements.

1.3 Research Objectives

The main objective of this research is to develop a deep-learning-based framework that is capable of accurately estimating the amount of PV generation and EV penetration level in the system, as well as EVs' time of charge as an example of

ToU prediction of flexible loads. The proposed framework must only utilize the already existing low-resolution, aggregated smart meter consumption data as its input, and achieve high inference scores on each sub-target. The objectives of this work are as follows:

- Introducing a model that is able to accurately monitor PV generation and EV penetration level status in the distribution system with no additional hardware requirement

- Enabling a scalable data-driven algorithm for flexible appliances ToU inference using only the already existing low-resolution aggregated smart meter consumption data. Unlike algorithms that require periodic surveys for appliance ownership information or the installation of specialized instrumentation for on-site training, the framework introduced in this study is able to recognize the appliance signature from the aggregated net consumption and infer the ToU by utilizing its WaveNet inspired sub-models and bidirectional recurrent neural network (BRNN) sub-model.

- Introducing an accurate behind-the-meter solar generation estimation model that utilizes net smart meter data to estimate the solar generation of invisible residential PV systems

1.4 Thesis Organization

The rest of the research is organized as follows. Chapter 2 discusses related studies including literature on non-intrusive load monitoring (NILM) and energy disaggregation as well as a brief background on deep-learning and deep-learning architectures. The proposed penetration level and ToU inference model and its building sub-models blocks are introduced and evaluated in chapter 3. Chapter 4 presents our behind-the-meter solar generation estimation model and its performance on unseen real data. Chapter 5 concludes the research.

Chapter 2

Literature Review and Background

2.1 Preamble

As the adoption of PVs and EVs has been rapidly increasing, many studies have addressed the consequences of this increase and the possible solutions. In this section, we discuss what has been presented in the literature in terms of the operational challenges introduced by the high penetration level of PVs and EVs, the presented solutions by DR programs and ToU inference, as well as the state of the art NILM and energy disaggregation solutions. After that, we briefly discuss deep-learning and artificial neural networks as the base learning model for deep-learning.

2.2 Literature Review

2.2.1 Penetration level and Time-of-Use Related Work

Nowadays, the adoption of roof-top PV generation and EVs is rapidly increasing [4] [6] [7] [8] [9] [10] [11] [12]. As a result, many studies have discussed the consequences of high PV and EV penetration levels of our distribution power system [2], [29]. The study at [29] discussed how roof-top PV generation poses operational challenges to our distribution power system when it is not properly monitored. Such challenges include but are not limited to impacting the quality of demand forecast and the ability to provide an adequate supply at the appropriate time. This in turn can lead to power congestion, voltage fluctuation, protection devices malfunctioning, and so forth. On the other hand, Chang et al. in his analysis in [2] explained that at high EV penetration levels, EV charge could make the peak

demand higher beyond our power distribution assets capacities if left unscheduled. To support the scheduling of EV charge or other flexible loads, ToU inference is essential.

Various studies that investigated DR and EVs can be found in the literature. In [30], E. S. Parizy et al. present a heuristic DR for scheduling the consumption of flexible loads for each customer in order to reduce PAR. While [31] discussed EV coordinated charging, the authors in [32] discussed the statistical modelling of EV uncoordinated charging. A method for controlling EV charging using vehicle controller agents (VC) for minimizing the charging cost has been studied in [33]. F. Rassaei et al. [27] introduced an algorithm for decentralized DR of EVs usage to shape the daily consumption and minimize the peak demand. While DR studies have discussed the scheduling of flexible appliances, most of these studies require costly smart meter upgrades to enable the monitoring of flexible loads and control their ToU.

In light of the importance of ToU inference, various predictive models that take the aggregate power time series data as its input and infer appliances' ToU, either directly or as a result of energy disaggregation, have been suggested. In this context, energy disaggregation refers to NILM or extracting the consumption of individual appliances from the aggregated power time-series. [34], [35] and [36] are well-known surveys in the field of NILM. Many NILM suggested inference models that are based on variations of Hidden-Markov model (HMM) or non-machine-learning models [37] [38] [39] [40]. In [37], a power disaggregation model based on Factorial HMM is presented as a NILM tool. The model is built on the assumption that only one device can change at a time. It also assumes that there is no correlation between the power consumption of different household appliances.

To relax the assumptions in HMM-based models such as that of [37], the authors of [38] utilized correlation studies to cluster correlated loads into "*super devices*" using a normalized cross-correlation measure at first. The consumption of "*super devices*" would then be disaggregated from the total consumption using Factorial-Hidden-Markov Model. After that, the consumption of each load would be obtained using a table for the level of consumption of each load in the "*super device*" cluster for each time sample.

D. Piga et al. [39] formulated the NILM problem as a least-square error minimization problem with the assumption that household appliances' consumption is piecewise constant overtime. The suggested algorithm in [39] requires the time-of-use probability of each appliance, which is behavioural specific and could differ from one customer to another. It also assumes that only one appliance can change its operation state at each time instant.

A pulse-to-appliance association algorithm for load disaggregation is suggested in [40]. In their study, the authors focused on processing the aggregated load power signal and extracting the pulses of the different appliances. The pulses are then matched with appliances based on various features: the existence of spikes, variance, time of use, neighbouring pulses, … etc. The algorithm requires the customers to register every appliance and its tabulated power value. It also relies on some behaviouristic parameters that would change from one customer to another based on their consumption behaviour, e.g. time of use probability, the sequence of operations, … etc. Not to mention that the algorithm relies on parameters that need to be calibrated for each house, which requires monitoring the consumption of each individual appliance for a specific period of time.

On the other hand, various machine-learning-based models were suggested as NILM solutions. Basu et al. [41] discussed the performance of multiple classifiers such as Bayesian classifiers and decision trees. In [42], a deep-learning energy disaggregation algorithm using Gate Recurrent Units (GRU) recurrent neural network (RNN) is proposed. The algorithm was built on the UK DALE dataset using a specified number of appliances (maximum of 20 appliances). Liu et al. [43] investigated the use of transfer learning to utilize pre-trained image classification models for NILM, based on the voltage and current trajectories. In [26] Afzalan et al. suggested a dense neural network for ToU inference from aggregated power time-series given appliance ownership information.

These models have been built based on customer-specific behaviouristic parameters that require on-site training, appliance ownership information, or high-resolution power time-series data (1 Hz or more) which complicate the scalability of such models for ToU inference. However, in this study, we present a high-performance model that takes existing low-resolution smart meter data as its only

input and predicts/monitors the penetration level of EVs and PV generation in the system, the time of charge of EVs as a tool for the scheduling of EV charging and it can be scaled to predict the ToU of other flexible appliances. In addition, while the majority of NILM studies are aimed at helping electricity customers reduce their consumption and save on electricity bills, the algorithm introduced in this research is aimed at presenting a tool for utility providers to monitor the penetration level of EVs and PV generation in the system, as well as assisting the scheduling of flexible appliances' consumption for DR purposes.

2.2.1 Behind-the-Meter Solar Generation Estimation Related Work

The adoption of roof-top solar generation can be driven by and associated with different factors. Some studies have focused on factors that are associated with the country or region in question, e.g. government incentive policies, environmental awareness and peer effect [6] [7] [8] [44] [45] [46] [47]. Other studies explored factors associated with the individual electricity customer, e.g. age, education level, income and house ownership [48] [49] [50] [51] [52].

Regardless of the dominating factor, the residential PV market continues to grow rapidly over the years [6] [7] [8] [9] [10]. For example, the U.S. solar market installed 3.6 GW of solar PV in the first quarter of 2020, setting the record for the largest amount of solar PV energy installation in a first-quarter by more than 1 GW over the second largest. Moreover, the Solar Energy Industries Association estimates that the U.S. market will install an additional 110 GW of solar energy from 2020 to 2025 [9].

The high penetration level of residential PV presents environmental benefits as well as operational opportunities for the utilities [18] [19] [20]. However, the lack of visibility of behind-the-meter solar generation could potentially pose many operational challenges. A significant amount of intermittent behind-the-meter solar generation reshapes the net load pattern [20]. At the same time, an accurate forecast of the net load is crucial for the scheduling of short-term power system operations as well as efficient power procurement plans. Additionally, if not accurately monitored, invisible solar generation can result in voltage fluctuation, increased system losses, and malfunctioning of voltage regulation and protection devices [29] [21] [22] [23] [24].

A residential customer with rooftop solar generation is commonly part of one of two policies: the feed-in-tariff (FIT) policy or the net metering policy. In the net metering policy, residential customers sell their excess solar energy to the utility at a retail value and it only requires one bi-directional smart meter while the FIT policy requires two meters. In this part of the research, we focus on behind-the-meter solar generation estimation of customers following the net metering policy.

Various studies explored the estimation of behind-the-meter solar generation following an unsupervised approach [22] [53] [54] [55]. In [22], Kabir et al. introduced an unsupervised algorithm combining an HMM regression model with a physical PV system performance model for disaggregating net load measurement into solar generation and electric load. The algorithm iterates between calculating the electric load using the estimated solar generation through the PV performance model and estimating the PV system parameters using the predicted electric load through the HMM regression model. Although the model is robust for disaggregating PV generation from the net load, it does not have a mechanism for identifying houses or consumption days without PV generation which potentially reduces the overall performance.

On the other hand, other studies suggested using supervised machine learning approaches [23] [56] [57] [58]. Shaker et al. [23] presented a supervised data-driven approach for estimating the total behind-the-meter solar generation of a certain area using the solar generation of monitored "*representative sites*". The model requires three to four months of on-site metered solar generation data from each customer house to determine the most appropriate representative sites using a suggested *K-means+PCA* approach. The study suggests metering the solar generation of these representative sites and utilizing the solar generation data of these sites to estimate the total generation of the area using a supervised regression model.

In another study by Shaker et al. [24], the authors suggested using a fuzzy model. After selecting the "*representative sites*" using the procedure mentioned above, fuzzy numbers associated with each area are developed using possible normalized simultaneous variations on the solar generation patterns from one site to another. After that, the suggested model is able to calculate a fuzzy number associated with the real-time total solar generation of the area using the metered solar

generation data from the representative sites and the developed variation fuzzy numbers.

In a study done by Padullaparthi et al. [29], the proposed approach is capable of estimating behind-the-meter PV capacity as well as estimating the capacity of the battery used by customers following the FIT policy. The approach exploits the energy balance in the residential building by tracking the maximum energy fed to the utility and the solar irradiance at that time to calculate the PV capacity.

Even though various studies have investigated approaches toward bringing visibility to behind-the-meter solar generation of net-metered customers, the suggested approaches require on-site training for some period of time for each new customer with PV generation, assume that all the houses have solar generation and that the solar generation is daily without considering PV panels mal-functioning or maintenance days, require high smart meter data resolution (≤1Hz), or unable to accurately estimates the solar generation for cloudy days or cope with the degradation of PV panels over time.

The deep-learning framework introduced in this research is capable of estimating the solar generation of behind-the-meter PVs of individual customers following the net metering policy. The framework is capable of identifying consumption days including PV generation from days without PV generations, as well as accurately estimating the solar generation at each minute of the day. It is capable of estimating the solar generation at different weather conditions and does not require on-site training or high-resolution smart meter data (≤1Hz).

2.3 Artificial Neural Network

An artificial neural network (ANN) is the connection of various non-linear units together in what is called a network. Each non-linear unit (i.e. neuron) will perform a weighted sum of it is inputs and pass the result through a non-linear function called the ‘activation function’ (e.g. logistic, sigmoid, ReLU, … etc.).

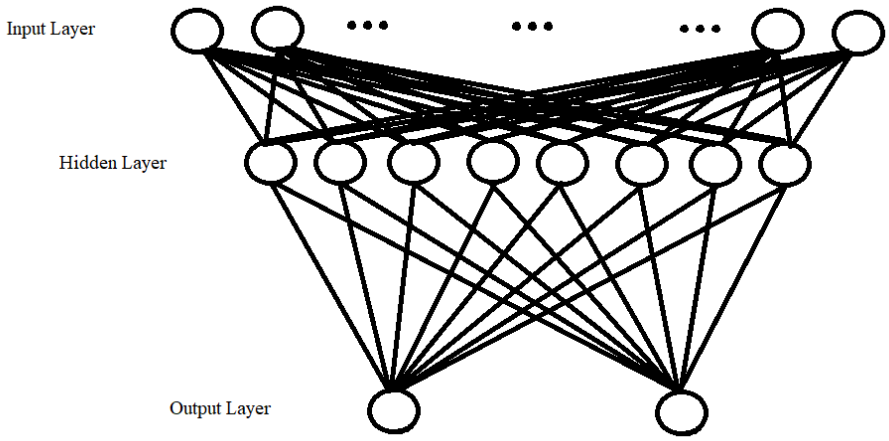


Fig. 2.1 Feed-forward artificial neural network

As shown in Fig. 2.1, ANNs consist of three types of layers:

- *Input Layer:*

  Each input node collects one feature/dimension of the input data and passes it to the first hidden layer.
- *Hidden Layer:*

  Each hidden unit computes a weighted sum of the result from the previous layer's units and passes the sum through a selected Activation Function.
- *Output Layer:*

  Each output unit computes a weighted sum of the results from the last hidden layer's units and passes the sum through a threshold function (possibly non-linear).

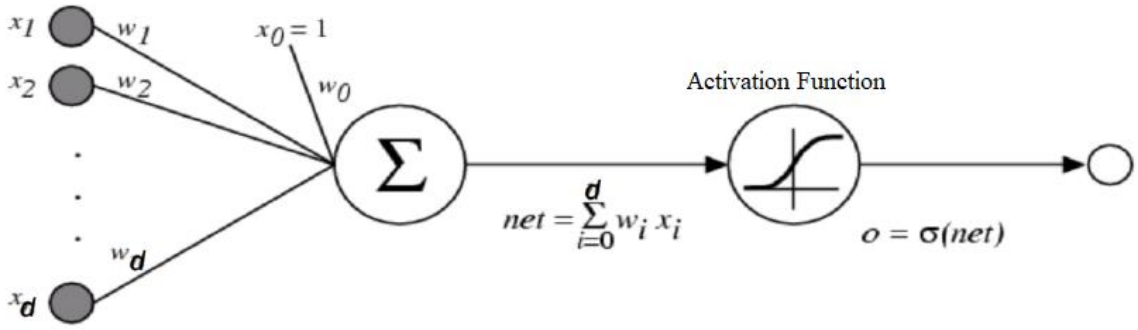


Fig. 2.2 Non-linear neuron topology

Fig. 2.2 shows the basic topology of a single neuron. As mentioned above, each neuron passes a sum of multiplication of learnable parameters called 'weights' with its inputs through an activation function. Examples of common activation functions are listed below:

- Logistic Function:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{2.1}$$

  Where L is the curve maximum value, x_0 is the midpoint of the S curve and k is the steepness of the curve.

- Sigmoid Function:

$$f(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}} \tag{2.2}$$

  Note that the sigmoid is a special case of a logistic function where $L = 1$, $x_0 = 0$ and $k = 1$.

- Tanh Function:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.3}$$

- Rectified Linear Unit (ReLU) Function:

$$f(x) = \max(0, x) \tag{2.4}$$

  ReLU or the rectifier function is an activation function defined as the positive part of it is input. Fig. 2.3 shows the graph of the ReLU function. More detail about ReLU activation functions benefits and applications can be found in section 2.4.12.

