NON-INTRUSIVE LOAD MONITORING USING MACHINE AND DEEP LEARNING TECHNIQUES

MOHAMMAD KAOSAIN AKBAR

A THESIS

IN

THE DEPARTMENT

OF

CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE (QUALITY SYSTEMS ENGINEERING) AT

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC, CANADA

MAY 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mohammad Kaosain Akbar**

Entitled: **Non-Intrusive Load Monitoring using Machine and Deep Learning Techniques**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair

Dr. Amin Hammad

_____ Examiner

Dr. Amin Hammad

_____ Examiner

Dr. Suryadipta Majumdar

_____ Supervisor

Dr. Nizar Bouguila

_____ Co-supervisor

Dr. Manar Amayri

Approved by _____

Dr. Zachary Patterson, Graduate Program Director
Department of Concordia Institute for Information Systems Engineering

_____ 2023

_____

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Non-Intrusive Load Monitoring using Machine and Deep Learning Techniques

Mohammad Kaosain Akbar

Non-intrusive Load Monitoring (NILM) is a computational technique that extracts individual appliance consumption and operation state change information from the aggregate power consumption made by a single residential or commercial unit. This technique has emerged as a reliable energy management approach that intends to reduce energy wastage and inform customers about their electricity consumption. NILM is considered as both Supervised and Semi-supervised Learning problems. The main contribution of this thesis is three-fold.

First, we evaluated some regression algorithms commonly used in NILM research based on 8 different training and testing scenarios which according to our knowledge covered major demographic factors that affect the appliance usage. The dataset used for the evaluation of the regression models, was collected from a research lab at Grenoble INP, in Grenoble, France. Furthermore, a novel Bayesian optimized Ensemble regressor model for predicting individual appliance consumption from aggregated load data is also proposed. Instead of just using the aggregated power information, the proposed model also uses demographic information from the dataset to estimate accurate consumption output of individual appliances.

NILM research often requires significant labeled data and obtaining such data by installing smart meters at the end of consumers' appliances is laborious and expensive and exposes users to severe privacy risks. Moreover, most NILM research uses empirical observations instead of proper mathematical approaches to obtain the threshold value for determining appliance operation states (On/Off) from their respective energy consumption value. The second fold of the thesis proposes a novel semi-supervised multilabel deep learning technique based on Temporal Convolutional Networks (TCN) and Long short-term memory (LSTM) for classifying appliance operation states from labeled and unlabeled data. The two thresholding techniques, namely Middle Point Thresholding and Variance Sensitive Thresholding, which are needed to derive the threshold values for appliance operation states, were also compared thoroughly. The proposed models were then evaluated using Redd, Uk-Dale and Refit datasets.

Third, we propose a novel NILM algorithm that utilizes deep learning Temporal Convolutional Networks (TCN) for the regression and classification NILM tasks. Most NILM models cannot simultaneously classify appliance operational status or estimate individual appliance power consumption. The deep TCN layers in the proposed architecture of the third fold of the thesis allow the simultaneous extraction of complex patterns in the data of the power consumption and the operational state of individual appliances. Refit data is used for the evaluation of this model.

Acknowledgments

In the name of Allah, the most gracious, the most merciful.

First and foremost, I would like to express my sincere appreciation and gratitude to my supervisor Professor Dr. Nizar Bouguila. Despite his immense academic and research responsibilities, Professor Nizar always provided comprehensive support to help me evolve into a qualified master's student. As an international student, arriving and adapting to the environment of a new country can be challenging. Professor Nizar gave me enough time and wise guidance to adapt to the surroundings. Whenever I doubted my research abilities, he encouraged me to think outside of the box. Because of Professor Nizar, I found an immense passion for research, and I will always be grateful to him for allowing me to work under his fantastic guidance. I am PROUD to be your student and will treasure your teachings and guidance forever.

Second, I would like to show sincere appreciation to my co-supervisor Professor Dr. Manar Amayri, for her endless patience and encouragement. Her constructive comments and guidance helped me towards the completion of my research and formulation of this thesis. Her instructions and approach to conducting research will be engraved throughout my life.

I am deeply grateful to my father, Dr. Shawkat Akbar for being the best support system. I owe my deepest gratitude to him for supporting me in achieving all my dreams and ambitions. No words can describe the love of my mother, who is constantly curtaining her feelings about me staying away from her for the past two years. She constantly believed in me, and I am blessed to be her son. To my younger sister Nuhi who inspired me unconditionally. Lastly, I would like to thank everyone here in Montreal who helped and motivated me throughout this.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The majority of the world's electricity, approximately 73%, is generated by fossil fuels and nuclear power, with coal-fired power making up the remaining 36.4% [1]. From the total electricity generated, residential and commercial electricity usage accounts for nearly 60% of the world's energy consumption [2]. High dependency on fossil fuel-based energy and increasing energy consumption created significant ecological concerns, especially regarding carbon dioxide (CO2) emissions, resulting in global warming [3]. In the past decade, there has been significant interest in optimizing energy management by analyzing the energy consumption of appliances in buildings. It is crucial to provide precise and fine-grained power usage information and operation patterns of individual appliances to increase energy efficiency and reduce greenhouse gas emissions for environmental sustainability. According to multiple studies [4-6], users can reduce their annual energy consumption by up to 12% when receiving feedback through load utilization data as seen in Figure 1.1. This can also help to facilitate communication between energy providers and end users. When electricity bills for each period are provided, consumers can keep track of their electricity costs while controlling and monitoring appliance status usage [7].

The conventional Intrusive Load monitoring (ILM) technique offers the benefit of acquiring more precise and comprehensive metering data since this approach requires the installation of sensors in each appliance to monitor changes in the appliance's status and gather data in real time. Unfortunately, installing numerous sensors leads to high construction and maintenance expenses and breaches customers' privacy [8]. Non-Intrusive Load Monitoring (NILM) technique does not require to deploy smart meters to each appliance but can estimate the power demand of each of them from the aggregate consumption of a household measured by a single meter installed at the entrance of the user's residential or commercial unit [9]. Figure 1.2 represents the difference between ILM and NILM. NILM is also often termed "Energy Disaggregation," as this technique breaks down the total energy consumed by numerous appliances into individual appliance consumption records [10]. The mathematical foundation and the framework behind this technique were first proposed by Hart in 1980s, where he explained how he monitored the active and reactive power of the load operation to estimate the number and operating characteristics of individual loads [9]. A general NILM framework is shown in Figure 1.3. This technique is gaining quick attention not only for developing the smart grid, but also due to its indisputable advantages. Some of those advantages are [11]:

- Detailed information on consumption: The key benefit for customers is that they will be able to adopt an energy-saving behaviour due to the analytical power consumption. Real-time information about running devices could also be a helpful tool, serving as a reminder for people to turn off appliances before they leave the house, particularly those devices that are dangerous or consume a lot of energy.

- Individual device power usage: This enables users to assess which appliances use the most energy in their homes and, more generally, how much each appliance contributes to overall energy usage.

- Identifying and detecting dysfunctional devices: This technique creates a detailed device usage record that helps monitor device status and identify defective equipment.

- Illegal load detection: It is more accurate to report potential energy theft in public and private buildings when abnormal loads are detected in homes.

- Environmental intelligence: NILM allows for alternative detection scenarios without the use of new sensors. Through this technique, one smart meter is sufficient to provide the data required to execute multiple energy-saving rules rather than having to convert all devices to smart ones, which is expensive and unsustainable for the environment.

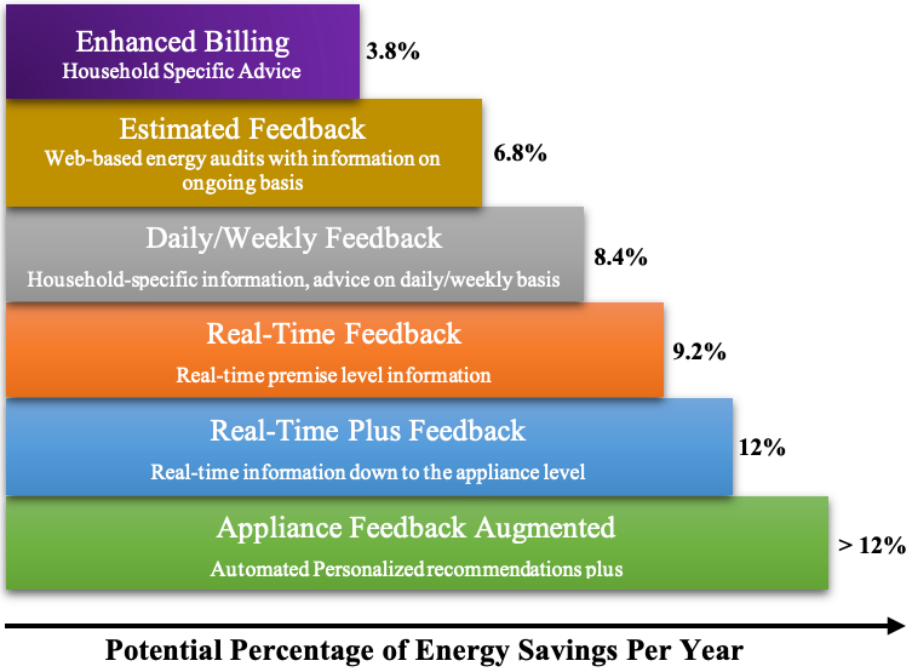


Figure 2.1: Representation of a study conducted by [4] regarding residential electricity consumption savings based on different types of consumption feedback.

NILM approaches are typically divided into supervised and unsupervised learning methods [12]. In the supervised NILM learning method, predicting the consumption of each device from the aggregated power data is formulated as a regression problem, whereas determining whether a device is On or Off is formulated as a classification problem. The unsupervised learning method, on the other hand, is often used to identify devices from the aggregate power load. It is important to note that a drawback of supervised NILM approach is the lack of sufficient labeled data. Acquiring labeled data can be expensive and time consuming which can lead to the limitation of scalability of NILM systems [13]. Individual appliances in a residential or a commercial unit of interest might be purchased anytime, so sensors might be needed to install or uninstall, requiring utility personnel to visit the building unit frequently. Thus, it may comprise the privacy of the occupants of the building, which is often not socially acceptable [14]. Furthermore, since there is a lack of a labeling stage that assigns appliance names to disaggregated profiles, unsupervised NILM approaches are primarily helpful for domain experts and not for any end customer [13]. Therefore, semi-supervised learning approach can also be used for NILM problems. The motivation behind using semi-supervised learning in NILM is to improve the performance of the model by leveraging both labeled and unlabeled data and to reduce the need for obtaining labeled data for each appliance, which can be time-consuming and expensive.

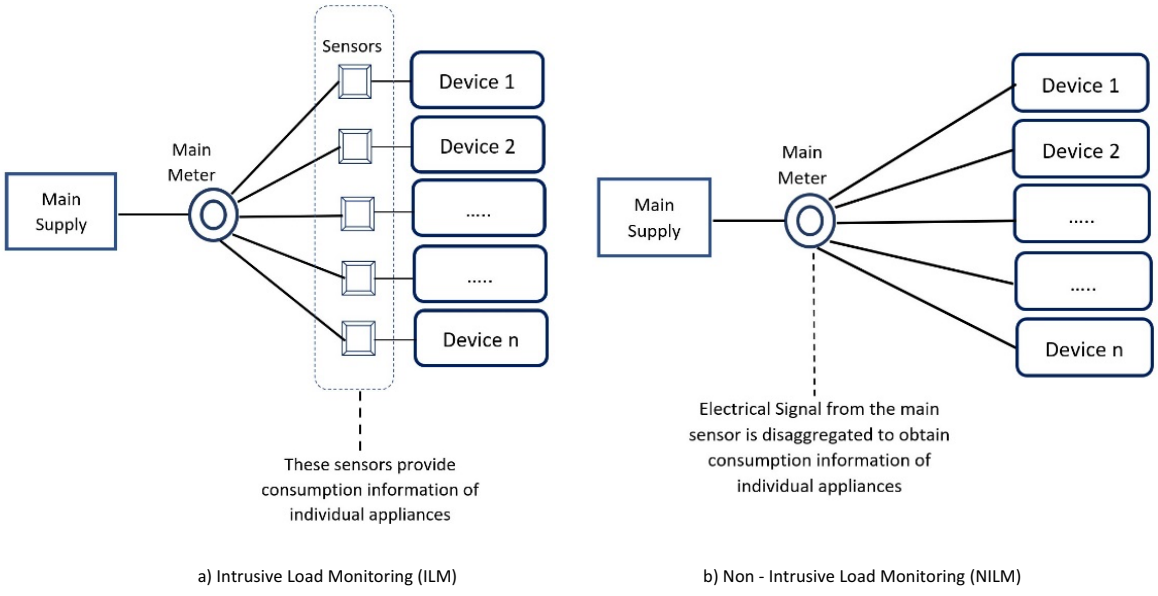


Figure 1.2: Difference between Intrusive and Non-Intrusive Load Monitoring.

1.2 Related Works

There were numerous studies conducted regarding supervised NILM for appliance load decomposition and states classification. Initially, for NILM research, the most applied techniques were different Hidden Markov Models (HMMs) approaches, some of which are highlighted in [22] and [23]. Buddhahai, and Makonin [24] proposed a multi-target regression using multi-target regression tree and rule induction from a Java-based data learning API called CLUS [25]. A key feature of the approach proposed in [24] was that apart from just considering the aggregate power load, several other relevant features to problems were considered. The approach required only a handful of model configurations for parameter tuning but had limited performance for appliances that had multiple operation states. Bi-directional LSTM NILM optimized by Bayesian inference was proposed in [26] to predict appliance consumption of four different appliances from AMPds dataset [27] and the superiority of the model is reflected by the MAE, RMSE scores when compared to NILM models implemented using generic LSTM or CNN architecture. Lin et al. [28] proposed a temporal convolutional neural network which is a CNN architecture that not only estimates appliance load but also offers the advantage of transfer learning where labelled data are obtained from source and unlabelled data are obtained from the target domain.

Hadi et al. [29] proposed a supervised machine learning network architecture which uses less computational space and time for decomposing accurate appliance load, but the proposed model was prone to overfitting with no option to extrapolate. The study in [30] used Gradient Boosting Regression through Empirical Mode Decomposition (EMD) to find consumption made by five different appliances. Schirmer et al [31] performed evaluation of four algorithms namely K-Nearest Neighbours, Support Vector Machines, Deep Neural Networks and Random Forest to find appliance load. After experimenting on five different datasets, Random Forest Regressor outperformed all the other three techniques with an accuracy of up to 93%. A combined regression along with classification subnetwork for NILM problem was proposed in [32], where an encoder-decoder mechanism is implemented in the regression network. Like the method proposed in [31], Konstantopoulos et al. [33] also proposed a similar approach using Decision Tree, Random Forest and k-NN to predict appliance consumption based on active power, reactive power, and crest factor. The experimental results show that Decision Tree and Random Forest generate more accurate output than k-NN.

Rao et al. [34] proposed an approach using Support Vector Machines with edge analysis for identifying devices and Autoregressive Moving Average method for predicting future appliance consumption. The proposed technique had an accuracy of 90% in predicting future consumption. A hybrid deep learning method using convex hull data selection technique was proposed in [35] for NILM but often predicts irrelevant consumption that does not fall under the target appliance. In [36] three steps energy decomposition method is proposed that is composed of a state identification stage, followed by individual power consumption estimation and daily electricity fitting. LightGBM shows the most promising result for obtaining the power consumption estimation from a dataset obtained from a gas station. Shin et al. [10] proposed a subtask gated network which is

comprised of two separate Deep Neural Networks – one for regression NILM and the other for classification.