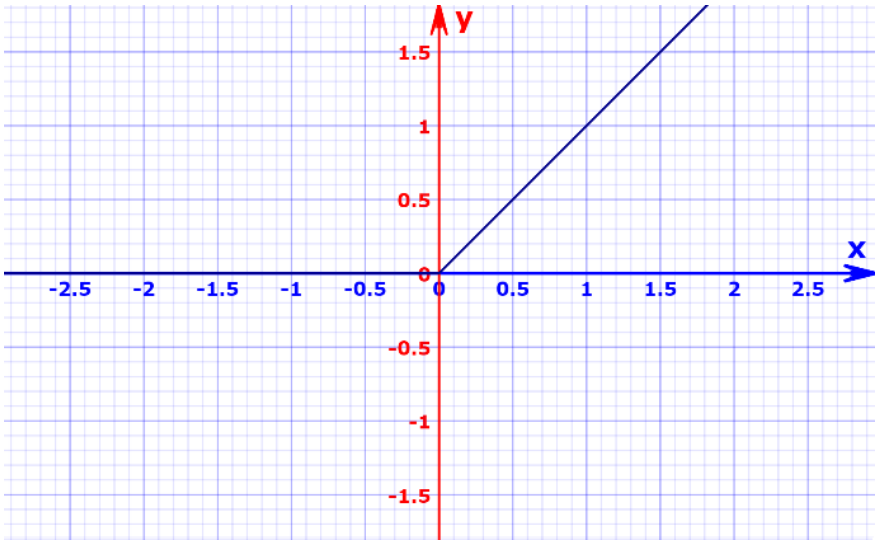


Fig. 2.3 Graph of ReLU activation function

The theory behind ANNs is that with enough hidden units an ANN can model any function. However, one challenge of ANN is regularization. Since only the number of inputs and outputs is known, the hidden structure number of layers and the number of units in each layer are arbitrary design parameters. On the one hand, with very few units, the network might have too few parameters to learn complex functions. On the other hand, with too many units, the network might be over-parameterized and hence will not be able to learn a generalized model. Nevertheless, various regularization methods have been introduced in the literature (see section 2.4).

While ANN in its simplest form consists of three layers (an input layer, a single hidden, and an output layer), a network with multiple hidden layers is capable of learning complex problems by extracting more complex features from the original input features that would facilitate solving the problem. Models that consist of multiple hidden layers are often referred to as deep-learning models. The connections between these units of every layer can be forward, backward or both. Moreover, the units could be fully connected or partially connected with each other. Each different arrangement or architecture makes a different type of deep-learning model.

2.4 Evolution of Deep-learning Neural Network

To cover the basics of deep-learning neural networks' different topologies, concepts and layers, this section briefly covers the evolution of deep-learning models throughout the years of deep-learning research.

2.4.1 Early Work on Artificial Neural Networks

Early versions of ANNs can be traced to the 1940s. McCulloch et al. [59] present a simple neuron model as a summing and thresholding component. However, this model didn't have learning associated with it. Simple ANNs that can be trained have been introduced in the following decades. Rosenblatt [60] introduced the "perceptron" model; a two-layer network consisting of an input layer and an output node with additional bias.

Since the basic form of supervised ANNs is essentially a variant of linear regression models, one could argue that ANNs go back to the 1800s.

In the research done by Hubel et al. [61] in 1962, simple and complex cells were found in cats' visual cortex. While simple cells fire in response to specific visual inputs such as edges, complex cells respond to more spatial invariant inputs. This finding inspired later award-winning deep ANNs architecture.

While works on networks with one hidden layer preceded the Group Method of Data Handling (GMDH) (e.g. [62]), GMDH trained networks as presented in [63] [64] [65] and [66] (1961-1971) were perhaps the earliest deep-learning models. In the GMDH trained network, the network number of layers is increased gradually and trained using a training set. After that, the unnecessary additional layers are pruned using a separate validation set. As a result, the number of layers in a network differs based on the learnt target. Various applications of GMDH trained networks can found in the literature, e.g. [64], [67], [68] and [69].

2.4.2 Convolution layers, weight sharing and pooling layers

Another early ANN that deserved to be labelled as a deep ANN is the Neocognitron introduced by Fukushima in 1979 [70] [71] [72]. It incorporated the neurophysiological findings of section 2.4.1 and introduced the weight-sharing concept of convolution neural networks (CNN). In CNN units, weights are shared through the network as a weight vector (also known as a filter) that is gradually shifted through the input array (e.g. the pixels of an image). The resulted array of this process can then be used as an input of later CNN units and so forth. As a result of this weight sharing mechanism, only a few learnable parameters are necessary for each convolution layer.

In addition, Neocognitron introduced down-sampling layers, currently known as pooling layers. In the pooling layers, the units have fixed weights in which the output is insensitive to small changes in the input. For example, the output could be the maximum value of the input vector (i.e. max pooling).

However, Fukushima's Neocognitron weights are not trained using back-propagation but rather set using local, winner-take-all, unsupervised learning rules. In addition, the down-sampling layers in the Neocognitron mainly used spatial averaging rather than the current commonly used max-pooling layer.

2.4.3 Artificial Neural Network Back-Propagation

Error minimization of non-linear, differentiable, multi-stage ANN systems through gradient descent has been used in the literature since the 1960s, e.g. [73], [74], [75] and [76]. However, the systems introduced in the 1960s back-propagated derivatives using standard Jacobian matrix from layer to another without clearly addressing direct links between consecutive layers or efficiency improvement due to the sparsity of the network.

According to [77], the first description of back-propagation of error in discrete, possibly sparsely connected, ANN-like network was first discussed in a 1970 master's thesis [78]. On the other hand, the first ANN application of efficient back-propagation (BP) was in the 1980s [79], after which several studies have been published in the application of BP to ANNs, e.g. [80], [81], [82] and [83].

In general, the back-propagation algorithm is an iterative algorithm for getting the gradient efficiently. For each training sample (one can also use a batch gradient) forward propagation is performed by computing the output of the training sample. After that, the current output units' error is computed. Then the hidden units' error is computed. Finally, the network weights are updated.

2.4.4 The Need for Improvement of Artificial Neural Network

Regardless of the efficiency of back-propagation, by the late 1980s, it was clear that back-propagation by itself is not sufficient. As a result, most ANNs applications consisted of networks with very few hidden layers, as there seemed to be no practical advantage of deeper networks. Many deep-learning scientists were convinced by the Universal Approximation Theorem [84] which suggests that a three-layer ANN is capable of modelling any possible decision function for mapping continuous input into a finite set of classes at any desired level of accuracy. However, the theorem does not state the number of required units. In general, while the number of required hidden units to cover the desired decision function could be exponential, using more layers reduces the required number of hidden units to achieve a generalizing network.

In addition, while several methods were introduced to deal with extended time lag in RNNs, e.g. Focused back-propagation that is based on decay factors in RNNs [85] [86], Time-Delay Neural Networks (TDNNs) [87] and their adaptive extension [88], and other methods [89] [90] [91], each method had its own major drawback [79]. As a matter of fact, specific benchmark problems that are used for the evaluation of these methods can be solved more rapidly by "randomly guessing RNN weights until a solution is found" [92].

2.4.5 Improved Back-Propagation through Advanced Gradient Decent

Studies have proposed various improvements of steepest decent through back-propagation. While Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) [93] [94] [95] [96] and least-square methods [97] [98] [99] are computationally expensive for large ANNs, partial BFGS [100] [101], conjugate gradient [102] [103], and other methods [104] [105] are sometimes useful fast options.

To improve the speed of back-propagation convergence, [106] suggests the use of momentum. Additionally, ad-hoc constants are added to the slope of linearized activation functions [107]. On the other hand, the non-linearity of the slope is signified [108].

R-prop back-propagation variant [109] takes into account the sign of the error derivatives, as well as iRprop+ [110], which was successfully applied to RNNs.

Based on the ANN structure, local gradient normalization can be achieved [111] using a Hessian approach [112] or other efficient approaches [113].

Many additional tricks related to improving ANNs have been introduced in the literature, e.g. [114], [115], [116], [117] and [118]

2.4.6 Back-Propagation for Convolution Neural Networks

A Neocognitron-like, weight-sharing, convolution layered network had back-propagation applied to it in 1989 [119] [120] [121]. The GPU-based CNN plus max-pooling layered network is currently an essential component of many modern, award-winning, computer vision neural networks. This research was also the introduction of the MNIST data set of handwritten digits [119], which is currently the most famous benchmark data set for machine learning. In the 1990s, CNNs achieved good results on the MNIST data set [120] and on fingerprint recognition [122]; as a result, similar CNNs were used commercially in this decade.

2.4.7 The Fundamental Deep-learning Problem of Gradient Decent

In 1991, Hochreiter [123] formally identified why deep feedforward ANNs or RNNs are hard to train using back-propagation: typical networks experience the famous problem of *vanishing or exploding gradients*. In this problem, with typical

activation functions, the propagated cumulative error signals decay or explode exponentially with the number of layers. This is also known as the *long time lag problem*. The problem encouraged many researchers in that decade, e.g. [124], [125] and [126]. Over the years, many solutions to the vanishing or exploding gradients have been explored:

- A deep learner that overcomes the problem using unsupervised pre-training for stacks of RNNs [127] [128] which facilitate consequent supervised training using back-propagation has been introduced. As for feedforward ANNs, similar results can be attained using Auto-Encoders [129] and Deep Belief Networks [130].
- Long-Short-Term-Memory (LSTM) networks overcome the problem through a special mechanism that is unsusceptible to the problem, see section 2.4.9.
- The computational capabilities of today's GPUs allow propagating the error a few further layers within a reasonable time when compared to the machines of the 1990s.
- Hessian-free optimization can overcome the problem for feedforward ANNs and RNNs [113] [131] [132] [133] [134].

2.4.8 Max-Pooling: toward Max-Pooling Convolution Neural Network

Inspired by the Neocognitron, the Cresceptron [135], which adapts its training topology, utilized max-pooling layers instead of the winner-take-all method in the Neocognitron. In max-pooling layers, the previous layer's units' activations are partitioned into small rectangular arrays. Each array is then replaced by the activation of its maximally active unit.

When max-pooling is combined with CNNs in alternating convolution and max-pooling layers, a network similar to Cresceptron is formed. However, unlike Cresceptron max-pooling CNNs are trained using back-propagation. Advantages of max-pooling CNNs were pointed out at a later date [136]. Max-pooling CNNs are

becoming pivotal to many modern, award-winning deep ANNs architecture, e.g. [137] and [138].

2.4.9 Long-Short-Term-Memory Recurrent Neural Network

LSTM RNNs [137] [139] [140] have performed very well in deep-learning benchmark problems and were able to overcome the *vanishing or exploding gradient* problem without prior unsupervised learning. Moreover, LSTM-based networks have the ability to learn targets without local sequence predictability solving very complex targets [141].

The basic concept of LSTM is as follows. Constant Error Carousels units (CECs) use the identity function as an activation function and have a connection to itself with a constant weight of one. Since the derivative of the identity function is a constant (= 1), the back-propagated errors would not vanish or explode but stay the same. To learn non-linear behaviours, CECs are connected to various non-linear adaptive units. The changes in the weights of these units are often the result of back-propagated error far in time through the CECs units. As a result, CECs allow LSTM-based networks to learn the importance of events that happened thousands of time-steps ago, while standard RNNs have failed the case of minimal time lag of 10 steps [77].

LSTMs are said to be neuro-physiologically plausible to a certain extent [142]. Various previously unsolvable deep-learning targets have been learnt using LSTM, e.g. arithmetic operations on continuous input streams, robust storage of high precision real numbers across extended time intervals and recognition of temporally extended patterns in noisy input sequences [137] [139]. In addition, LSTMS significantly outperformed standard RNNs in problems that require knowledge of the rules of regular languages that can be described by deterministic *Finite State Automata* [143] [144] [145] [146] [148].

Recently, LSTM-based RNNs have won various pattern recognition competitions and set the records for various large and complex benchmark data sets, e.g. [149], [150] and [151].

2.4.10 Recent Tricks for improving Deep Artificial Neural Network Performance

Dropout layers randomly disregard some hidden units during the training to improve the generalization of the network [152] [153]. Some deep-learning scientists view dropout layers as an ensemble method in which various data models are trained at the same time [154]. A deterministic approximation of dropout layers called *fast dropout* [155] was adapted for RNNs for faster training and evaluation [156]. Dropout layers are closely related to a technique that is neuro-physiologically plausible, in which noise is added to the neurons during the training to achieve a perturbation-resistant network [157] [158] [159] [160] [161].

ReLU activation function has been widely adopted in various deep-learning applications [162] [163]. In fact, ReLU has outperformed sigmoidal activation on various benchmarks [164] and helped in setting the record for on several other benchmarks [165] [166].

In our framework, we exploit the aforementioned tricks with our WaveNet inspired models (which essentially a form of 1-D CNN) and LSTM BRNN.

2.5 Summary

In this chapter, we discussed a literature review of related work for both penetration and ToU estimation and behind-the-meter solar estimation. Additionally, ANNs and the evolution of deep learning have been discussed.

Chapter 3

Penetration Level and Time-of-Use Estimation

3.1 Preamble

In this chapter, we discuss the framework used to infer the PV generation and EV penetration level and estimate the ToU of EV as an example of flexible appliance ToU inference. After that, we will discuss the evaluation of this framework on unseen real data. The framework takes a full day worth of 1-minute resolution smart meter data of the targeted houses as its input. Based on the daily smart meter data, it first predicts whether or not there was any PV generation, EV consumption, or consumption of any other flexible appliance in question. After that, the model infers the ToU using specified sub-models. The inference is made by recognizing the power usage signature of the targeted appliances, EVs and PV generation in this case, from the aggregated power usage of the target house. The framework is illustrated in Fig. 3.1. The building components of this framework as well as the results on unseen real data are discussed in detail in the following sections.

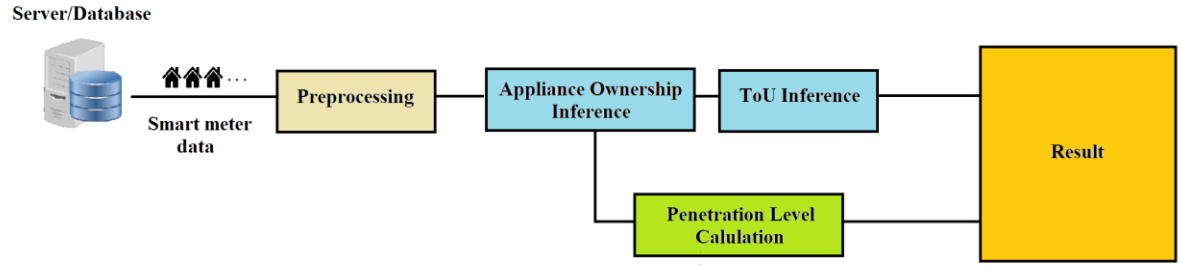


Fig. 3.1 General structure of the inference framework

3.2 Data Requirements

The data used for model training and evaluation purposes consisted of:

1) *aggregated smart meter power time-series sampled at 1-minute resolution,*
2) *information about the ownership of targeted power signature (PVs and EVs) for each house,*
3) *data from sub-meters at the level of targeted appliances.*

Recently, with the demand for publicly available consumption data, datasets that satisfy such requirements are available for hundreds of houses [167]. Fig. 3.2 shows an example of a one-day data sample that is used for model training.

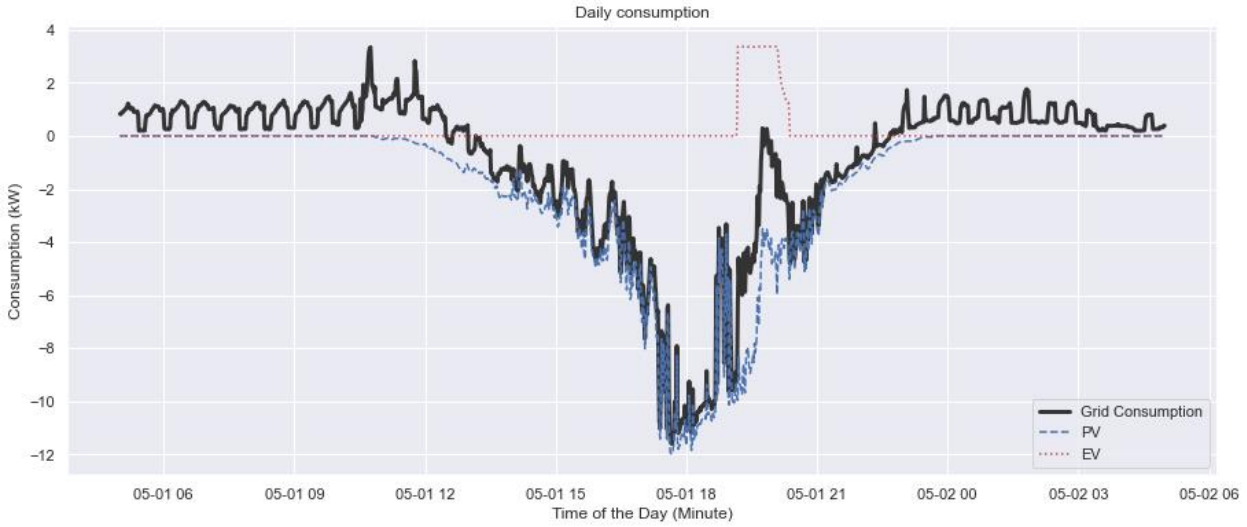


Fig. 3.2 One-day smart meter consumption data sample

Once the model is trained, it can be employed for inference on new data (out of the training data). The only required characteristic of the new data, for which the trained model is to make an inference, is as follows: aggregated smart meter power time-series sampled at 1-minute resolution. Such data are already available using the vastly adopted smart meters.