A multi-output CNN architecture was proposed in [37], where the proposed approach solves both the classification and regression problems at the same time. Similar research to detect the state and estimate the power of the appliances was done in [21], where the researchers proposed a one-dimension CNN based on U-Net architecture. Another CNN-based architecture named Scale and Context-aware CNN was used by Chen et al. in [38] for obtaining improved disaggregation results from multiple appliances. A temporal convolutional network (TCN) based on the sequence-to-point model for NILM was proposed in [39], where the authors showed that for a certain number of appliances, TCN outperformed the conventional CNN in predicting appliance load. There are also other notable works on NILM based on machine learning techniques such as Decision Trees [40], Support Vector Machines [41], k-NN and Naïve-Bayes [42]. Even genetic algorithms and graph signal processing approaches have been also used for NILM research [43] and [44].

Similarly, to [39], a bi-directional TCN used by a sequence to point model is proposed in [45], that estimates the consumption of five appliances. Another NILM regression-based task was done in [46] where fully convolutional and casual neural network comprised of encoder-decoder and TCN is used for estimating appliance load consumption. An improved TCN composed of two casual convolution and one non-linearity layer was proposed in [47] and used for load prediction of four appliances in REDD dataset. Another TCN based sequence model for predicting appliance load and states is proposed in [48] and a temporal convolutional network integrated with a graphical model called conditional random field (CRF) for predicting multiple states of appliances is proposed in [49]. Handful of works using LSTM have also been done in NILM research. In [52] LSTM is used in disaggregating appliance load from the total consumption signal through experimenting with different numbers of layers and hidden units. Another deep learning model comprised of a CNN, a LSTM layer and random forest algorithm, used for identifying appliances was proposed in [50]. In [51] another appliance identification for NILM was proposed based on multilayer LSTM. All these deep learning methods for NILM based on TCN or LSTM are supervised and require large number of labeled data.

Annotating large amount of data for NILM research is expensive, time consuming and might compromise privacy of the residents or users of the commercial/residential building of interest. Thus, semi-supervised approach is feasible for NILM tasks as this learning uses small number of labeled data and large number of unlabeled data. A semi-supervised learning (SSL) based on support vector machine (SVM) for NILM task in classifying appliance states was proposed in [13]. This SSL NILM model trains the SVM using labeled data and then set labels for the unlabeled data which are then again reintegrated in the model's learning. NILM multilabel classification based on a mean teacher-student model for semi-supervised learning is proposed in [56]. The model uses both labeled and unlabeled data to classify operational states of multiple appliances at once. UK-DALE and REDD datasets are used to train and evaluate a model that adopted teacher-student structure based on Gaussian kernel trick-based maximum mean discrepancy (gkMMD) and TCN in [57]. The efficiency of the proposed model is presented by

comparing its performance with five other models. Semi-supervised learning for automated residential appliance annotation (SARAA) classifies state of a single appliance by using 1-NN semi-supervised technique [55]. When compared to the benchmarking techniques SARAA had a F1 score about 15% lower. Using decision tree classifier as eager learner and nearest-neighbor as lazy learner, a semi-supervised learning approach was proposed in [54] where features are extracted with the help of wavelet design and Procrustes Analysis. Graph based semi-supervised learning was proposed in [14] where graph-based technique is used to label the data and Multilabel K-Nearest Neighbor (MLkNN) is used to train the multiple classifiers to predict appliance state. In [58] a combination of semi-supervised (based on random forest classifier) and active learning is proposed which simultaneously handles shortage of labeled data and improves classification accuracy. An expectation maximization based semi-supervised multi-label classification technique for NILM was proposed in [53] where random k-label set (RAKEL) is used as base classifier for semi-supervised learning. The proposed approach achieved higher accuracy in classifying appliance states when compared to classification algorithms such as RAKEL and MLkNN.

A Convolutional Neural Network (CNN) based architecture was proposed to classify the operational states of the appliance in the LIT dataset by da Silva Nolasco et al. [59]. The model had around 95% accuracy in estimating the appliances' operating status. Feed Forward Neural network for predicting the appliance states was proposed in [60], where the model achieved an average F1 score of 0.77 for six appliances in the UK-Dale dataset. Xiao et al. [61] used five fully connected Deep Neural Networks (DNN) for appliance state classification. When compared to the existing Hidden Markov model (HMM) and RNN-based models, the proposed technique in [61] had a significantly better F-Measure score obtained from classifying various appliances in three houses of the REDD dataset. Kim et al. [62] proposed a combination of Gated Recurrent Unit (GRU) and Recurrent Neural Network (RNN) architectures to classify states of a total of 20 appliances in five different houses of the UK-Dale Dataset, and the proposed model achieved an average F-Measure score of 0.86 across all the houses.

When comes to NILM Deep Learning techniques for regression tasks, Zhang et al. [63] proposed a sequence to point technique based on five convolutional layer, one dense layer and two seq2seq layers. A Seq2Seq layer used for the model was comprised of CNN and RNN layer [64]. The model presented in [64] predicted the consumption made by five household appliances and had an overall Mean Absolute Error (MAE) score of 95. Serafini et al. [65] used Convolutional Recurrent Neural Network (CRNN), a combination of convolutional layer, gated recurrent units and linear layer to predict the power consumption of five different appliances from the aggregate load of five houses in the UK-Dale dataset. Based on predicted consumption values, the model had an average (MAE) score of 70. LightNILM model for predicting appliance power consumption was proposed in [66] which is comprised of a convolutional layer and sliced recurrent neural network block. Five appliances from houses of three publicly available NILM datasets were used to train and evaluate the model.

In case of models that performed both NILM regression and classification task, Saraswat et al. [19] proposed a deep neural network which predicts the operational states and power consumption made by appliances of eight residential buildings. A model comprised of LSTM and CNN architectures for doing both NILM regression and classification task was proposed by Naderian in [20]. The model was trained and evaluated using five household appliances in REFIT dataset. Another model comprised of CNN and Bidirectional GRU architectures was developed to perform same NILM tasks in [18]. Three appliances of the UK-Dale dataset were used to train and test the model whose regression and classification performances were evaluated using MAE and F1-score, respectively. Faustine et al. [21] proposed a one-dimensional CNN architecture-based NILM model which predicted both the state and power consumption of five household appliances from the aggregate power load in UK-Dale dataset.

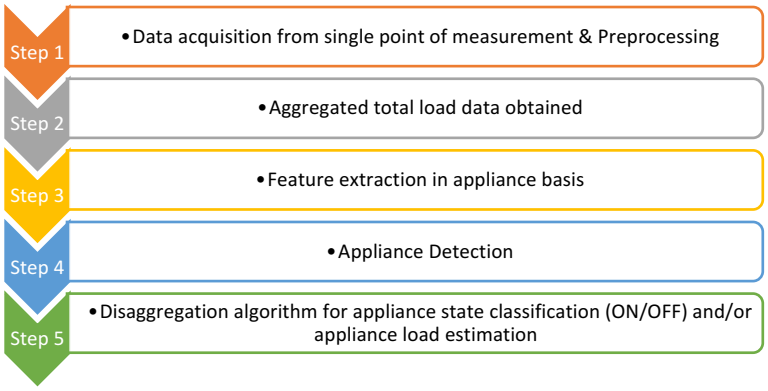


Figure 1.3: A general framework of Non-Intrusive Load Monitoring Technology

1.3 Problem Formulation

Consider a residential or commercial unit with total of A appliances. For time t period, each appliance a consume power p_t, and is in the operational state s_t that denotes On/Off (i.e. s_t is either 0 or 1). Then the aggregate power $\boldsymbol{P}_t$ is represented by

$$\boldsymbol{P}_t = \sum_{a=1}^{A} S_t^{(a)} p_t^{(a)} + \epsilon_t \qquad (1.1)$$

where, ϵ_t is the noise term. The purpose of the proposed NILM technique in this thesis is to breakdown the aggregate power $\boldsymbol{P}_t$ into power $p_t^{(a)}$ consumed and operational state $s_t^{(a)}$ for each appliance a simultaneously at time timestep t. Obtaining the power consumed is termed as NILM regression task and obtaining the appliance operating states is termed as NILM classification task.

The state S of an appliance a at time t are assumed to either be:

$$S_t^{(a)} = \begin{cases} 0\ (OFF\ State) \\ 1\ (ON\ State) \end{cases} \quad (1.2)$$

In order to solve supervised NILM problems, the dataset D contains all labeled data. But for semi-supervised NILM problems, the dataset D will consist of both labeled dataset D_{Label} and unlabeled dataset $D_{UnLabel}$. Therefore, the training dataset can be represented as:

$$D = D_{Label} \cup D_{UnLabel} \quad (1.3)$$

The labeled dataset is comprised of feature vector of aggregate power signals P_i and appliance state S_i as corresponding label vector so that

$$D_{Label} = \{ (P_i, S_i) : P_i \in \mathbb{R}, S_i \in \{0,1\}^A, i = 1,2, \dots, Label \} \quad (1.4)$$

where A is the number of appliances and states of the label data are either 0 (Off state) or 1 (On state). The unlabeled dataset, similar to the labeled dataset, is also comprised of feature vector of aggregate power signals P_j but the appliance state S_j being the corresponding label vector is assigned value -1 which indicates that the state of the a^{th} appliance is unlabeled. Therefore, unlabeled dataset can be represented as

$$D_{UnLabel} = \{ (P_j, S_j) : P_j \in \mathbb{R}, S_j \in \{-1\}^A, j = Label + 1, Label + 2, \dots, Label + Unlabel \} \quad (1.5)$$

The dataset D is then used to train and evaluate multilabel classifiers for NILM model that predict the states of multiple appliances at the same time.

1.4 Contributions

The contribution of this thesis are as follows:

1. **Evaluation of Regression Models and Bayes-Ensemble Regressor Technique for Non-Intrusive Load Monitoring:**

   Most of the regression centric NILM research uses only the aggregate load information that includes power, voltage, and frequency to predict the consumption of each appliance without considering the impact of certain demographic parameters. These parameters include but are not limited to hours of the day, weekends, days of the week, months, week of the year, seasons, quarters of the year and a few others towards consumption of each device. Researchers also tend to find one suitable technique that shows minimum error in estimating each appliance consumption without considering that appliance usage is not constant over time during the training of their machine learning models. It is inevitable to

understand that appliance usage varies according to demographic factors. For instance, the usage of the Heating appliance is more from the end of the Fall season to the beginning of Spring. Moreover, Cooling appliance tends to be used more during the summer season. Considering these demographic factors during the training of the machine learning models might not only contribute toward the accuracy of estimating appliance consumption but will also aid in predicting future consumption. In this contribution, we are trying to cover these research gaps by analyzing different suitable regression approaches for estimating power consumption for different appliances over various periods. Moreover, a novel Bayesian Optimized Ensemble regression model has also been proposed which results compare favorably with the performance of existing approaches. Here we have performed detailed analysis of six conventional machine learning algorithms which are applied to the novel Grenoble NILM dataset to find individual load decomposition of six appliances. For various demographic factors (such as seasons, working hours, weekends, etc.), we have shown that different regression algorithms tend to portray their dominance in estimating the individual consumption of the six different appliances of the Grenoble NILM dataset. We derived the ideal number of demographic factors considered during the training of those machine learning models and performance of the regression algorithms under the influence of the demographic parameters is presented through 8 different training and testing scenarios. Lastly, we proposed a new multi-output Bayesian Optimized Ensemble regression model for Non-Intrusive Load Monitoring which estimates the individual appliance energy consumption.

2. Semi-Supervised TCN – LSTM Based Deep Learning Technique with Middle-Point Thresholding Method for Non-Intrusive Load Monitoring:

NILM research often requires large number of labeled datasets to generate models that estimate power consumption or operational states of individual appliances. Obtaining such large volume of labeled datasets can be expensive and compromise the privacy of the residents or users of the building of interest. Moreover, each instance of NILM datasets typically contains consumption value of individual appliances and the aggregated load. For NILM classification problem, we have to know the operational states of each appliance. Most NILM research uses empirical approaches for identifying threshold values based on which the appliances are labeled as either ON or OFF state. The empirical approach often uses an average value above which the appliance starts to operate without thoroughly analyzing the consumption information of the dataset. This empirical approach may generate incorrect threshold values that may lead to the development of NILM models which predict impractical operational states or consumption patterns of appliances. Two appliance thresholding methods, namely Middle-point thresholding (MPT) and Variance-Sensitive thresholding (VST), are elaborately explained and discussed in [15]. These thresholding techniques help assign appliance status labels based on consumption made by individual appliances within the dataset, reducing overall error in the prediction of appliance status generated by the NILM classification model.

In order to address the issues mentioned above, this thesis suggests using a semi-supervised multilabel deep learning framework based on Long Short-Term Memory (LSTM) and Temporal Convolutional Network (TCN) for multilabel classification in NILM. LSTM is a type of recurrent neural network (RNN) that can process and use information with long-term dependencies. LSTMs can remember important information from the past and use it to make decisions in the present. This is particularly useful when dealing with time series data [16]. Temporal Convolutional Network (TCN) is a type of architecture well-suited for modelling sequential data. It employs dilated causal convolutions and residual connections, which make it efficient at stacking deep layers [17]. The proposed TCN-LSTM semi-supervised learning (SSL TCN-LSTM) is able to learn and recognize the unique energy consumption patterns of individual appliances and monitor multiple appliances at the same time. Moreover, instead of using any empirical approach, MPT and VST were used separately to deduce the thresholds of appliances for labeling their operational states in the dataset. The effectiveness of the proposed SSL TCN-LSTM model is demonstrated through case studies using three real-open access read world datasets (UK-DALE, REDD and REFIT) for NILM.

3. Deep Learning Based Solution for Appliance Operational State Detection and Power Estimation in Non-Intrusive Load Monitoring

Due to the success of Deep learning techniques, researchers are now exploring the use of deep learning methods in combination with NILM technology. Various deep learning-based NILM techniques were proposed which either predict the operational states or the energy consumed by individual appliances with only a handful of research that explores NILM deep learning techniques which performs regression and classification NILM task at the same time. However, majority of these approaches used different stacked Deep Neural Network layers [18-21] which requires sufficient computational memory and often faces vanishing gradient problem. In order to mitigate these challenges, we proposed a novel NILM deep learning technique based on Temporal Convolutional Network (TCN) architecture that simultaneously performs both NILM regression and classification tasks. The main idea of this proposed model is to disaggregate the total load into fine-grained and accurate consumption values and the operational states (either ON or OFF) of individual appliances.

1.5 Thesis Overview

- Chapter 1 introduces the motivation, background and related works of our research work and contributions.

- Chapter 2 explores performance of traditional machine learning algorithms for NILM regression tasks under the influence of various time-based parameters. A novel Bayesian optimized regressor model is proposed which is trained and evaluated using a novel NILM dataset.

- Chapter 3 introduces a new semi-supervised NILM technique to estimate appliance operational states. With limited labeled data and a large portion of unlabeled data, the model can predict the operational states of various appliances. Three real world dataset is used for the training and evaluation of the proposed model.

- Chapter 4 presents a framework which performs both regression and classification NILM tasks using deep learning techniques based on TCN architecture.

- Chapter 5 concludes the thesis and discusses future works.

Chapter 2

Evaluation of Regression Models and Bayes-Ensemble Regressor Technique for Non-Intrusive Load Monitoring

In this chapter, different suitable regression approaches for estimating power consumption for different appliances over various periods are explored. Detailed analysis of six conventional machine learning algorithms is applied on the novel Grenoble NILM dataset to find individual load decomposition of six appliances. Performance of the regression algorithms under the influence of the demographic parameters are presented through 8 different training and testing scenarios. Additionally, a novel Bayesian Optimized Ensemble regression model is proposed which results compare favorably with the performance of existing approaches.

2.1 Dataset Used

The dataset used in this literature is a novel dataset obtained from an interactive platform developed by Grenoble INP Ense3 and by G2E Lab at Institut Polytechnique de Grenoble, in Grenoble, France. This dataset records the individual consumption of six different appliances along with the aggregate power load from January 2017 to December 2021 [71].