3.3 Appliance Ownership Inference and Penetration Level Estimation

Given the time-series of aggregated smart meter power measurement for the daily consumption of house i on day n as $P_{i,n}(t)$ ($t \in [1,1440]$), where t is in minutes, the objective is to recognize the signature of the target j (PV, EV or other flexible loads) in aggregated smart meter data and estimate whether it has been used during this day $y_{i,n,j}$. $y_{i,n,j}$ has the following binary form:

$$y_{i,n,j} = \begin{cases} 1, & if\ appliance\ j\ was\ used \\ 0, & if\ appliance\ j\ was\ not\ used \end{cases} \quad (3.1)$$

The inferred $y_{i,n,j}$ can then be used to estimate the penetration level in a city or a neighbourhood. Given $y_{i,j}$ of I houses the penetration level α_j of target j in these houses can be calculated as follows:

$$\alpha_j = \frac{\sum_{i=1}^{I} y_{i,j}}{I} \quad (3.2)$$

Where $y_{i,j}$ is obtained after monitoring $y_{i,n,j}$ for several days and can be used for targeting houses with appliance j for DR programs. On the other hand, α_j can be used for monitoring purposes.

3.4 Appliance Time-of-Use Inference

Given the predicted $y_{i,n,j}$ and the time-series of aggregated smart meter power measurement for the daily consumption of house i on day n as $P_{i,n}(t)$ ($t \in [1,1440]$), where t is in minutes, the objective is to recognize the consumption signature of the targeted appliance j and estimate the ToU vector $\Gamma_{i,n,j}$ as $\{\gamma_1, \gamma_2, \ldots, \gamma_m, \ldots, \gamma_M\}$, where M represents the number of time bins in a day. On the other hand, γ_m is a binary variable that is defined as follows:

$$\gamma_m = \begin{cases} 1, & if\ appliance\ j\ was\ used\ in\ the\ time\ bin\ m \\ 0, & if\ appliance\ j\ was\ not\ used\ in\ the\ time\ bin\ m \end{cases} \quad (3.3)$$

The inferred $\Gamma_{i,n,j}$ can be used to verify if house i has used appliance j in their assigned time bin of the day in accordance with a DR program. In this study, we suggest that a separate sub-model should be trained for each time bin to provide flexibility in model selection and improve overall inference framework performance. In addition, even though ToU sub-models follow the appliance ownership inference,

training these sub-models against misclassified $y_{i,n,j}$ increases the robustness of these sub-models and reduces error propagation throughout the inference framework as explained in section 3.7.

3.5 Learning Models

Even though various learning models have been evaluated for each sub-model, two main deep-learning models achieved the best performance and were chosen for our inference framework because of their capability to relate different time samples and "learn through time":

- WaveNet

First introduced by Google [168] as a fully probabilistic autoregressive model in which the predictive distribution for each sample depends on all previous samples. It was introduced as an audio generative model for text-to-speech applications (TTS) where it outperformed other similar models. It is the ability to model long-term temporal dependencies that allowed the WaveNet model to generate speech audio with naturalness never achieved before in the field of TTS. It utilizes causal convolution to preserve the order of the data and model the $p(x_{t+1} \mid x_1, \ldots, x_t)$. Even though the model usually requires many layers, it is relatively faster when compared to recurrent models because it does not have any recurrent connections. Where l represents the number of layers and F represents the filter length, the receptive field R can be computed as follows:

$$R = l + F - 1 \tag{3.4}$$

WaveNet utilizes a gated activation unit for which the output z can be defined as follows:

$$z = \tanh\left(W_{f,k} * x\right) \odot \sigma\left(W_{g,k} * x\right) \tag{3.5}$$

Where σ (.) is a sigmoid function, $\odot$ donate element-wise multiplication, * is a convolution operator, g and f donate filter and gate, respectively, and W is a learnable convolution filter.

- Bidirectional Recurrent Neural Network

M. Schuster et al. [169] introduced BRNN to overcome the limitations of regular RNN where the output at any time step can only be related to inputs of current and previous time steps but not future time steps. BRNN forms recurrent connections in both forward and backward directions for which back-propagation through time (BPTT) can be used to update the weights. In this research, we used the Long Short-Term Memory (LSTM) variation of BRNN.

The LSTM variation models the temporal dependency using a forget gate and an update gate. The first step in the model is to calculate the forget Ω_f at any time step t. For that, the model uses the information of the previous time step h_{t-1}, the current time sample x_t and the gate's weight and bias values:

$$\Omega_f = \sigma\,(W_f x_t + U_f h_{t-1} + b_f) \tag{3.6}$$

Where W_f and U_f, donate weight of the input and recurrent connection, respectively, and b_f donate the bias. The output is between '0' and '1'. A Ω_f value of '0' means completely forget the information and '1' means completely store the information.

The second step is to calculate the new information to be stored in the cell state. This is done in two stages. The first stage is calculating the update gate Ω_u which is similar to the forget gate but it decides which information to be updated:

$$\Omega_u = \sigma\,(W_u x_t + U_u h_{t-1} + b_u) \tag{3.7}$$

The second stage is calculating a vector of new candidate values $C`_t$ that could be added to the state cell using the tanh layer:

$$C`_t = \tanh\,(W_c x_t + U_c h_{t-1} + b_c) \tag{3.8}$$

The third step is to update the memory cell state C_t:

$$C_t = \Omega_f * C_{t-1} + \Omega_u * C`_t \tag{3.9}$$

Finally, the output gate activation vector o_t and the current hidden state vector h_t are calculated as follows:

$$o_t = \sigma\,(W_o x_t + U_o h_{t-1} + b_o) \tag{3.10}$$

$$h_t = o_t * \tanh(C_t) \tag{3.11}$$

3.6 Balancing Training and Testing Samples for Learning Models

As deep-learning is all about associations and learning patterns from training samples, imbalanced training samples degrades the performance of the learning model [170]. As a result, since the nature of the ToU data are of an imbalanced type in which the number of OFF samples is more than the ON samples, balancing the data is essential.

The data samples have been balanced using Random Under-Sampling (RUS) sampling technique. The data samples are balanced by randomly discarding instances from the majority class. Even though various advanced sampling techniques have been introduced in the literature, the RUS sampling technique is known for its practical performance in the field of predictive learning [171].

3.7 Results and Evaluation

After pre-processing, a dataset of 21,438 smart meter consumption days from 74 different buildings, primarily located in New York, California and Austin, was obtained from Pecan Street [167] and used for model training and evaluation. The data were pre-processed for missing data and measurement errors. A small number of consumption days with measurement errors and missing measurement was removed from our analysis for simplicity; however, in the case of limited data samples, methods such as interpolation and regression could be used reasonably. From these samples, 9,335 consumption days included PV generation and 3500 included EV charging, 1717 of these samples included both PV generation and EV charging. The samples have been used to train and evaluate the appliance ownership inference and penetration level estimation using PV generation and EV charging as our targeted power signatures. The samples also allowed for the evaluation and training of the time of use inference framework for predicting the time of charge during the consumption day. To prevent the sub-models from developing any unwanted bias toward any class of the population of the dataset, a stratified sampling technique was used in selecting the training and testing data for each sub-model. Using stratified sampling ensured a balanced distribution of the various classes over

the dataset. The result of the evaluation of the predictive models is discussed in the following sub-sections.

3.7.1 Appliance Ownership Inference and Penetration Level Estimation Results

Using the Pecan Street database [167], the consumption days were labelled for whether they included PV generation, EV charging or both based on the available sub-meter data. After cleaning and labelling the data, splitting the data into training and testing (80%-20% split), and normalizing the data, several predictive models were trained and compared to achieve the best sub-model for every one of our two targets: detecting PV generation and detecting EV charging in daily consumption smart meter data. To this end, we investigated K-Nearest Neighbors (KNN) model, Support Vector Machine (SVM) model, Artificial Neural Network (ANN), our WaveNet inspired model and Bidirectional Recurrent Neural Network model (BRNN).

The complexity of each learning sub-model was designed to be proportionate to the complexity of the targeted problem. For instance, the complexity of the sub-models built to detect PV generation during daily consumption is lower than the complexity of the models built to detect EV charging. This is because, in general, PV generation reshapes the daily consumption more significantly when compared to EV charging making it easier to detect, see Fig. 3.3 (a) and (b). The complexity of the learning models can be increased by adjusting the number of layers or the number of neurons in each layer in a deep learning model, or choosing a more complex kernel in an SVM model, driving the model to learn more complex features to solve the targeted problem. Fig 3.4 shows the inherent features of EV charging signals that should be learnt by the predictive models for accurate inference.

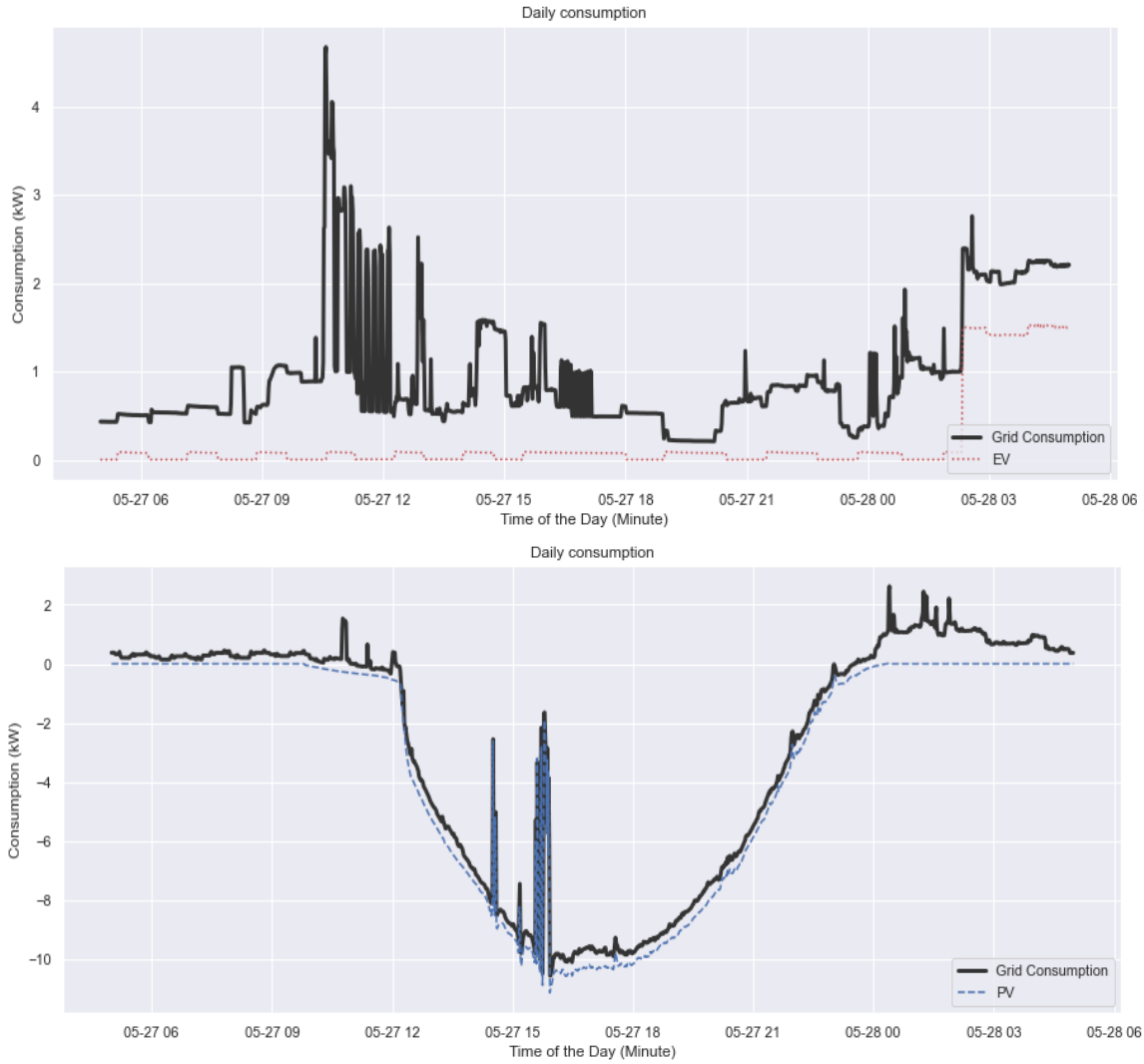


Fig. 3.3 Comparison between the reshaping effect of EV charging and PV generation on daily smart meter consumption graphs: (a) daily consumption with EV charging, (b) daily consumption with PV generation

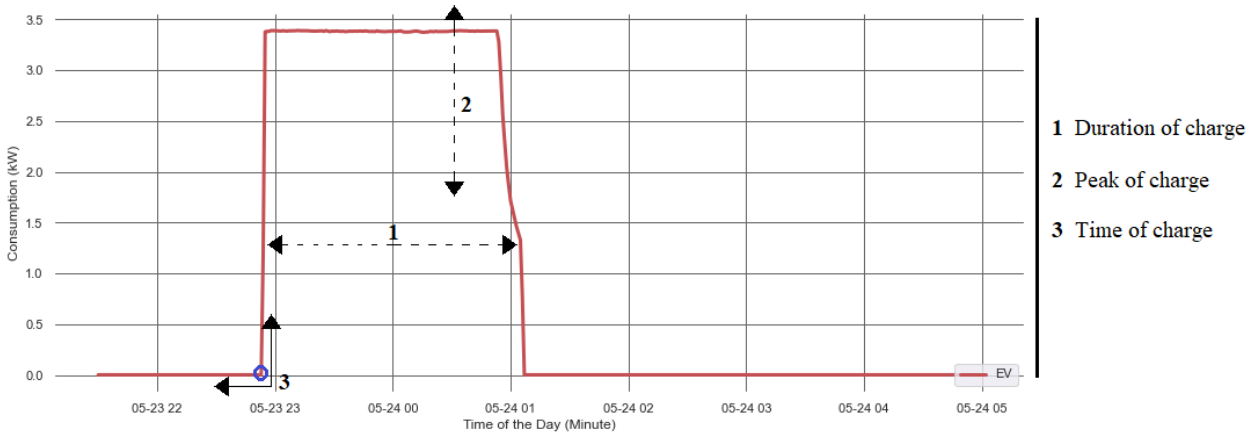
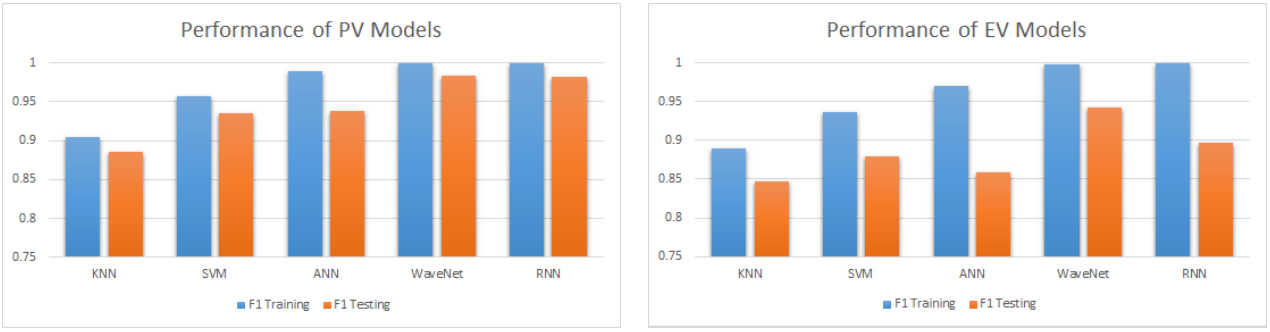


Fig. 3.4 Characteristics that define the EV charging signal

To provide a comparison between the performance of the different learning models, Fig. 3.5 shows the F-score values achieved by the four learning models in detecting both PV generation and EV charging using varying model architectures. As shown in Fig. 3.5 (a) and (b), all the tested robust models achieved relatively high performance on detecting PV generation when compared to detecting EV charging. Nevertheless, due to the temporal dependency of the targets, our WaveNet inspired model and BRNN outperformed all other models. As shown in Fig. 3.5 (a), our WaveNet inspired model achieved a 98.3% F-score result in detecting PV generation. In addition, on the target of detecting EV charging during the consumption day, Fig. 3.5 (b) shows that our WaveNet inspired model achieved a 94.3% F-score, making the WaveNet models our choice for both the PV and EV ownership inference sub-models.