2.1.1 GreEn-ER Building

GreEn-ER is a 22000 m² building in Grenoble hosting Ense3 engineering school and research with G2Elab, with about 1,500 students and hundreds of professors, researchers, and staff. Because it is a large building, its power consumption is also significant. On a typical day, the active power can be more than 300 kW. There are more than 1,500 meters, including more than 300 electricity consumption ones. The electric meters measure not only the consumption of the various switchboards, regarding the aggregated consumption of different zones in the building, but also some individual loads, such as the lighting and the power outlets of certain switchboards, the air handling units (AHUs), chillers, pumps, etc. The other meters concern internal and external conditions, thermic energy data, etc. The measured data are used to control the internal conditions, regarding the comfort of the occupants and to monitor the consumption. The tertiary sector is a very heterogeneous one, including activities such as office, education, malls, lodging, warehouse,

public assembly, retail, health and food. In the USA, total energy expenditures in 2018 counted for $141 billion, $23,900 per building, $1.46 per square foot [67]. Table 2.2 represents the maximum, mean and minimum consumption values all six appliances of the Grenoble dataset. As described in Figure 2.2(a), the different consumption usages are space heating, cooling, ventilation, water heating, lighting, cooking, refrigeration, office equipment, computing, and other.

Figure 2.1: GreEn-ER: a building for energy learning and research.

Figure 2.2(b) shows the energy consumption by usage in GreEn-ER building during 5 years from 2017 to 2021. Note that the decomposition by use is like the average use in the United States shown above. Space heating and cooling are important in the consumption and are mostly dependent on external conditions like outdoor temperature. As it can be seen in Figure 2.3, there is a correlation between outdoor temperature and energy consumption. Learning technics like forecasting or desegregation can take advantage of such physical behavior [68].

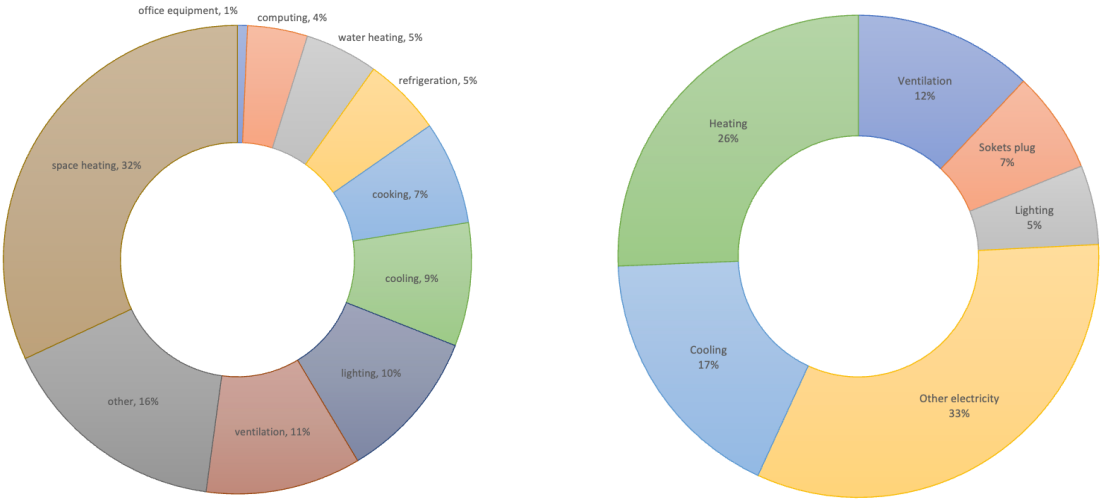


Figure 2.2 (a) U.S.A energy consumption in buildings by end use of 2018. (b) GreEn-ER consumption by end use from 2017 to 2021.

2.1.2 Available Datasets

The progress of research related to the use of machine learning in the energy consumption field depends directly on the availability of datasets, either for training, in the case of supervised approaches, or for performance testing, in the case of both supervised and unsupervised approaches. Even though datasets containing synthetic data have had their importance, datasets containing real data of electricity consumption provide, especially in the buildings field, further advances, despite increasing the difficulty in developing and applying algorithms.

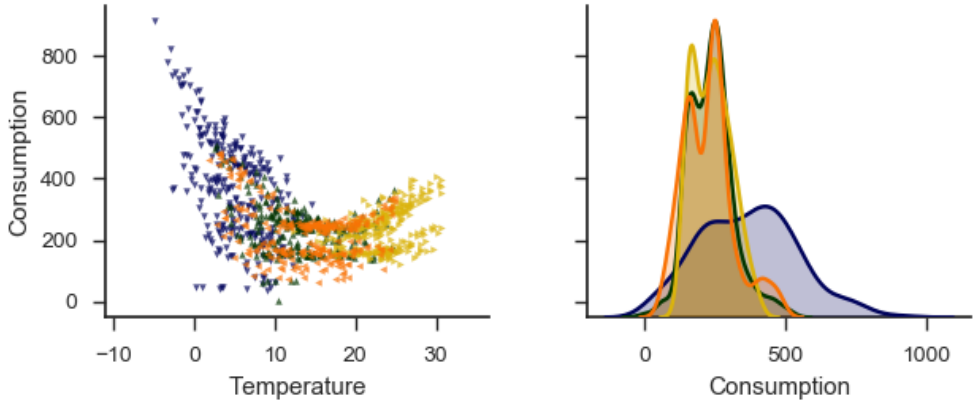


Figure 2.3: The first graph presents the correlation between energy consumption and outdoor temperature. The second graph shows energy consumption during four different seasons of France.

Therefore, it is clear the need to use real data of electricity consumption measurements for researchers to advance in the field of machine and deep learning in buildings. There are several datasets publicly available, with both aggregated and disaggregated consumption. Some of them are mentioned in [69], but most are dedicated to the residential sector (16 datasets over 20). In Table 2.1, three datasets dedicated to tertiary sectors are highlighted. It is obvious that it is necessary to develop datasets dedicated to buildings of the tertiary sector for which this dataset has been proposed. Among the existing datasets for tertiary sectors, there is Mendeley dataset [70], and the other one is accessible online in real time [71].

Table 2.1: Available datasets of energy consumption in tertiary buildings.

Dataset	Description	Country	Reference
Tracebase	It is available energy consumption data of equipment used in residential and tertiary buildings that were measured with a commercial sensor.	Germany	[83]
BERDS	Contains power measurements (active, reactive, and apparent) of a university campus, of several equipment, such as lighting, hydraulic pumps and air conditioning system.	USA	[84]
COMBED	There is data from 200 meters installed in a university campus in India.	India	[85]

The ease with which data sharing is carried out nowadays helps popularize scientific knowledge, allowing researchers from many places to develop theories from data originally obtained from other parts of the world. In addition, sharing scientific knowledge makes research more efficient, more visible, and less redundant [72]. The unrestricted dissemination of research publication and data can be called open science.

Table 2.2: Maximum, Mean and Minimum consumption values in kWh of all appliances and aggregate power load for the entire dataset and individual years.

	Total	Ventilation	Sockets plug	Lighting	Other electricity	Cooling	Heating	
Maximum Value	2037	793	980	132	971	1641	1120	Entire Dataset
Mean Value	269.09	32.43	18.40	14.30	87.8	47.08	68.99	
Minimum Value	3.13	0	0	0	0	0	0	
Maximum Value	1427	479	49.80	80	765	622	763	2017
Mean Value	298.14	36.52	18.84	16.68	89.13	53.82	83.14	
Minimum Value	25.35	0	4.22	0	0	5.84	0.25	
Maximum Value	1913.4	608	103	132	879	1641	1120	2018
Mean Value	246.46	26.58	17.99	15.59	58.38	60.70	67.19	
Minimum Value	20.49	0	3.91	0	0	0.18	0	
Maximum Value	2037	793	547	71.50	971	1599	850	2019
Mean Value	294.41	30.77	18.23	15.45	116.20	53.77	59.97	
Minimum Value	17.92	0	0.56	0	0	0	0	
Maximum Value	1167.40	322	622	65.90	537	543	724	2020
Mean Value	240.30	29.02	16.70	10.47	97.99	33.56	52.54	
Minimum Value	34.09	0.87	0	0.62	0	0	0	
Maximum Value	1180.70	569	980	68.10	215	558	691	2021
Mean Value	266.48	39.44	20.33	13.44	77.51	33.88	81.85	
Minimum Value	3.13	0	0	0	0	0	0	

2.2 Pre-processing Data and Selecting Ideal Demographic Input Parameters

As mentioned in the previous section, the novel Grenoble NILM dataset does not have missing ('NAN' and/or 'NULL') values. Therefore, no techniques were applied for missing data. Since the data were recorded through smart meters, we assume that there were no outliers. Outliers reading in this scenario would represent faulty appliances which are vital to keep for models training. Moreover, this dataset does not have any demographic parameters which we aim to use in this

research. The parameters such as day of the week, week of the year, month, quarter can be obtained from the time parameter that is already there within the dataset. Using Pandas [73], a Python programming language library, these demographic parameters were obtained. Table 2.3 shows the list of parameters with their description.