(a) (b)

Fig. 3.5 F-score evaluation results of the different learning models on appliance ownership inference targets: (a) performance of PV models, (b) performance of EV models.

To support the aforementioned inference models analysis, Fig. 3.6 (a) and (b) present the Receiver Operating Characteristics (ROC) evaluation of the five models on both the PV and EV targets. Fig. 3.6 (a) and (b) show that our WaveNet model and BRNN model achieved consistently high Area Under the Curve (AUC) on both the PV and EV targets. Fig. 3.6 also shows that our WaveNet model was the top-performing model on both the PV generation and EV charging targets by 0.996 and 0.979 AUC score (AUC=1 is the perfect classification score) on both targets respectively.

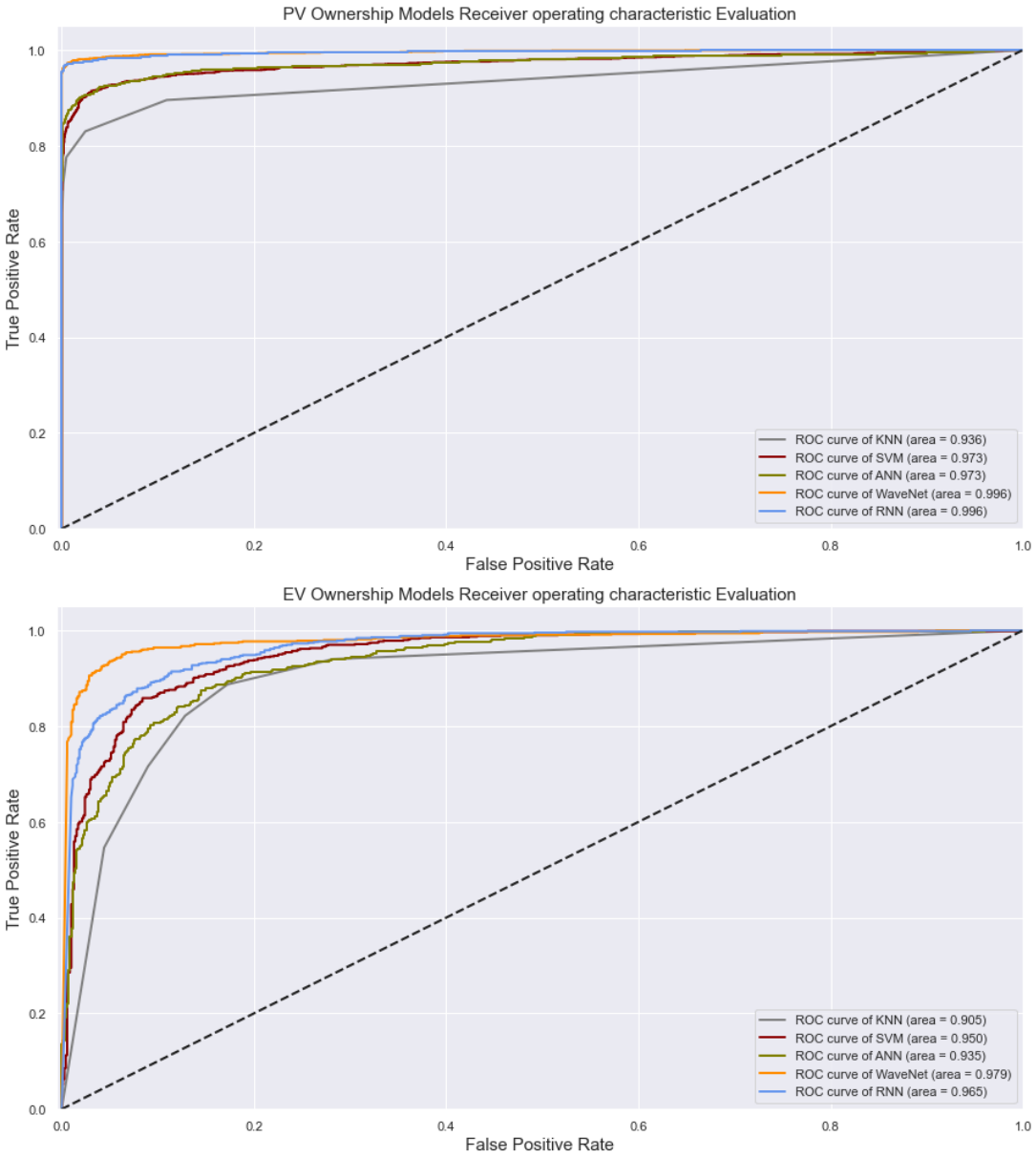
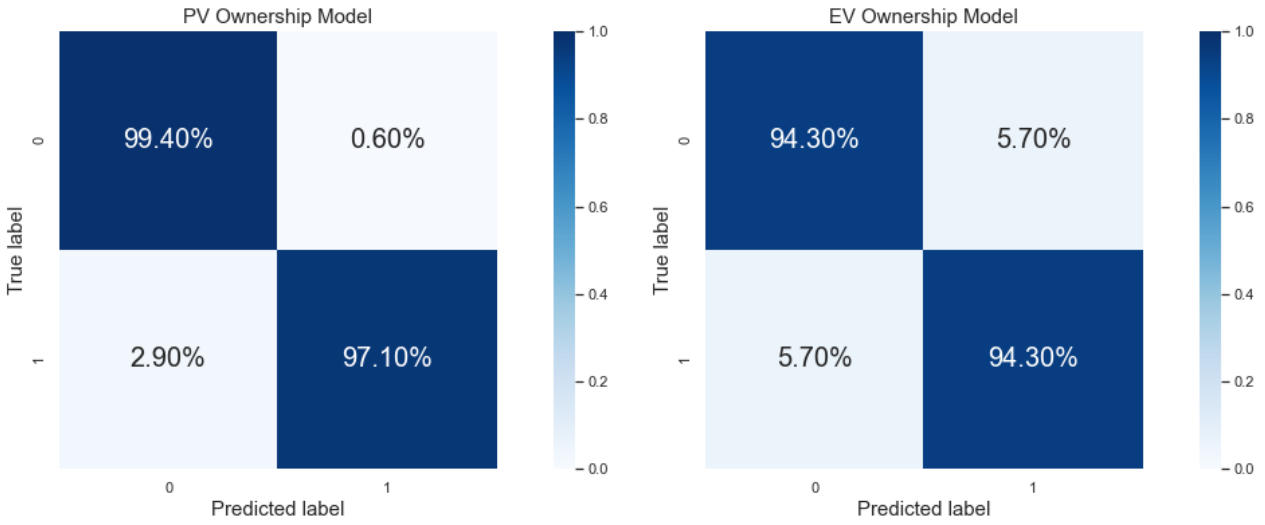


Fig. 3.6 Receiver Operating Characteristics evaluation of the different learning models on appliance ownership inference targets: (a) performance of PV models, (b) performance of EV models.

Fig. 3.7 (a) and (b) show the confusion matrix for our final WaveNet sub-models on both the PV and EV targets respectively. On the PV target, our WaveNet PV sub-model correctly classified 1,813 positive test samples which represent 97.1% of the total true positive test samples, and correctly classified 1,856 negative test samples which represent 99.4% of the total true negative test samples, as can be inferred from Fig. 3.7 (a). On the other hand, our WaveNet EV sub-model correctly classified 661 of both positive and negative test samples which represents 94.3% of

test samples of both classes, as shown in Fig. 3.7 (b). Notice that the data samples have been balanced using the RUS sampling technique as discussed in section 3.6.



(a) (b)

Fig. 3.7 Confusion matrix result for the final appliance ownership inference sub-models: (a) PV sub-model, (b) EV sub-model

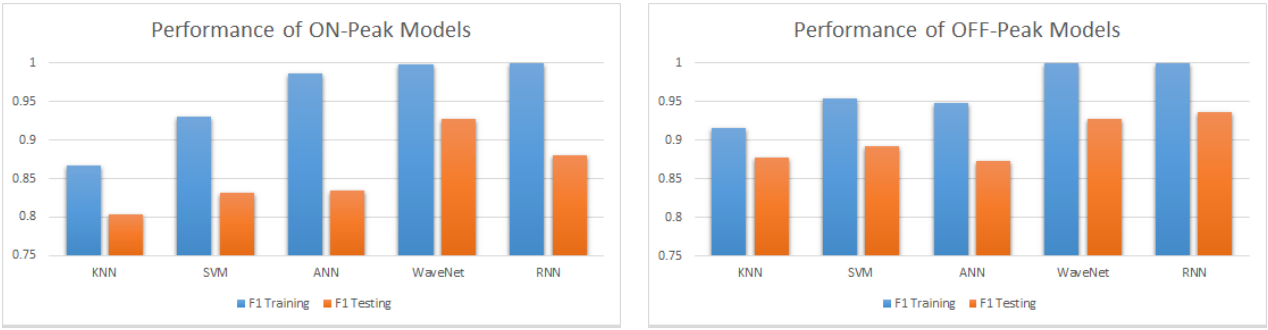
Using these inference sub-models for monitoring the daily consumption of each house in a specific area for several days allows for very accurate PV and EV penetration level estimation as the accuracy would increase by averaging the results over those several days.

3.7.2 Appliance Time-of-Use Inference Results

The consumption days that included EV charging from the Pecan Street database [167] have been labelled for 'OFF-Peak' and 'ON-Peak' charging as our two time-bins (M=2) for testing our appliance ToU inference framework. To provide flexibility in model selection and improve the overall inference framework performance, a sub-model was trained for each time-bin. After the EV sub-model classifies a consumption day as 'included EV charging', the consumption day is fed to the ToU inference sub-models to determine the time of charge. However, to increase the robustness of the overall inference framework and reduce error propagation, ToU sub-models have been trained and tested against days that do not include EV charging.

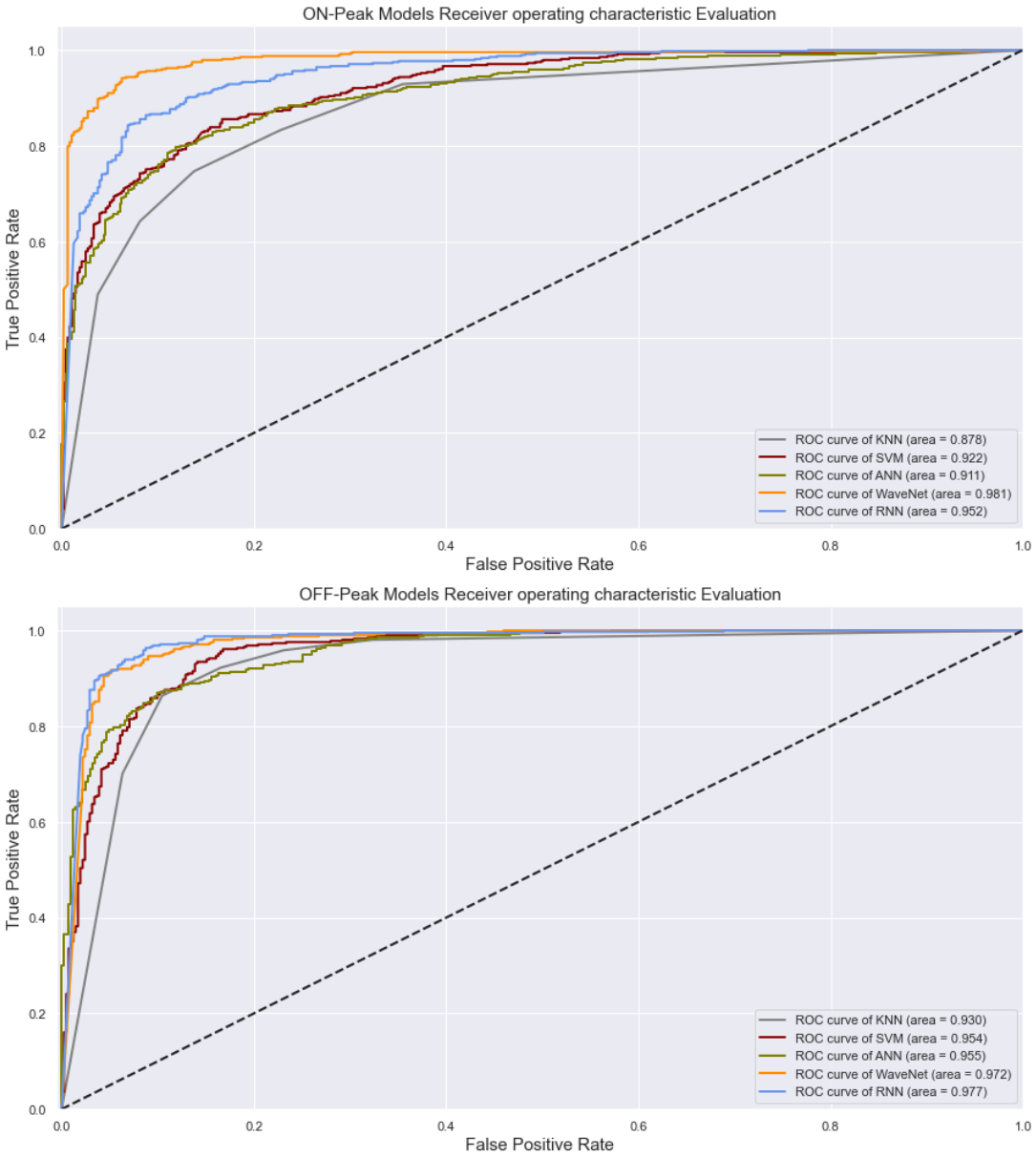
As the complexity of detecting the ToU of EV charging is indeed more than detecting EV charging during the day, the learning models have been modified to

account for the complexity increase. Fig. 3.8 (a) and (b) show the F-score result achieved by the different learning models on detecting 'ON-Peak' and 'OFF-Peak' EV charging, respectively. As shown in Fig. 3.8 (a), while all the tested learning models achieved over 80% F-score on the 'ON-Peak' EV charging target, our 'ON-Peak' WaveNet model achieved 92.7% F-score outperforming other learning models. On the 'OFF-Peak' EV charging target, we selected the BRNN model as our 'OFF-Peak' sub-model since it achieved the best F-score with 93.7%. Fig. 3.9 (a) and (b) present the ROC evaluation of the different learning models on both the 'ON-Peak' and 'OFF-Peak' EV charging targets. As shown in Fig. 3.9 (a), our WaveNet 'ON-Peak' sub-model achieved a 0.981 AUC score. At the same time, our BRNN 'OFF-Peak' sub-model achieved a 0.977 AUC score, as shown in Fig. 3.9 (b).



(a) (b)

Fig. 3.8 F-score evaluation results of the different learning models on appliance time-of-use inference targets: (a) performance of ON-Peak models, (b) performance of OFF-Peak models.



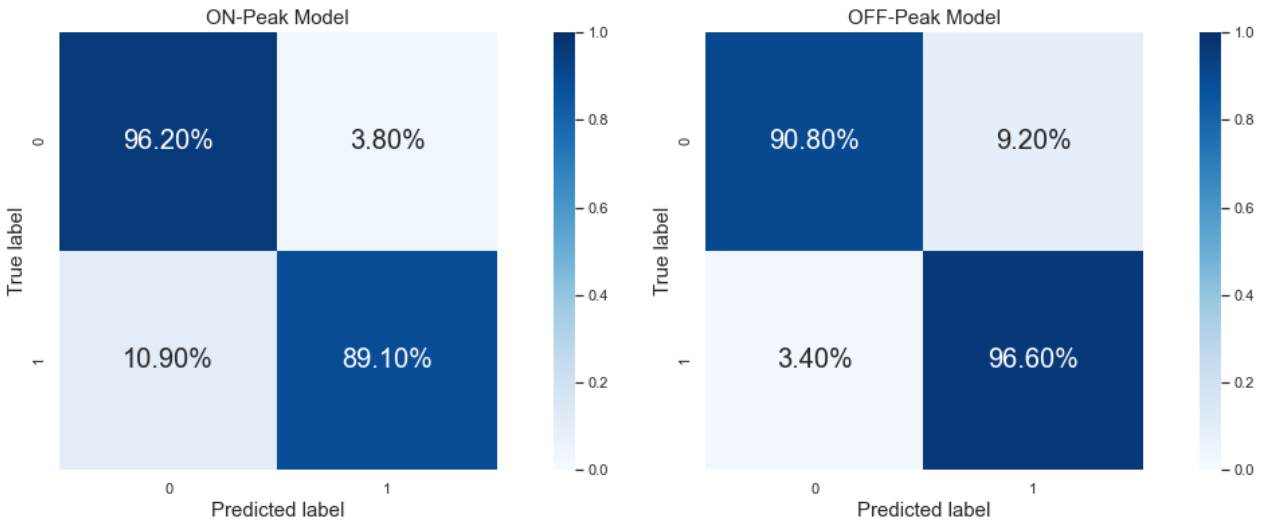
(a)

(b)

Fig. 3.9 Receiver Operating Characteristics evaluation of the different learning models on appliance time-of-use inference targets: (a) performance of ON-Peak models, (b) performance of OFF-Peak models.

The confusion matrix results of both final sub-models are shown in Fig. 3.10 (a) and (b). On the 'ON-Peak' EV charging target, our WaveNet 'ON-Peak' sub-model correctly classified 426 positive 'ON-Peak' EV charging samples, which represent 89.1% of the total true positive samples. On the other hand, the same sub-model classified 461 negative samples correctly representing 96.2% of the true negative samples. On the 'OFF-Peak' EV charging target, our 'OFF-Peak' BRNN sub-model correctly classified 397 positive samples and 373 negative samples which

represent 96.6% of the true positive samples and 90.8% of the true negative samples, respectively, as shown in Fig. 3.10 (b).



(a) (b)

Fig. 3.10 Confusion matrix result for the final appliance time-of-use inference sub-models: (a) PV sub-model, (b) EV sub-model

2.6 Summary

In this chapter, our framework for penetration level estimation and ToU inference was explained. Additionally, we discussed the data required for our model training and evaluation. After that, we discussed the details of both our appliance ownership and penetration level estimation models and ToU inference models. Moreover, the chosen deep learning models have been covered as well as the data balancing techniques used in this research. Finally, the evaluation of the presented framework on real unseen data was discussed.

Chapter 4

Behind-the-Meter Solar Generation Estimation

4.1 Preamble

In this chapter, we discuss our behind-the-meter solar generation inference approaches and their evaluation on unseen data. We explore two different approaches for this target. The approaches make use of our penetration level and appliance ownership estimation sub-models to identify consumption days that include PV generation from those which does not, to overcome the common assumption in other state-the-art behind-the-meter solar generation approaches. Both approaches introduced in this research require only the low-resolution net metering data as their inputs. The approaches utilize state-of-the-art supervised deep-learning models that have been trained and evaluated on real consumption data to address the problem of behind-the-meter solar generation estimation. The visibility of residential solar generation obtained from applying these approaches facilitates efficient distribution power system planning, accurate net load forecasting and competent demand response programs.

4.2 Behind the Meter Solar Generation Estimation Approaches

The goal here is to estimate the power generated from a residential PV system that is hidden behind the main smart meter and not being monitored by a separate smart meter. In other words, given the time-series of aggregated smart meter power measurement for the daily consumption of house i on day n as $P_{i,n}(t)$ ($t \in [1,1440]$), where t is in minutes, and the predicted $y_{i,n,PV}$ binary variable from our appliance ownership sub-model the objective is to recognize the PV generation signature

during the day and estimate $P_{i,n,PV}(t)$ using a regression model. Where $P_{i,n,PV}(t)$ is the time-series of the power generated using the hidden PV system in the house *i* on day *n*. We achieve this objective using two approaches:

- Full-Day Prediction Approach

In this approach, the model will take full-day net smart meter data as its input and predict the PV generation as follows:

$$P_{i,n,PV}(t) = \begin{cases} \widehat{P_{\iota,n,PV}(t)}, & 4{:}00\ am \le t \le 9{:}00\ pm \\ 0, & t < 4{:}00\ am\ or\ t > 9{:}00\ pm \end{cases} \quad (4.1)$$

Where $\widehat{P_{\iota,n,PV}(t)}$ is the predicted PV generation during the day time by the regression model using $P_{i,n}(t)$ of the day time as its only input. On the night time, PV generation prediction is limited to 0 as the generation of the photovoltaic cells in solar panels require sunlight to generate electricity. The day time is chosen to be the time between 4:00 am and 9:00 pm (1021 minute) to account for the whole year at a wide range of locations.

- 1021 Prediction per Day Approach

In this approach, the model will perform a minute-by-minute PV generation prediction using the last five minutes of predicted PV generation and five minutes of smart meter data as its input. In other words, to predict $\widehat{P_{\iota,n,PV}(t)}$ it requires the following as its input:

Table 4.1 1021 Prediction per day approach input

$P_{i,n}(t-2)$	$P_{i,n}(t-1)$	$P_{i,n}(t)$	$P_{i,n}(t+1)$	$P_{i,n}(t+2)$
$\widehat{P_{\iota,n,PV}(t-5)}$	$\widehat{P_{\iota,n,PV}(t-4)}$	$\widehat{P_{\iota,n,PV}(t-3)}$	$\widehat{P_{\iota,n,PV}(t-2)}$	$\widehat{P_{\iota,n,PV}(t-1)}$

The model utilizes the previous 5 minutes of PV generation predictions to estimate the next minute PV generation. This type of regression model is called an auto-regression model. Such models are used commonly as an effective solution for time-series forecasting problems.

4.3 Auto-Regression Model

Auto-regressive models, in general, predict the future behaviour of a process based on its past behaviour. Auto-regressive models are commonly used when there is a strong correlation between the past values and the future values of the process in question. To determine if the problem of behind-the-meter solar generation estimation fits this profile, we performed an auto-correlation analysis between the solar PV generation data and its past values at different time shifts, see Table 4.2. Additionally, we performed the correlation analysis between solar generation data and the net smart meter data shifted for both future and past time instants, see Table 4.3 and Table 4.4.

Table 4.2 Auto-correlation analysis between solar generation data and its past values at different time shifts

	P(i,n,PV) (t)	P(i,n,PV) (t-1)	P(i,n,PV) (t-2)	P(i,n,PV) (t-3)	P(i,n,PV) (t-4)	P(i,n,PV) (t-5)	P(i,n,PV) (t-6)	P(i,n,PV) (t-7)	P(i,n,PV) (t-8)	P(i,n,PV) (t-9)	P(i,n,PV) (t-10)
P(i,n,PV) (t)	1	0.975239	0.954386	0.942048	0.933344	0.926313	0.920335	0.915469	0.911727	0.908337	0.90535
P(i,n,PV) (t-1)	0.975239	1	0.975239	0.954385	0.942048	0.933344	0.926313	0.920335	0.915468	0.911727	0.908337
P(i,n,PV) (t-2)	0.954386	0.975239	1	0.975239	0.954385	0.942048	0.933344	0.926312	0.920334	0.915468	0.911727
P(i,n,PV) (t-3)	0.942048	0.954385	0.975239	1	0.975239	0.954385	0.942047	0.933344	0.926312	0.920334	0.915468
P(i,n,PV) (t-4)	0.933344	0.942048	0.954385	0.975239	1	0.975239	0.954385	0.942047	0.933343	0.926312	0.920334
P(i,n,PV) (t-5)	0.926313	0.933344	0.942048	0.954385	0.975239	1	0.975239	0.954385	0.942047	0.933343	0.926312
P(i,n,PV) (t-6)	0.920335	0.926313	0.933344	0.942047	0.954385	0.975239	1	0.975239	0.954385	0.942047	0.933343
P(i,n,PV) (t-7)	0.915469	0.920335	0.926312	0.933344	0.942047	0.954385	0.975239	1	0.975239	0.954385	0.942047
P(i,n,PV) (t-8)	0.911727	0.915468	0.920334	0.926312	0.933343	0.942047	0.954385	0.975239	1	0.975239	0.954384
P(i,n,PV) (t-9)	0.908337	0.911727	0.915468	0.920334	0.926312	0.933343	0.942047	0.954385	0.975239	1	0.975239
P(i,n,PV) (t-10)	0.90535	0.908337	0.911727	0.915468	0.920334	0.926312	0.933343	0.942047	0.954384	0.975239	1

Table 4.3 Correlation analysis between solar generation data and preceding net meter data at different time shifts

	P(i,n,PV) (t)	P(i,n) (t)	P(i,n) (t-1)	P(i,n) (t-2)	P(i,n) (t-3)	P(i,n) (t-4)	P(i,n) (t-5)	P(i,n) (t-6)	P(i,n) (t-7)	P(i,n) (t-8)	P(i,n) (t-9)
P(i,n,PV) (t)	1	0.948818	0.927848	0.912916	0.903941	0.897408	0.892097	0.887564	0.883956	0.881258	0.878635
P(i,n) (t)	0.948818	1	0.980421	0.963389	0.952893	0.945353	0.939175	0.933865	0.92925	0.925287	0.921525
P(i,n) (t-1)	0.927848	0.980421	1	0.980421	0.963389	0.952893	0.945353	0.939175	0.933865	0.92925	0.925287
P(i,n) (t-2)	0.912916	0.963389	0.980421	1	0.980421	0.963389	0.952893	0.945353	0.939175	0.933865	0.92925
P(i,n) (t-3)	0.903941	0.952893	0.963389	0.980421	1	0.980421	0.963389	0.952893	0.945353	0.939175	0.933865
P(i,n) (t-4)	0.897408	0.945353	0.952893	0.963389	0.980421	1	0.980421	0.963389	0.952893	0.945353	0.939175
P(i,n) (t-5)	0.892097	0.939175	0.945353	0.952893	0.963389	0.980421	1	0.980421	0.963389	0.952893	0.945353
P(i,n) (t-6)	0.887564	0.933865	0.939175	0.945353	0.952893	0.963389	0.980421	1	0.980421	0.963389	0.952893
P(i,n) (t-7)	0.883956	0.92925	0.933865	0.939175	0.945353	0.952893	0.963389	0.980421	1	0.980421	0.963389
P(i,n) (t-8)	0.881258	0.925287	0.92925	0.933865	0.939175	0.945353	0.952893	0.963389	0.980421	1	0.980421
P(i,n) (t-9)	0.878635	0.921525	0.925287	0.92925	0.933865	0.939175	0.945353	0.952893	0.963389	0.980421	1

Table 4.4 Correlation analysis between solar generation data and succeeding net meter data at different time shifts

	P(i,n,PV) (t)	P(i,n) (t)	P(i,n) (t+1)	P(i,n) (t+2)	P(i,n) (t+3)	P(i,n) (t+4)	P(i,n) (t+5)	P(i,n) (t+6)	P(i,n) (t+7)	P(i,n) (t+8)	P(i,n) (t+9)
P(i,n,PV) (t)	1	0.951858	0.936481	0.915433	0.903146	0.89465	0.887895	0.882278	0.877539	0.873888	0.87065
P(i,n) (t)	0.951858	1	0.978502	0.959799	0.948364	0.940118	0.933338	0.927512	0.922494	0.918261	0.914268
P(i,n) (t+1)	0.936481	0.978502	1	0.978416	0.959762	0.948328	0.940083	0.933304	0.927479	0.922462	0.918231
P(i,n) (t+2)	0.915433	0.959799	0.978416	1	0.978456	0.959799	0.948363	0.940117	0.933338	0.927512	0.922494
P(i,n) (t+3)	0.903146	0.948364	0.959762	0.978456	1	0.978456	0.959799	0.948363	0.940117	0.933338	0.927512
P(i,n) (t+4)	0.89465	0.940118	0.948328	0.959799	0.978456	1	0.978456	0.959799	0.948363	0.940117	0.933338

P(i,n) (t+5)	0.887895	0.933338	0.940083	0.948363	0.959799	0.978456	1	0.978456	0.959799	0.948363	0.940117
P(i,n) (t+6)	0.882278	0.927512	0.933304	0.940117	0.948363	0.959799	0.978456	1	0.978456	0.959799	0.948363
P(i,n) (t+7)	0.877539	0.922494	0.927479	0.933338	0.940117	0.948363	0.959799	0.978456	1	0.978456	0.959799
P(i,n) (t+8)	0.873888	0.918261	0.922462	0.927512	0.933338	0.940117	0.948363	0.959799	0.978456	1	0.978456
P(i,n) (t+9)	0.87065	0.914268	0.918231	0.922494	0.927512	0.933338	0.940117	0.948363	0.959799	0.978456	1

As shown in Table 4.2, there is a very high correlation between the solar data and its past values presenting a possibility for building a good auto-regression model. Additionally, there is also a significant correlation between solar data and the early shifted time instants smart meter data values, in both the future and the past, as shown in Table 4.3 and Table 4.4. As a result of these preliminary analyses, the 1021 prediction per day auto-regressive approach, as explained in section 4.2, was explored.

4.4 Data Requirements

To train and evaluate potential regression models for the suggested approaches, one must find a dataset that fulfills the following requirements:

1) Daily net smart meter data containing PV generation of several residential houses over a long period of time to cover different weather conditions as well as different consumption patterns.

2) Metered daily generation data of the solar PV panels connected to the net meter for each house along the same period of the available net meter data.

The Pecan Street database [167] presents a suitable dataset for training and evaluating potential regression models. After preprocessing and cleaning the data and dealing with the outliers (measurement errors), the dataset used for training and evaluating our behind-the-meter solar generation models consisted of 9,350 consumption days of net smart meter data, as well as solar PV generation data, of 33 different houses, each house is metered for more than half a year.

The dataset has been restructured for training and evaluating potential regression models for the two explored behind-the-meter solar generation estimation approaches.

4.5 Behind the Meter Solar Generation Estimation Results

The samples of the consumption days that have been predicted to contain PV generation by our PV appliance ownership model, which achieved a 98.3% F-score result, would then be restructured for our two approaches to estimate the solar generation during the day. To compare both approaches we have trained various WaveNet based regression models and chose the best performing model for each approach. The choice of WaveNet based models is due to their unique architecture that enables efficient learning of temporal dependencies. In addition, the WaveNet based models achieved significantly high results on our appliance ownership targets as well as ToU inference targets.

After training the chosen learning models for both approaches, we evaluated both approaches on the same testing data in order to justly compare their performances. The mean square error (MSE) and the coefficient of determination, also known as R-squared (R^2), were chosen to be our error metrics in the following analysis of the two approaches as they are commonly used in related literature.

The MSE can be defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \tag{4.2}$$

Where *n* is the number of samples, *Yi* is the true value and $\widehat{Y}_i$ is the predicted value.