Machine learning methods typically makes decision according to how the data is provided to them and often the algorithms make better inferences out of the data by calculating the distance between the data points. Therefore, if the data points of the features are closer to each other, then the machine learning models can be trained more efficiently and quickly else if the feature values have high differences between them, then the models will require more time to train, and accuracy of the output might be lower. Table 2.2 shows that some appliances might consume electricity as high as 1000 kWh and others might consume 75 kWh of electricity. Such high differences between the values of the features might cause the models and the proposed technique to generate inaccurate output. Thus, the input features were scaled using Standard Scaler in the Sklearn Library [74].

Table 2.3: Various demographic parameters with proper description.

Name of the attribute	Attribute Description
Hour	The hour value corresponding to the timestamp of the data
Day	Date of the recorded instance
Month	Month of the recorded instance
Year	Year of the recorded instance
Day_of_week	Value representing what day of the week corresponding to the timestamp of the data
Week_of_year	Value representing what week number of the year corresponding to the timestamp of the data
Quarter_of_year	Value representing which quarter number of the year when the instance is recorded

2.3 Evaluation Metrics

2.3.1 R^2 Score

For evaluating and comparing the performance of the machine learning models, R^2 Score is used. R^2 score is a statistical measure which shows the proportion of the variance for a dependent variable which is explained by an independent variable in a regression model. This evaluation metric explains to what extent the variance of one variable is explaining the variance of another variable:

$$R^2 = \frac{\sum_{i=1}^{n} (p_i - \overline{o})^2}{\sum_{i=1}^{n} (o_i - \overline{o})^2} \qquad (2.1)$$

where,

o_i is the observed value
p_i is the predicted value
$\overline{o}$ mean of the observed concentration
n number of observations

First the data points, both dependent and independent variables are used to train a regression model. After the regression model has been generated, we can obtain the predicted value from the model. Final R^2 score is obtained when variance explained by the model is divided by the total variance. The output of the score lies between 0 and 1. R^2 score of 0 represents a model which does not explain any variation in response variable around its mean. Score of 1 represents model which is capable of explaining all variations in response variable around its mean.

Table 2.4: Clusters of data created from the Grenoble dataset which is used towards the training of regression algorithms.

Various Clusters Generated from Grenoble NILM dataset	Description
Spring	Data from 1st of March to 31st of May
Summer	Data from 1st of June to 31st of August
Fall	Data from 1st of September to 30th of November
Winter	Data from 1st of December to end of February
Working hours	Data from 0900 hrs to 1700hrs except for Saturday and Sunday
Non-working hours	Data from 1700hrs to 0900 hrs the next day except for Saturday and Sunday
Weekends	All data recorded starting from 0000hrs on Saturday to 2359 hrs on Sunday

2.3.2 Mean Absolute Error (MAE) Score

Alongside R^2 Score, the performance of the proposed Bayes Ensemble NILM Regressor model is evaluated using MAE score. Mean Absolute Error (MAE) is defined by the average absolute error that occurs between actual and predicted values. In statistics, MAE represents the result of measuring the differences between continuous variables. MAE is not sensitive towards

outliers and is useful if the distribution of the data is multimodal. This evaluation metric is known to be a scale-dependent accuracy measure and so cannot be applied towards the comparison between series that use different scales. Since the data used in this work is a time series data, MAE is ideal choice. Equation (2.2) represents the formula for MAE. The absolute sign is used to discard the formation of any negative score since the predicted outcome of the machine learning model can be larger than the true value. MAE score typically ranges from 0 to ∞ and this score is negatively oriented, which means the closer the value of MAE is to 0, the more accurate the prediction of the learning model is.

$$\text{MAE} = \frac{\sum_{i=0}^{n}(|y_i - y_p|)}{n} \qquad (2.2)$$

where,

y_i is the actual value
y_p is the predicted value
n number of observations

2.4 Evaluation of Regression Models for NILM using demographic parameter

Table 2.5: Input features provided to the model and output consumption values of the appliances

Input Features	Expected Outputs
Total Consumption	
Hour	Ventilation
Day	Socket Plugs
Month	Lighting
Year	Other Electricity
Day of the week	Cooling
Week of the year	Heating
Quarter of the year	

The first part of this evaluation focuses on investigating six traditional regression algorithms. All these algorithms can perform multi-output regression. Therefore, after every computation, each algorithm will have six outputs for six respective appliances. The input and output variables are presented in Table 2.5. The regression algorithms are used to train models based on 8 different scenarios. For each scenario, suitable algorithms are noted according to their performance for the individual appliances, rather than simply seeking an ideal regression technique which can generate an average satisfactory result for all the appliances. The parameters for the regression techniques are tuned manually in order to find the best outcome. Table 2.6 shows the train-test scenarios which were adopted for this research. Based on the training and testing data size, each algorithm required different computational time.

Table 2.6: Description of the test-train scenarios

Scenarios	Description	Number of Instances
Scenario 1	Training: entire data of the year 2017 and 2018	**17459**
	Testing: entire data of the year 2019	**8760**
Scenario 2	Training: data of summer season of the year 2017 and 2018	**4416**
	Testing: data of summer season of the year 2019	**2208**
Scenario 3	Training: data of spring season of the year 2017 and 2018	**4414**
	Testing: data of spring season of the year 2019	**2207**
Scenario 4	Training: data of fall season of the year 2017 and 2018	**4370**
	Testing: data of fall season of the year 2019	**2185**
Scenario 5	Training: data of winter season of the year 2017 and 2018	**4259**
	Testing: data of winter season of the year 2019	**2160**
Scenario 6	Training: data of working hours of the year 2017 and 2018	**5716**
	Testing: data of working hours of the year 2019	**2871**
Scenario 7	Training: data of non-working hours of the year 2017 and 2018	**6751**
	Testing: data of non-working hours of the year 2019	**3393**
Scenario 8	Training: data of weekends of the year 2017 and 2018	**4992**
	Testing: data of weekends of the year 2019	**2496**

2.5 Machine Learning Techniques

2.5.1 Decision Tree Regressor

Decision Tree (DT) regressor usually performs partition on both the feature space as well as the output value on the partition unit that has been constructed by recursive segmentation. Then the feature with the highest information gain value is split first. Usually training phase of DT comprises of feature selection, tree generation and pruning. Decision Tree algorithm breaks down the dataset into smaller subsets and simultaneously, the tree continues to grow until a stopping criterion such as max depth is reached. A fully developed DT regressor model contains decision nodes and leaf

nodes. The topmost decision node is termed as root node and this node is considered as the best predictor. DT regressor model comes with an advantage of reduced storage requirement which is controlled by tuning parameters such as minimum number of leaf nodes and maximum depth of the tree. Decision trees tend to overfit.

2.5.2 Random Forest Regressor

Random Forest (RF) Regressor is one of the most popular machine learning techniques which was first proposed by in [75]. This technique does not require any preliminary knowledge on distribution of data in the training set and is usually trained by bagging method. Because of randomness, this approach has a better generalization performance and so any model generated by RF has lower variance. Often Random Forest is robust to various condition with minimum effort. The only parameter of RF which can be tuned is the number of trees. The major benefit of RF over DT is its ability to tackle overfitting. The bagging method of RF resolves the issue of inaccurate outcomes.

2.5.3 K-Nearest Neighbor Regressor

KNN regressor is also another non-parametric machine learning method which keeps the training realizations to generate numerical predictions of target output based on similarity measures such as Euclidean distance or Manhattan Distance. A simple kNN regression implementation comprises of computing the average of the target value of ‘k’ nearest neighbors in the training set. The performance of kNN algorithm is often affected when a variation in dimension occurs due to the usage of multi-dimension distance in a highly dimensional feature space. Another limitation of this technique is that it can capture information only when the information is one-dimensional.

2.5.4 Gradient Boosting Regressor

Boosting is an efficient approach for combining various base classifiers to produce a form of a group whose performance in general is better than any base classifier [76]. Gradient boosting technique constructs a machine learning model in a stage-wises fashion and uses gradient descent method to overcome any minimization problem that may arise while building the model. Gradient Boosting Regressor (GBR) is considered as a generalization of gradient boosting that has three elements namely a loss function that is to be optimized, a weak learner to make predictions and an additive model to add weak learners to minimize loss function [77]. The choice of loss function depends on the researcher. GBR can handle data of mixed type naturally and is robust to outliers. Other advantage of GBR can be high predictive power and supporting different loss functions. Due to the sequential nature of boosting, GBR can be difficult to parallelize and thus, has a disadvantage with respect to scalability.

2.5.5 Light Gradient Boosting Machine (LGBM)

LGBM is a gradient boosting library implemented by Microsoft in 2017 [78] with an aim of making gradient boosting on decision tree faster by using two concepts: Gradient Boosting Decision Tree and Gradient-based one-sized sampling. The LGBM algorithm is faster than other tree-based algorithms because it progresses vertically in contrast to other algorithms which typically progresses horizontally or level-wise. The root and leaf of this technique can either grow vertically or horizontally. A major advantage of this machine learning technique is that it provides results with high accuracy despite being lightweight and requiring low memory for performing computation over large datasets [79]. LGBM is ideal for large datasets and might overfit if the dataset is small [80].

2.5.6 Random Sample Consensus (RanSac)

Random sample consensus (RanSac) regressor is an iterative ML technique that is designed to perform parameter estimation even if the input data is comprised of large proportion of outliers [81]. The first step of this regressor technique involves the creation of model hypothesis by repeatedly choosing random subsets of observations [82]. In the second step, these hypotheses are ranked based on their consensus with all the observations. The top ranked hypothesis is then returned as the final estimate by the regressor model.

2.6 Performance Evaluation

Eight train-test scenarios, as seen in Table 2.6, are used to train the six regression models and the input and output features of those models are also illustrated in Table 2.5. Each of the regression algorithms has parameters that can be tuned to increase the accuracy of those models. Here, grid search technique is used to find appropriate parameters for the algorithms. Empirical study of the algorithms aided to find the suitable parameters which were tuned to increase the accuracy of consumption prediction made by the regression algorithms.

Table 2.7: Parameters tuned using Grid Search for the Six regression machine learning algorithms

Algorithms	Parameters tune by Grid Search
Random Forest Regressor	*max_depth , n_estimator*
Gradient Boosting Regressor	*min_sample_leaf, n_estimator*
K Nearest Neighbor (kNN) Regressor	*leaf_size, n_neighbors*
RanSac	*max_trial*
LGBM	*boosting_type = 'gbdt', num_leaves, n_estimator*
Decision Tree Regressor	*max_depth, min_sample_leaf*

These parameters are shown in Table 2.7. Even though, some of the algorithms have similar parameters, grid search is executed on the parameters of individual algorithms separately. So, selecting an ideal parameter value of one algorithm will not influence the output performance of another algorithm. Table 2.8 shows the R^2 score for the regression models for estimating the output consumption made by individual appliances.

2.6.1 Scenario 1

In scenario 1, the model is trained using all the data of years 2017 and 2018. All six regression algorithms were trained with the mentioned data and the input and output features stated in Table 2.5 were used towards training the models. There is a total of 17459 instances used to train the model along with grid search approach to identify suitable hyperparameters. 8760 instances were used to test the models' performance. On testing the model with the entire data of 2019 and obtaining the R^2 score, model trained by RanSac regressor produced good score of 0.33 for Ventilation, 0.25 for Socket Plugs and 0.77 for Cooling. Decision Tree Regressor model is ideal for lightning and other appliances with a score of 0.62 and 0.12 respectively. For heating appliance, K-NN regressor model performed well with a score of 0.61. Table 2.7 shows the hyperparameters for the regression techniques and Table 2.8 shows the hyperparameters values which were needed to generate the best performing model for Scenario 1.

2.6.2 Scenario 2

Here, the training data is now associated with summers 2017 and 2018, having a total training instance of 4416. The testing set consists of 2208 instances of summer 2019. For this scenario, RanSac regressor model had the best score for Socket plugs (0.07), other electricity appliances (0.23) and cooling (0.80). Random Forest provided the best score of 0.14 among other techniques for the Heating appliance. Gradient Boosting Regressor scored the maximum of 0.46 for ventilation and for Lighting, LGBM regressor model had better performance for estimating consumption with a R^2 score of 0.72. The heating appliance had the lowest score in contrast to the other electricity appliances. This is mainly because during summer, heating appliance is less likely to be used and had made little consumption. Figure 2.4(f) illustrates the minimum usage of heater during the summer season.

2.6.3 Scenario 3

For scenario 3 data of spring 2017 and 2018, comprised of 4414 instances, is used for training the models. The models are then tested against 2207 instances of spring 2019. After the models were trained, based on the scores, Random Forest regressor generated models that output good consumption estimates for the ventilation, socket plugs and lighting appliances with a score of 0.54, 0.48 and 0.75, respectively. Models trained by RanSac had good output performance for other electricity appliances having score of 0.33 and cooling appliance with a score of 0.12. For heating

appliance, Gradient boosting regressor model dominates with a score of 0.60. For this scenario, cooling appliance had the lowest score in contrast to the other appliances. Unlike scenario 2 which dealt with summer data, scenario 3 trained models using spring data. Cooling appliance is less likely to be used during spring and so had made little consumption. Figure 2.4(e) demonstrates the minimal usage through lower consumption of electricity made by the cooling appliance during spring season.

2.6.4 Scenario 4

Here models are trained using 4370 instances of the entire data of falls 2017 and 2018 and 2185 instances of fall 2019, is used for testing. For this scenario better estimate of ventilation consumption is made by Decision tree model with a score of 0.53. For Socket plug and lighting Gradient boosting regressor had a score of 0.51 and 0.56, respectively. For other electricity (0.78) and cooling (0.33) appliances, it is the model trained by RanSac regressor and for heating appliance the score was 0.58 which is achieved by Gradient boosting regressor model.

2.6.5 Scenario 5

This scenario trains the model using all the winter data of 2017 and 2018 and test them with the winter data of 2019. Training set has 4259 instances and testing set contains 2160 instances. For this scenario, only two regressor models generate good estimates for all the appliances. RanSac regressor model provided satisfactory output for Ventilation (0.16), other electricity (0.18) and heating (0.19) appliances, whereas for the remaining appliance namely Socket plugs (0.43), lighting (0.47) and cooling (0.33), Gradient boosting regressor model performed well in contrast to the other regressor models.

2.6.6 Scenario 6

Unlike previous four scenarios, this scenario does not train models using seasonal data. Instead, this train-test approach will help to understand which regression models are more suitable towards finding accurate appliance consumption during office hours. Table 2.4 mentions the typical working hours in France, where for this scenario every working day for years 2017 and 2018, from 0900 to 1700 is considered for training. For testing the models, data of year 2019 are considered with the exact same condition for days and working hours. There were 5716 instances for training the regressors and 2871 instances for testing the models' performance. Once the regression models are trained with the data of this scenario, Gradient boosting regressor model is able to generate satisfactory estimates for ventilation, socket plugs and lighting having R2 score of 0.12, 0.29 and 0.64, respectively. RanSac regressor model had good outputs for the other electricity appliances with score of 0.59 and heating appliance with score of 0.60. Cooling consumption is accurately estimated by Random Forest regressor having a score of 0.76.