The R^2 is a statistical measure that represents the proportion of the dependent variable variance that is explained by an independent variable in a regression model

$$R^2 = 1 - \frac{RSS}{TSS} \tag{4.3}$$

Where *RSS* is the residual sum of squares and *TSS* is the total sum of squares and are defined as follow:

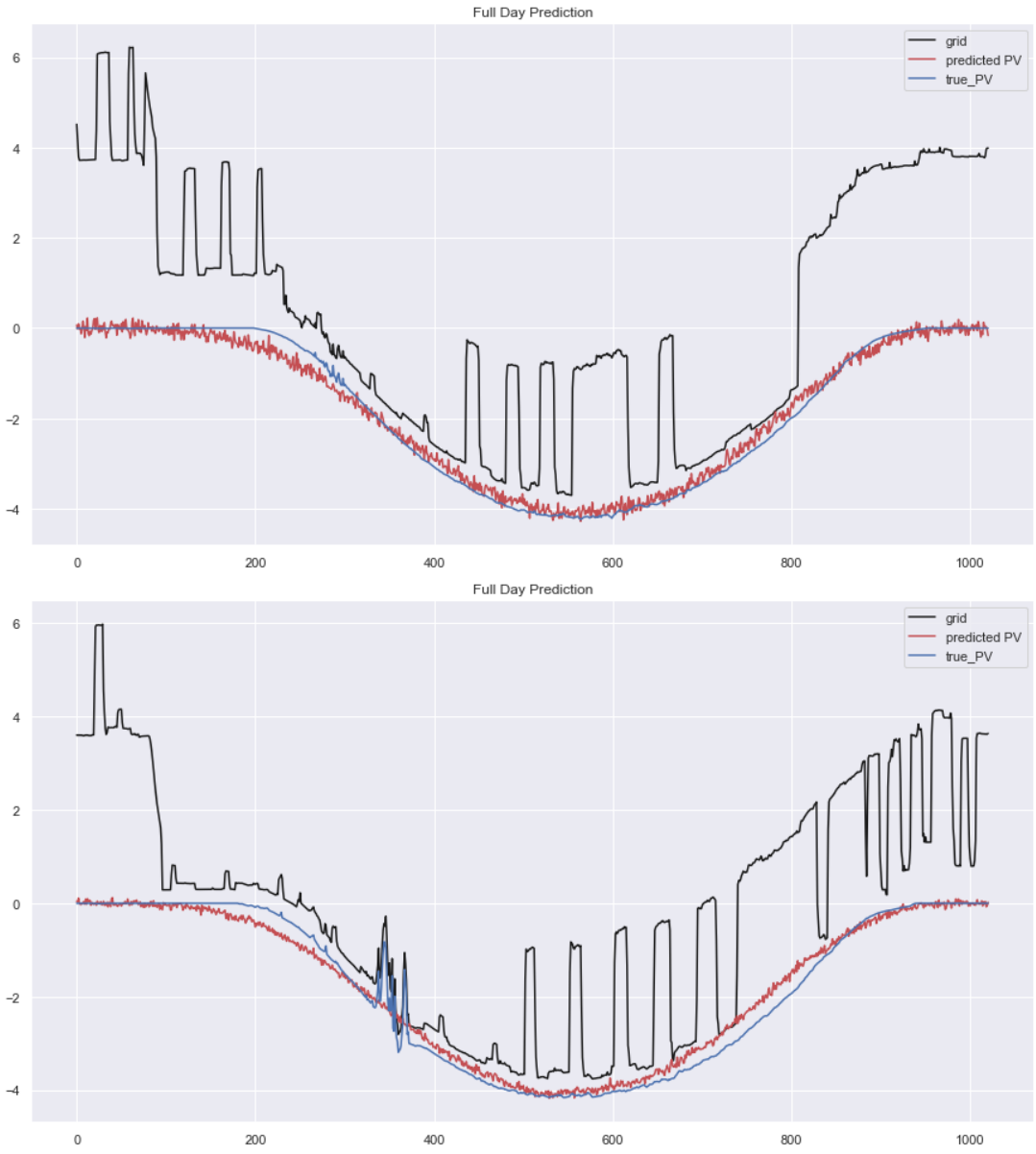
$$RSS = \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 \quad (4.4)$$

$$TSS = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 \quad (4.5)$$

Where $\bar{Y}$ is the mean of the observed data.

Fig. 4.1 (a) and (b) show the estimated PV generation for two test consumption days that achieved low MSE on the full-day prediction approach.

As shown in Fig. 4.1, the full-day prediction approach model was able to estimate the solar generation accurately regardless of the large variance in the consumption signal due to other electrical appliances usage.

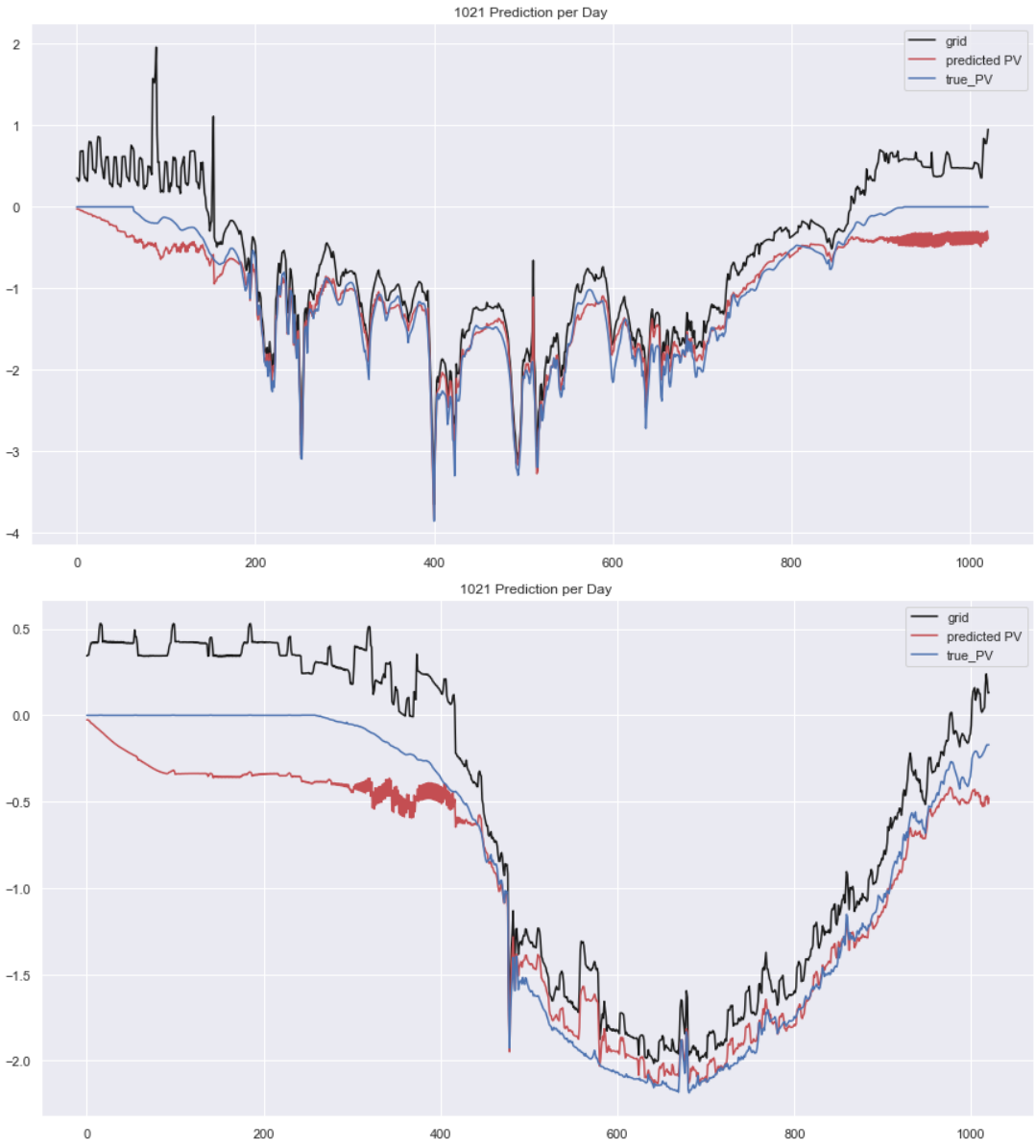


(a)

(b)

Fig. 4.1 Full-day prediction approach low MSE predicted days: (a) MSE= 0.043, (b) MSE= 0.063

On the other hand, Fig. 4.2 shows the result of estimating the PV generation of two test days that achieved low MSE using the 1021 prediction per day approach.



(a)

(b)

Fig. 4.2 1021 prediction per day approach low MSE predicted days: (a) MSE= 0.041, (b) MSE= 0.046

As shown in Fig. 4.2, the 1021 prediction per day approach model achieved a very low MSE on estimating the intermittent solar generation of cloudy days as the one shown in Fig. 4.2 (a). Cloudy days solar generation estimation represents a tough problem as the solar generation is affected by the clouds' movement as it limits the amount of sunlight reaching the solar panels resulting in a high variance intermittent solar generation.

Fig. 4.3 (a) and (b) show two different days with an average MSE score for the full-day prediction approach model. As shown in the figure, the model

performance is degraded with higher variance solar generation, i.e. cloudy days, as contrary to the 1021 prediction per day approach model.

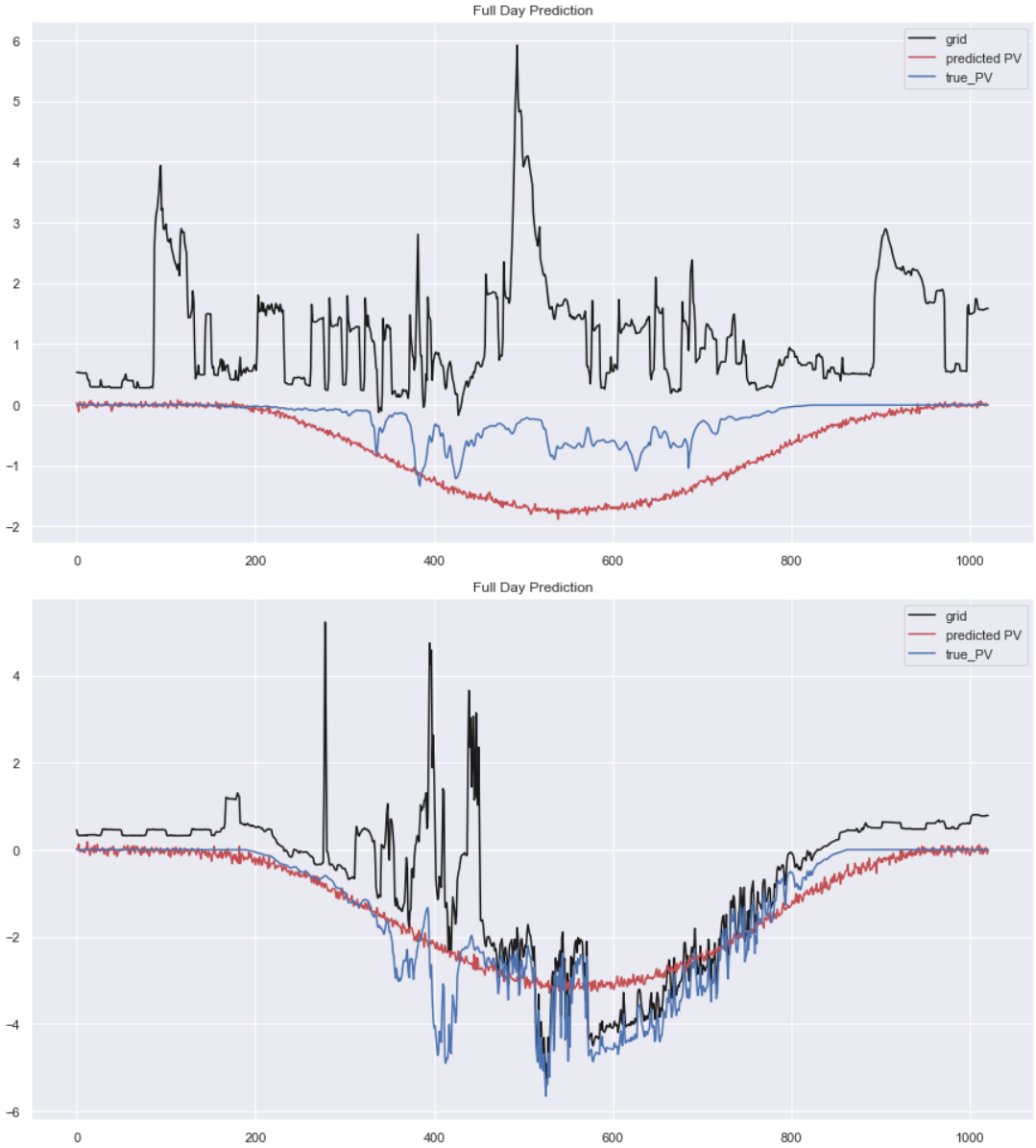
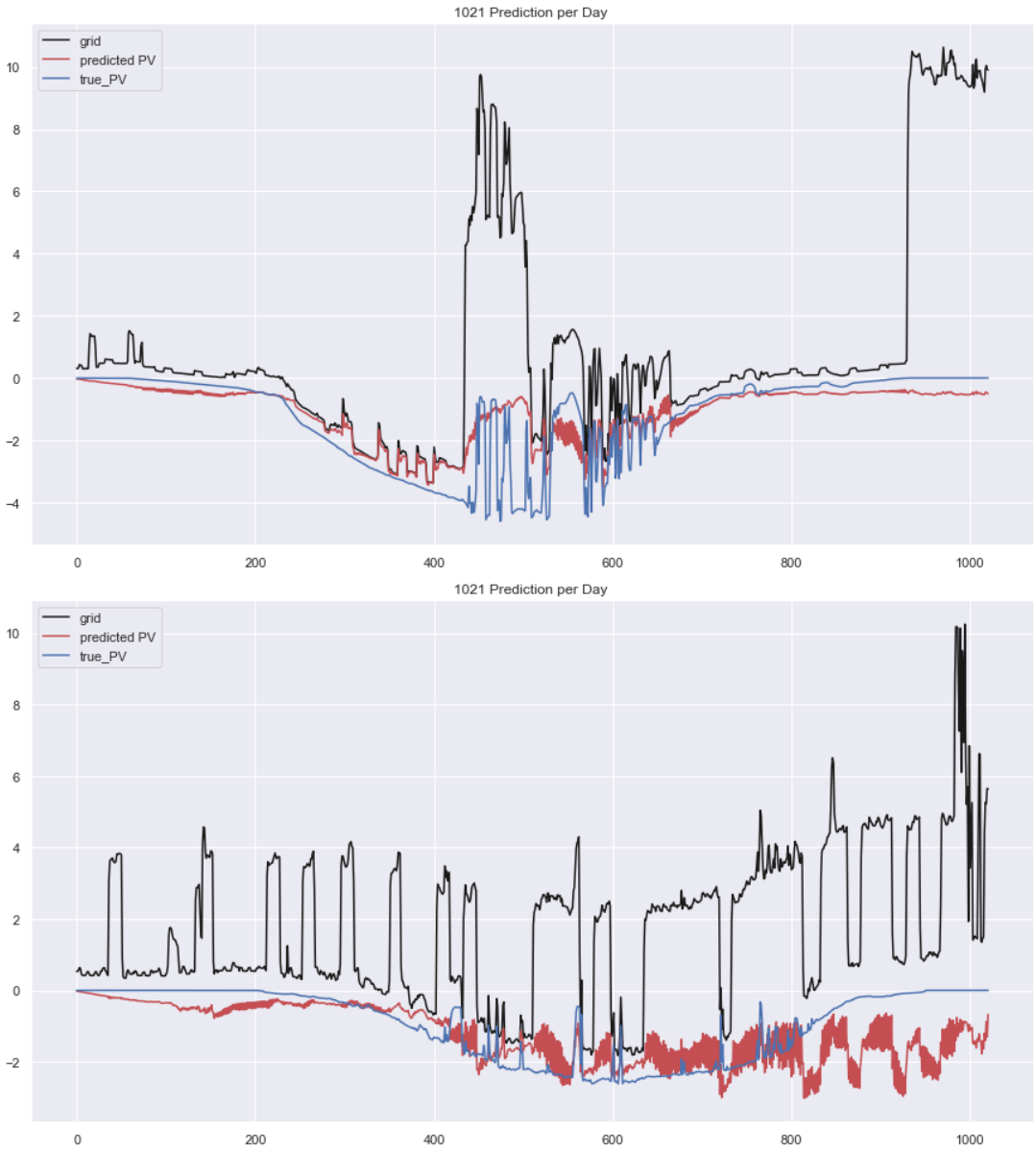


Fig. 4.3 Full-day prediction approach average MSE predicted days: (a) MSE= 0.439, (b) MSE= 0.44

Conversely, Fig. 4.4 (a) and (b) present two average MSE scoring consumption days for which the PV generation has been estimated using the 1021 prediction per day approach model. Due to high variance in the electric consumption as a result of the different electrical appliances usage in those two days, the estimations of the 1021 prediction per day approach model is not as accurate as it were on the two day shown in Fig 4.2.



(a)

(b)

Fig. 4.4 1021 prediction per day approach average MSE predicted days: (a) MSE= 0.673, (b) MSE= 0.675

Nevertheless, both approaches achieved a low average MSE score on the unseen test data. Table 4.5 shows the average normalized MSE and R^2 result of both approaches.

Table 4.5 MSE and R^2 scores of the behind-the-meter solar generation estimation suggested approaches

	MSE	R^2
Full-day prediction approach	0.00199	0.98
1021 prediction per day approach	0.00014	0.72

2.7 Summary

In this chapter, we discussed our suggested approaches to the problem of behind-the-meter solar generation estimation. We also presented the analysis that led to exploring the 1021 prediction per day approach. After that, the characteristics of the data required to train and evaluate our suggested approaches potential regression models have been discussed. Finally, we present the result of our suggested approaches for estimating the behind-the-meter solar generation on unseen real data. In conclusion, although both approaches achieved a low average MSE score, each one had it is own advantages and disadvantages. While the full-day approach achieved a better result on estimating the hidden PV generation regardless of the coexistence of high variance electrical consumption appliances which degraded the performance of the 1021 prediction per day approach, the 1021 prediction per day approach achieved a significantly better result on estimated the solar generation on cloudy days when compared to the full-day prediction approach.

Chapter 5

Conclusion

In this study we introduced a framework for estimating the penetration level of PV generation and EVs in a monitored area; in addition to predicting EVs time of charge given aggregated smart meter data as an application of our framework for flexible appliance time-of-use inference as well as estimating the behind-the-meter solar generation. The framework obtains its estimations using existing smart meter low-resolution data as its only input without the need for any on-site training or appliance ownership information. The framework infers the appliance ownership information in its early sub-models and uses the information in its consequent penetration level estimation, flexible appliances' ToU inference and behind-the-meter solar generation estimation.

The framework exploits various sub-models for every sub-target, breaking down the inference question into smaller binary classification inquires, in addition to the regression sub-target, each sub-model is designed and customized for its specific sub-target and trained to mitigate error propagation from precedent sub-models. The sub-models are built of WaveNet inspired learning models and a Bidirectional Recurrent Neural Network model. Upon receiving new daily smart meter consumption data, the framework's early sub-models infer whether this daily consumption included PV generation and/or EV charging as part of its appliance ownership inference. Subsequently, this inferred information is used for penetration level estimation, enabling accurate ToU inference and estimating the generated solar power behind the existing smart meters. After that, the framework's later sub-models infer the ToU vector of flexible appliances which can be used in demand response (DR) and demand-side-management programs. In addition, the framework's behind-the-meter solar generation estimation model can be utilized for improving the power procurement plans of distribution power companies. The framework has been tested on real smart meter data obtained from the Pecan Street database [167] and achieved

an F-score classification result between 93% to 98% across all its sub-models which proves the feasibility and scalability of our approach.

Finally, the work presented in this study can be extended in various ways. First, further studies can be done on a ToU enabled DR that takes into account the flexible appliance use history of customers to minimize the peak-to-average ratio (PAR) while maintaining a high level of comfort for end-customers. Alternatively, a forecasting framework that utilizes appliance penetration level estimation for improving its forecasting capabilities can be investigated. Nevertheless, even though both our presented approaches to tackle the target of behind the meter solar estimation achieved low mean square error (MSE), each approach showed advantages over the other. Therefore, more studies are required to determine an optimum solution to this problem.

References

[1] United Nations Framework Convention on Climate Change, unfccc.int/2860.php.

[2] N. Chang, D. Baek and J. Hong, "Power consumption characterization, modelling and estimation of electric vehicles," 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, 2014, pp. 175-182, doi: 10.1109/ICCAD.2014.7001349.

[3] V. R. J. H. Timmers and P. A. J. Achten, "Non-exhaust PM emissions from electric vehicles", Atmos. Environ, vol. 134, pp. 10-17, 2016.

[4] R. Vidhi and P. Shrivastava, "A review of electric vehicle lifecycle emissions and policy recommendations to increase EV penetration in India", Energies, vol. 11, no. 3, pp. 1-16, 2018.

[5] F. G. Praticò, P. G. Briante and G. Speranza, "Acoustic Impact of Electric Vehicles," 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 2020, pp. 7-12, doi: 10.1109/MELECON48756.2020.9140669.

[6] P. Balcombe, D. Rigby and A. Azapagic, "Motivations and barriers associated with adopting microgeneration energy technologies in the UK", Renewable and Sustainable Energy Reviews, vol. 22, no. C, pp. 655-666, 2013.

[7] J. Leenheer, M. de Nooij, and O. Sheikh, "Own power: Motives of having electricity without the energy company," Energy Policy, vol. 39, pp. 5621-5629, Sept. 2011.

[8] M. Graziano and K. Gillingham, "Spatial patterns of solar photovoltaic system adoption: The influence of neighbors and the built environment", J. Econ. Geogr., vol. 15, no. 4, pp. 815-839, 2015.

[9] The Solar Energy Industries Association (SEIA)/Wood Mackenzie Power & Renewables U.S. Solar Market Insight, “US Solar Market Insight Report 2020 Q2”, The Solar Energy Industries Association (SEIA), 2020. [Online]. Available:

https://seia.org/research-resources/solar-market-insight-report-2020-q2. [Accessed: Nov. 19, 2020].

[10] L. Strupeit and A. Palm, "Overcoming barriers to renewable energy diffusion: business models for customer-sited solar photovoltaics in Japan Germany and the United States", Journal of Cleaner Production, 2015, [online] Available: http://dx.doi.org/10.1016/j.jclepro.2015.06.120.

[11] T. Trigg, P. Telleen, R. Boyd, F. Cuenot, D. D'Ambrosio, R. Gaghen, J. Gagné, A. Hardcastle, D. Houssin, A. Jones et al., "Global EV Outlook: Understanding the Electric Vehicle Landscape to 2020", Int. Energy Agency, pp. 1-40, 2013

[12] The International Energy Agency (IEA), “Global EV Outlook 2020”, The International Energy Agency (IEA), 2020. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2020. [Accessed: Nov. 25, 2020].

[13] S. Rahman and G. B. Shrestha, "An investigation into the impact of electric vehicle load on the electric utility distribution system", IEEE Trans. Power Del., vol. 8, no. 2, pp. 591-597, Apr. 1993.

[14] G. A. Putrus, P. Suwanapingkarl, D. Johnston, E. C. Bentley and M. Narayana, "Impact of electric vehicles on power distribution networks", Proc. IEEE Veh. Power Propulsion Conf. (VPPC), pp. 827-831, 2009.

[15] P. S. Moses, M. A. S. Masoum and S. Hajforoosh, "Overloading of distribution transformers in smart grid due to uncoordinated charging of plug-in electric vehicles", Proc. IEEE PES Innov. Smart Grid Technol. (ISGT), pp. 1-6, Jan. 2012.

[16] K. Clement-Nyns, E. Haesen and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid", IEEE Trans. Power Syst., vol. 25, no. 1, pp. 371-380, Feb. 2010.

[17] A. Bilh, K. Naik and R. El-Shatshat, "A Novel Online Charging Algorithm for Electric Vehicles Under Stochastic Net-Load," in IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1787-1799, May 2018, doi: 10.1109/TSG.2016.2599819.

[18] B. Mountain and P. Szuster, "Solar, Solar Everywhere: Opportunities and Challenges for Australia's Rooftop PV Systems," in IEEE Power and Energy

Magazine, vol. 13, no. 4, pp. 53-60, July-Aug. 2015, doi: 10.1109/MPE.2015.2416113.

[19] T. J. Geiles and S. Islam, "Impact of PEV charging and rooftop PV penetration on distribution transformer life," 2013 IEEE Power & Energy Society General Meeting, Vancouver, BC, 2013, pp. 1-5, doi: 10.1109/PESMG.2013.6672211.

[20] J. Peng and L. Lu, "Investigation on the development potential of rooftop PV system in Hong Kong and its environmental benefits", Renew. Sustain. Energy Rev., vol. 27, pp. 149-162, 2013.

[21] H. Sadeghian, M. H. Athari and Z. Wang, "Optimized solar photovoltaic generation in a real local distribution network," 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, 2017, pp. 1-5, doi: 10.1109/ISGT.2017.8086067.

[22] F. Kabir, N. Yu, W. Yao, R. Yang and Y. Zhang, "Estimation of Behind-the-Meter Solar Generation by Integrating Physical with Statistical Models," 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Beijing, China, 2019, pp. 1-6, doi: 10.1109/SmartGridComm.2019.8909743.

[23] H. Shaker, H. Zareipour and D. Wood, "A Data-Driven Approach for Estimating the Power Generation of Invisible Solar Sites," in IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2466-2476, Sept. 2016, doi: 10.1109/TSG.2015.2502140.

[24] H. Shaker, H. Zareipour and D. Wood, "Estimating Power Generation of Invisible Solar Sites Using Publicly Available Data," in IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2456-2465, Sept. 2016, doi: 10.1109/TSG.2016.2533164.

[25] M. Pipattanasomporn, M. Kuzlu, and S. Rahman, ‘‘An algorithm for intelligent home energy management and demand response analysis,’’ IEEE Trans. Smart Grid, vol. 3, no. 4, pp. 2166–217 3, Dec. 2012.

[26] M. Afzalan and F. Jazizadeh, "A Machine Learning Framework to Infer Time-of-Use of Flexible Loads: Resident Behaviour Learning for Demand

Response," in IEEE Access, vol. 8, pp. 111718-111730, 2020, doi: 10.1109/ACCESS.2020.3002155.

[27] F. Rassaei, W. Soh and K. Chua, "Demand Response for Residential Electric Vehicles with Random Usage Patterns in Smart Grids," in IEEE Transactions on Sustainable Energy, vol. 6, no. 4, pp. 1367-1376, Oct. 2015, doe: 10.1109/TSTE.2015.2438037.

[28] P. Palensky and D. Dietrich, “Demand side management: Demand response, intelligent energy systems, and smart loads,” IEEE Trans. Ind. Informat., vol. 7, no. 3, pp. 381–388, Aug. 2011.

[29] V. R. Padullaparthi, V. Sarangan and A. Sivasubramaniam, "sUncover: Estimating the Hidden Behind-the-meter Solar Rooftop and Battery Capacities in Grids," 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2019, pp. 1-5, doi: 10.1109/ISGT.2019.8791573.

[30] E. S. Parizy, H. R. Bahrami and S. Choi, "A Low Complexity and Secure Demand Response Technique for Peak Load Reduction," in IEEE Transaction on Smart Grid, vol. 10, no. 3, pp. 3259-3268, May 2019, doi: 10.1109/TSG.2018.2822729.

[31] K. Clement, E. Haesen and J. Driesen, "Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids", Proc. IEEE/PES Power Syst. Conf. Expo. (PSCE09), pp. 1-7, Mar. 2009.

[32] F. Rassaei, W. S. Soh and K. C. Chua, "A statistical modelling and analysis of residential electric vehicles charging demand in smart grids", Proc. 6th Int. Conf. Innov. Smart Grid Technol. (ISGT15), 2015.

[33] E. L. Karfopoulos and N. D. Hatziargyriou, "A Multi-Agent System for Controlled Charging of a Large Population of Electric Vehicles," in IEEE Transactions on Power Systems, vol. 28, no. 2, pp. 1196-1204, May 2013, doi: 10.1109/TPWRS.2012.2211624.

[34] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," in IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 76-84, February 2011, doi: 10.1109/TCE.2011.5735484.

[35] A. Zoha, A. Gluhak, M. Imran and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey", Sensors, vol. 12, no. 12, pp. 16838-16866, Dec. 2012.

[36] S. M. Tabatabaei, S. Dick and W. Xu, "Toward Non-Intrusive Load Monitoring via Multi-Label Classification," in IEEE Transactions on Smart Grid, vol. 8, no. 1, pp. 26-40, Jan. 2017, doi: 10.1109/TSG.2016.2584581.

[37] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial HMMs with application to energy disaggregation,” in International Conference on Artificial Intelligence and Statistics, 2012, pp. 1472–1482.

[38] Y. Li, Z. Peng, J. Huang, Z. Zhang, and J. H. Son, “Energy disaggregation via hierarchical factorial HMM,” in Proceedings of the 2nd International Workshop on Non-Intrusive Load Monitoring, 2014.

[39] D. Piga, A. Cominola, M. Giuliani, A. Castelletti and A. E. Rizzoli, "Sparse Optimization for Automated Energy End Use Disaggregation," in IEEE Transactions on Control Systems Technology, vol. 24, no. 3, pp. 1044-1051, May 2016, doi: 10.1109/TCST.2015.2476777.

[40] G. C. Koutitas and L. Tassiulas, "Low Cost Disaggregation of Smart Meter Sensor Data," in IEEE Sensors Journal, vol. 16, no. 6, pp. 1665-1673, March15, 2016, doi: 10.1109/JSEN.2015.2501422. \

[41] K. Basu, L. Hawarah, N. Arghira, H. Joumaa and S. Ploix, "A prediction system for home appliance usage", Energy Buildings, vol. 67, pp. 668-679, Dec. 2013.

[42] T. Le, J. Kim and H. Kim, "Classification performance using gated recurrent unit recurrent neural network on energy disaggregation," 2016 International Conference on Machine Learning and Cybernetics (ICMLC), Jeju, 2016, pp. 105-110, doi: 10.1109/ICMLC.2016.7860885.

[43] Y. Liu, X. Wang and W. You, "Non-intrusive load monitoring by voltage–current trajectory enabled transfer learning", IEEE Trans. Smart Grid, vol. 10, no. 5, pp. 5609-5619, Sep. 2019.

[44] C. J. Sarasa-Maestro, R. Dufo-López and J. L. Bernal-Agustín, "Photovoltaic remuneration policies in the European Union", Energy Policy, vol. 55, pp. 317-328, 2013.

[45] C. Dong et al., "Forecasting residential solar photovoltaic deployment in California", Technological Forecasting & Social Change, vol. 117, pp. 251-265, April 2017.

[46] M.C. Claudy, C. Michelsen, A. O'Driscoll and M.R. Mullen, "Consumer awareness in the adoption of microgeneration technologies. An empirical investigation in the Republic of Ireland", Renewable and Sustainable Energy Reviews, vol. 14, pp. 2154-2160, 2010.

[47] A. Palm, "Peer effects in residential solar photovoltaics adoption—A mixed methods study of Swedish users," Energy Res. Social Sci., vol. 26, pp. 1–10, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.erss.2017.01.008

[48] C. L. Kwan, "Influence of local environmental social economic and political variables on the spatial distribution of residential solar PV arrays across the United States", Energy Policy, vol. 47, pp. 332-344, 2012.

[49] J. Palmer, G. Sorda and R. Madlener, "Modelling the diffusion of residential photovoltaic systems in Italy: An agent-based simulations", Technological Forecasting & Social Change, vol. 99, no. 2015, pp. 106-131, 2015.

[50] S. Dharshing, "Household dynamics of technology adoption: A spatial econometric analysis of residential solar photovoltaic (PV) systems in Germany". Energy Research & Social Science, vol. 23, pp.113–124, 2017.

[51] T. Islam and N. Meade, "The impact of attribute preferences on adoption timing: The case of photo-voltaic (PV) solar cells for household electricity generation," Energy Policy, vol. 55, pp. 521–530, 2013.

[52] Carolyn Davidson, Easan Drury, Anthony Lopez, Ryan Elmore and Robert Margolis, "Modelling photovoltaic diffusion: an analysis of geospatial datasets", Environmental Research Letters, vol. 9, no. 7, pp. 074009, 2014.

[53] F. Sossan , L. Nespoli, V. Medici, and M. Paolone , "Unsupervised disaggregation of photovoltaic production from composite power flow

measurements of heterogeneous prosumers," IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 3904 3913, Sep. 2018.

[54] W. Stainsby, D. Zimmerle , and G. P. Duggan, "A method to estimate residential PV generation from net metered load data and system install date," Applied Energy, vol. 267, p. 114895, Jun. 2020.

[55] R. H. M. Zargar and M. H. Yaghmaee Moghaddam, "Development of a Markov-Chain-Based Solar Generation Model for Smart Microgrid Energy Management System," in IEEE Transactions on Sustainable Energy, vol. 11, no. 2, pp. 736-745, April 2020, doi: 10.1109/TSTE.2019.2904436.

[56] K. Li, F. Wang, Z. Mi, M. Fotuhi-Firuzabad, N. Duić and T. Wang, "Capacity and output power estimation approach of individual behind-the-meter distributed photovoltaic system for demand response baseline estimation", Appl. Energy, vol. 253, Nov. 2019.

[57] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, Data-driven probabilistic net load forecasting with high penetration of behind the meter PV," IEEE Transactions on Power Systems, vol. 33, no. 3, pp. 3255 3264, May 2018.

[58] F. Wang, Z. Zhen, Z. Mi, H. Sun, S. Su and G. Yang, "Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting", Energy Buildings, vol. 86, pp. 427-438, Jan. 2015.

[59] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity", Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.

[60] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain", Psychological Review, vol. 65, pp. 386-408, 1958.

[61] D. H. Hubel and T. N. Wiesel, "Receptive fields binocular interaction and functional architecture in the cat's visual cortex", J. Physiology (London), vol. 160, pp. 106-154, 1962.

[62] R. D. Joseph, "Contributions of perceptron theory". Unpublished doctoral dissertation, Cornell University, Ithaca, NY, 1961.

[63] A. G. Ivakhnenko, "The group method of data handling a rival of the method of stochastic approximation", Soviet Automatic Control, vol. 13, no. 3, pp. 43-55, 1968.

[64] A. C. Ivakhnenko, "Polynomial theory of complex systems", IEEE Trans. Syst. Man Cybernetics, vol. SMC-1, pp. 364-378, Oct. 1971.

[65] A. G. Ivakhnenko and V. G. Lapa, Cybernetic Predicting Devices., New York, NY, USA:CCM Information Corporation, 1965.

[66] A.G. Ivakhnenko, Cybernetics and forecasting techniques, New York:American Elsevier, 1967.

[67] S.J. Farlow, “Self-Organizing Methods in Modelling: GMDH Type Algorithms”, New York:Marcel Dekker, 1984.

[68] S. Ikeda, M. Ochiai and Y. Sawaragi, "sequential GMDH algorithm and its application to river flow prediction", IEEE Trans. Syst. Man Cybern., vol. SMC-6, pp. 473-479, July 1976.

[69] A. G. Ivakhnenko and G. A. Ivakhnenko, "The review of problems solvable by algorithms of the group method of data handling (GMDH)", Pattern Recogn. Image Anal., vol. 5, no. 4, pp. 527-535, 1995.

[70] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position", Proceedings of IECE, vol. J62-A/10, pp. 658-665, 1979.

[71] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", Biological Cybernetics, vol. 36, pp. 193-202, 1980.

[72] K. Fukushima, "Artificial vision by multi-layered neural networks: Neocognitron and its advances", Neural Netw., vol. 37, pp. 103-119, 2013.

[73] S. Amari, "A Theory of Adaptive Pattern Classifiers," in IEEE Transactions on Electronic Computers, vol. EC-16, no. 3, pp. 299-307, June 1967, doi: 10.1109/PGEC.1967.264666.

[74] A. E. Bryson, "A gradient method for optimizing multi-stage allocation processes", Proceedings of the Harvard University Symposium on Digital Computers and Their Applications, 1961-April.

[75] S. Director and R. Rohrer, "Automated Network Design-The Frequency-Domain Case," in IEEE Transactions on Circuit Theory, vol. 16, no. 3, pp. 330-337, August 1969, doi: 10.1109/TCT.1969.1082967.

[76] S. E. Dreyfus, "The numerical solution of variational problems", Journal of Mathematical Analysis and Applications, vol. 5, no. 1, pp. 30-45, 1962.

[77] J. Schmidhuber, "Deep-learning in neural networks: An overview", Neural Netw., vol. 61, pp. 85-117, Jan. 2015.

[78] S. Linnainmaa, The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors, 1970.

[79] P. Werbos, "Applications of advances in nonlinear sensitivity analysis", Systems Modelling and Optimization: Proc. 10th IFIP Conf., 1981.

[80] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation" in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Mass., Cambridge:MIT Press, vol. 1, pp. 318-362, 1986.

[81] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence," in IEEE Transactions on Neural Networks, vol. 11, no. 3, pp. 697-709, May 2000, doi: 10.1109/72.846741.

[82] P. Baldi, "Gradient descent learning algorithm overview: a general dynamical systems perspective," in IEEE Transactions on Neural Networks, vol. 6, no. 1, pp. 182-195, Jan. 1995, doi: 10.1109/72.363438.

[83] Gherrity, "A learning algorithm for analog, fully recurrent neural networks," International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 1989, pp. 643-644 vol.1, doi: 10.1109/IJCNN.1989.118645.

[84] K. Hornik, M. Stincheombe and H. White, "Multilayer feedforward networks are universal approximators", Neural Networks, vol. 2, pp. 359-366, 1989.

[85] M. C. Mozer, "A focused back-propagation algorithm for temporal pattern recognition", Complex Systems, vol. 3, pp. 349-391, 1989.

[86] M. C. Mozer, "Induction of multiscale temporal structure" in Advances in Neural Information Processing Systems, CA, San Matteo:Morgan Kaufmann, vol. 4, pp. 275-282, 1992.

[87] K. J. Lang, A. H. Waibel and G. E. Hinton, "A time-delay neural network architecture from isolated word recognition", Neural Networks, vol. 3, no. 1, pp. 23-44, 1990.

[88] U. Bodenhausen and A. Waibel, "The tempo 2 algorithm: Adjusting time-delays by supervised learning", Advances in Neural Information Processing Systems 3, pp. 155-161, 1991.

[89] Tsungnan Lin, B. G. Horne, P. Tino and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," in IEEE Transactions on Neural Networks, vol. 7, no. 6, pp. 1329-1338, Nov. 1996, doi: 10.1109/72.548162.

[90] ElHihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in NIPS1995, 1996.

[91] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[92] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[93] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations", Math. Comp., vol. 19, pp. 577-593, 1965.

[94] P. Fletcher and M. J. D. Powell, "A rapid decent method for minimization", Computer J., vol. 6, no. 2, pp. 163-168, 1963.

[95] D. Goldfarb, "A family of variable-metric methods derived by variational means", Math. Comp., vol. 24, pp. 23-26, 1970.

[96] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization", Math. Comp., vol. 24, pp. 647-656, 1970.

[97] K. Levenberg. A method for the solution of certain problems in least squares. Quarterly Applied Math. 2, pp. 164-168, 1944., 2:164-168, 1944.

[98] D. Marquardt, "An algorithm for least squares estimation of non-linear parameters", J. Soc. Ind. Appl. Math., pp. 431-441, 1963.

[99] R. Schaback and H. Werner, Numerische Mathematik, Germany, Berlin:Springer-Verlag, 1992.

[100] R. Battiti, "First- and second-order methods for learning: Between steepest descent and Newton's method", Neural Computation, vol. 4, no. 2, pp. 141-166, 1992.

[101] K. Saito and R. Nakano, "Partial BFGS update and efficient step-length calculation for three-layer neural networks", Neural Comput., vol. 9, no. 1, pp. 123-141, 1997.

[102] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradient for solving linear systems", J. Res. Nat. Bur. Stand., vol. 49, no. 6, pp. 409-436, Dec. 1952.

[103] M. Mller, Exact calculation of the product of the Hessian matrix of feed-forward network error functions and a vector in time, 1993.

[104] G. Cauwenberghs, "A fast stochastic error-descent algorithm for supervised learning and optimization", Advances in Neural Information Processing Systems, vol. 5, pp. 244-251, 1993.

[105] J. Schmidhuber, "Accelerated learning in back-propagation nets" in Connectionism in Perspective, The Netherlands, Amsterdam:Elsevier, pp. 439-445, 1989.

[106] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation" in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Mass., Cambridge:MIT Press, vol. 1, pp. 318-362, 1986.

[107] S. E. Fahlman, An empirical study of learning speed in back-propagation networks, 1988.

[108] A. West and D. Saad, "Adaptive back-propagation in on-line learning of multilayer networks", Adv. Neural Inf. Process. Syst., vol. 8, pp. 323-329, 1996.

[109] M. Riedmiller and H. Braun, "A direct adaptive method for faster back-propagation learning: The RPROP algorithm", Proc. IEEE Int. Conf. Neural Networks, pp. 586-591, 1993.

[110] C. Igel and M. Hsken, "Empirical evaluation of the improved Rprop learning algorithm", Neurocomputing, vol. 55, no. C, pp. 105-123, 2003.

[111] N. N. Schraudolph and T. J. Sejnowski, "Tempering backpropagation networks: Not all weights are created equal", Advances in Neural Information Processing Systems, pp. 563-569, 1996.

[112] S. Becker and Y. LeCun, Improving the convergence of back-propagation learning with second-order methods, Sept. 1988.

[113] N. N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. Neural Computation, 14(7): 1723-1738, 2002.

[114] R. Battiti, "Accelerating Backpropagation learning two optimization methods", Complex Syst., vol. 3, pp. 331-342, 1989.

[115] Xiao-Hu Yu, Guo-An Chen and Shi-Xin Cheng, "Dynamic learning rate optimization of the backpropagation algorithm," in IEEE Transactions on Neural Networks, vol. 6, no. 3, pp. 669-677, May 1995, doi: 10.1109/72.377972.

[116] R. Neuneier and H. Zimmermann Georg, "How to train neural networks," in Neural networks: Tricks of the trade, Genevieve B. Orr, Klaus-Robert Müller, Berlin: Spring, 1998, pp. 273-423.

[117] Neural Networks: Tricks of the Trade, 1998.

[118] Y. Bengio, Neural Networks: Tricks of the Trade, 2013.

[119] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., "Backpropagation applied to handwritten zip code recognition", Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.

[120] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., "Handwritten digit recognition with a back-propagation network", Advances in Neural Information Processing Systems 2 (NIPS'89), 1990.

[121] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[122] P. Baldi and Y. Chauvin, "Neural networks for fingerprint recognition," Neural Computation, vol. 5, no. 3, pp. 402-418, 1993.

[123] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen", 1991.

[124] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[125] S. Hochreiter, Y. Bengio, P. Frasconi and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies" in A Field Guide to Dynamical Recurrent Networks, Piscataway, NJ, USA:IEEE Press, 2001.

[126] P. Tino and B. Hammer, "Architectural Bias in Recurrent Neural Networks: Fractal Analysis", Neural Computation, vol. 15, no. 8, pp. 1931-1957, 2003.

[127] J. H. Schmidhuber, "Learning complex extended sequences using the principle of history compression", Neural Computa., vol. 4, no. 2, pp. 234-242, 1992.

[128] J. Schmidhuber, "My first deep-learning system of 1991 + deep-learning timeline 1962–2013", CoRR, vol. abs/1312.5548, 2013.

[129] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks", Proc. Neural Information and Processing Systems, 2006.

[130] G. E. Hinton, S. Osindero and Y. Teh, "A fast learning algorithm for deep belief nets", Neural Comput., vol. 18, no. 7, pp. 1527-1554, 2006.

[131] J. Martens, "Deep-learning via Hessian-free optimization", Proc. 27th Int. Conf. Machine learning, pp. 735-742, 2010.

[132] M. F. Moller, Exact Calculation of the Product of the Hessian Matrix of Feed-Forward Network Error Functions and a Vector in O(N) Time, 1993.

[133] B. A. Pearlmutter, "Fast exact multiplication by the Hessian", Neural Computation.

[134] J. Martens and I. Sutskever, "Learning Recurrent Neural Networks with Hessian-Free Optimization", Proc. Int'l Conf. Machine Learning, 2011.

[135] J. Weng, N. Ahuja and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively," [Proceedings 1992] IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA, 1992, pp. 576-581 vol.1, doi: 10.1109/IJCNN.1992.287150.

[136] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In International Conference on Artificial Neural Networks, 2010. 5

[137] D. Ciresan, A. Giusti, L. Gambardella and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks", Proc. MICCAI, 2013.

[138] D. Cireşan and U. Meier, "Multi-Column Deep Neural Networks for offline handwritten Chinese character classification," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, 2015, pp. 1-6, doi: 10.1109/IJCNN.2015.7280516.

[139] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to forget: Continual prediction with LSTM", Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN), vol. 2, pp. 850-855, Sep. 1999.

[140] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[141] J. A. Pérez-Oitiz, F. Gers, Douglas Eck, and Juergen Schmidhuber. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. Neural Networks, 2002. In press.

[142] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," Journal of Machine Learning Research, vol. 3, pp. 115-143, 2002.

[143] R. C. O'Reilly and M. J. Frank "Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia, " Neural Computation, vol. 18, pp. 283-328, 2006.

[144] A. D. Blair and J. B. Pollack, "Analysis of dynamical recognizers", Neural Comput, vol. 9, no. 5, pp. 1127-1142, 1997.

[145] M. P. Casey, "The dynamics of discrete-time computation with application to recurrent neural networks and finite state machine extraction", Neural Comput., vol. 8, no. 6, pp. 1135-1178, 1996.

[146] P. Manolios and R. Fanelli, "First order recurrent neural networks and deterministic finite state automata", Neural Comput., vol. 6, no. 6, pp. 1155-1173, 1994.

[147] C. W. Omlin and C. Lee Giles, "Extraction of rules from discrete-time recurrent neural networks", Neural Networks, vol. 9, pp. 41-52, 1996.

[148] A. Vahed and C. Omlin, "A machine learning method for extracting symbolic knowledge from recurrent neural networks", Neural Comput., vol. 16, no. 1, pp. 59-71, Jan. 2004.

[149] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, May 2009, doi: 10.1109/TPAMI.2008.137.

[150] A. Graves, A. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.

[151] E. Indermühle, V. Frinken and H. Bunke, "Mode Detection in Online Handwritten Documents Using BLSTM Neural Networks," 2012 International Conference on Frontiers in Handwriting Recognition, Bari, 2012, pp. 302-307, doi: 10.1109/ICFHR.2012.232.

[152] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks", Proc. Advances Neural Information Processing Systems, pp. 3084-3092, 2013.

[153] G. E. Hinton et al., "Improving neural networks by preventing co-adaptation of feature detectors", arXiv preprint arXiv:1207.0580, 2012.

[154] P. Baldi and P. Sadowski, "The dropout learning algorithm", Artif. Intell., vol. 210, pp. 78-122, May 2014.

[155] S. I. Wang and C. D. Manning, "Fast dropout training", International Conference on Machine Learning, 2013.

[156] J. Bayer, C. Osendorfer, D. Korhammer, N. Chen, S. Urban and P. Van der Smagt, On fast dropout and its applicability to recurrent networks, 2013.

[157] G. An, "The effects of adding noise during backpropagation training on a generalization performance", Neural Comput., vol. 8, pp. 643-674, 1996.

[158] S. J. Hanson, "A Stochastic Version of the Delta Rule", Physica D, vol. 42, pp. 265-272, 1990.

[159] K. Jim, B. G. Horne and C. L. Giles, "Effects of noise on convergence and generalization in recurrent networks", Advances in Neural Inform. Processing Syst. 7., pp. 649-656, 1995.

[160] A. F. Murray and P. J. Edwards, "Synaptic weight noise during MLP learning enhances fault-tolerance generalization and learning trajectory", Advances in Neural Information Processing Systems 5., 1993.

[161] J. P. Nadal and N. Parga, "Nonlinear neurons in the low-noise limit: A factorial code maximizes information transfer", NETWORK, vol. 5, pp. 565-581, 1994.

[162] A. L. Maas, A. Y. Hannun and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models", ICML Workshop on Deep-learning for Audio Speech and Language Processing, 2013.

[163] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807-814, 2010.

[164] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. IMLR W&CP, vol. 15, pp. 315-323, 2011.

[165] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8609-8613, doi: 10.1109/ICASSP.2013.6639346.

[166] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Proc. Adv. Neural Inform. Process. Syst., pp. 1097-1105, 2012.

[167] "Pecan street database," http://www.pecanstreet.org. [Online]. Available: http://www.pecanstreet.org/

[168] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, et al., "WaveNet: A generative model for raw audio", 2016.

[169] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[170] T. Hasanin, T. M. Khoshgoftaar, J. L. Leevy and N. Seliya, "Examining characteristics of predictive models with imbalanced big data", J. Big Data, vol. 6, no. 1, pp. 69, Dec. 2019.

[171] T. Hasanin, T. M. Khoshgoftaar, J. L. Leevy and R. A. Bauder, "Severely imbalanced big data challenges: Investigating data sampling approaches", J. Big Data, vol. 6, no. 1, pp. 107, Dec. 2019.