(c)

(d)

(e)

(f)

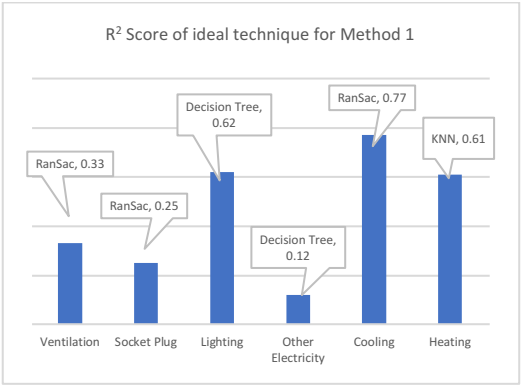
Figure 2.4: Graph showing different trends of consumption of each appliance during four seasons of France.

2.6.7 Scenario 7

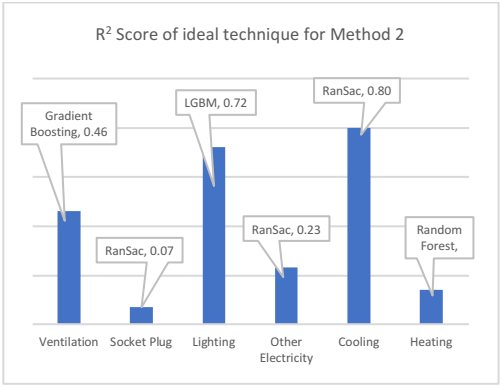
Like scenario 6, scenario 7 trains regressors with non-working hours data of years 2017 and 2018, and test regressors with non-working hours data of year 2019. The training data, like scenario 6, is from Monday to Friday, but the time now lies in between 1700 hr to 0900 hr. The training set contains 6751 instances and testing set has 3393 instances. Once tested, Gradient boosting regressor had comparatively accurate consumption output for Socket plug and lighting with R^2 scores of 0.10 and 0.84, respectively. RanSac regressor model is suitable for other electricity appliances with score of 0.53 and heating with score of 0.40. K Nearest Neighbor regressor is ideal for ventilation having accuracy score of 0.12 and for cooling appliance LGBM had the best score of 0.80.

2.6.8 Scenario 8

The final scenario uses weekend-based data. Table 2.4 shows that every data point that lies on Saturday and Sunday is considered as weekend data. Models were trained using weekend data of 2017 and 2018 and trained using data of 2019. There is a total of 4992 training instances and 2496 testing instances. Based on R^2 score, RanSac regressor is able to perform a good consumption estimate of ventilation with score 0.03, socket plugs having a score of 0.02, other electricity appliance scoring 0.70 and heating with score of 0.22. LGBM model is able to achieve satisfactory score of 0.70 for cooling and 0.14 for lighting appliance.



(a)



(b)

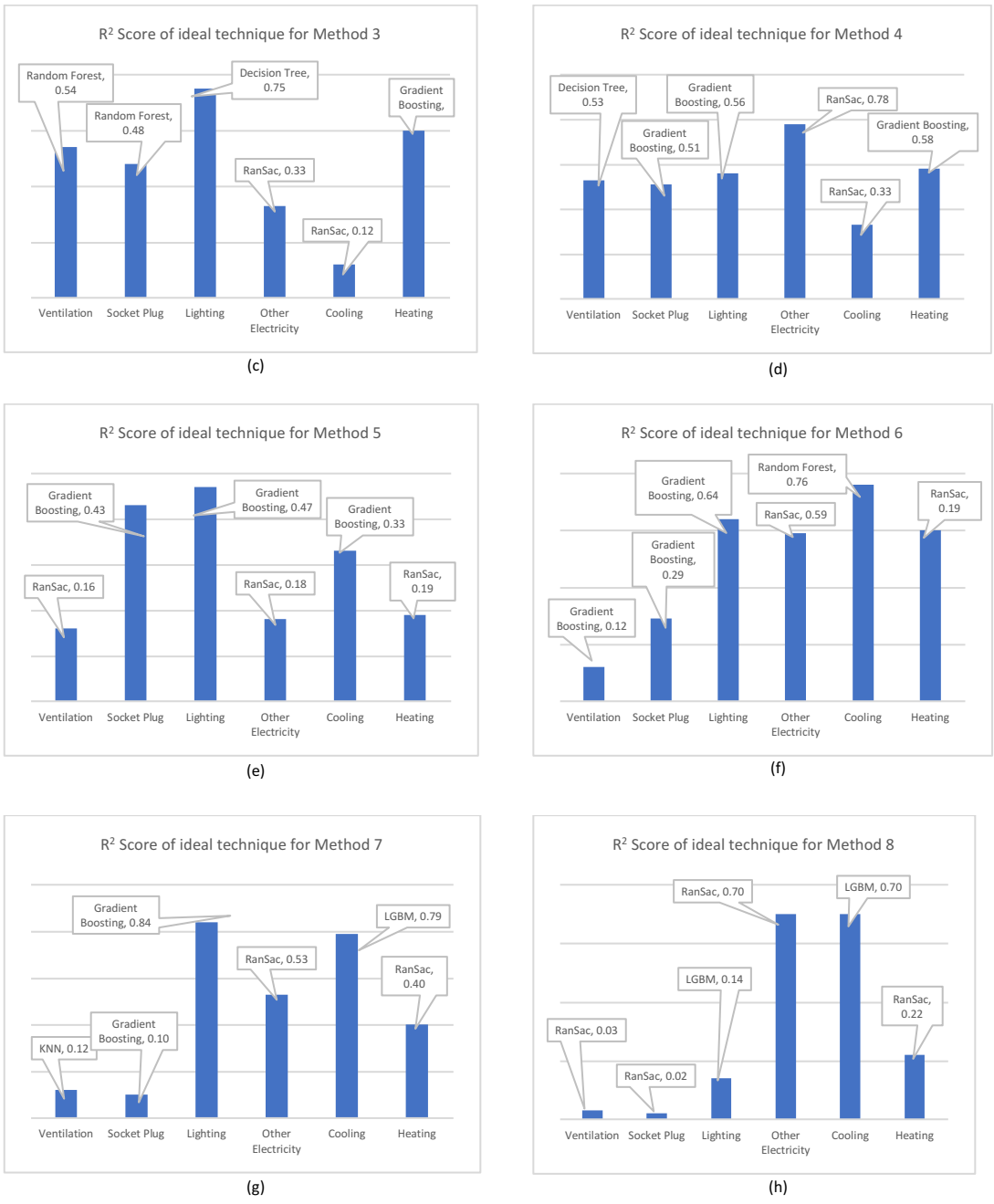


Figure 2.5: Bar plots showing the R^2 Score of the best machine learning techniques to estimate different appliance consumptions of all eight scenarios.

The results of training six different regression algorithms using 8 train-test scenarios show that it is difficult to consider a single machine learning technique for estimating the consumption made by different appliances of NILM dataset. Moreover, the performance of each technique varies under different demographic conditions which means that a set of regression models might be efficient in estimating appliance consumption during summer season, but the same regression

models might not be suitable to estimate the consumption made by the same appliances during winter season. Therefore, this section asserts the idea that instead of one machine learning algorithm, different techniques might be suitable in estimating different appliances consumptions. Additionally, it is also crucial to understand that under the influence of different demographic scenarios such as seasons, working hours, weekdays, there is a variation in performance of different regression models in estimating the power consumption of different appliances. Lastly, it is observed that using Grid Search technique to optimize the parameters of the regression models were computationally expensive in terms of overall training time and usage of memory.

Table 2.8: Output of 8 train-test scenarios

Scenarios	Techniques	Ventilation	Socket Plug	Lighting	Other Electricity	Cooling	Heating
Scenario 1	*Random Forest*	0.30 [6, 82]	0.11 [5,24]	0.35 [5,19]	0.05 [2,1]	0.70 [10,3]	0.36 [3,50]
	Gradient Boosting	0.29 [5,15]	0.14 [46,14]	0.36 [5,1]	0.05 [2,11]	0.76 [40,6]	0.39 [3,8]
	K Nearest Neighbor	0.13 [141]	0.05 [122]	0.28 [116]	0.55 [168]	0 [160]	**0.61** [181]
	RanSac	**0.33** [108,36]	**0.25** [125,21]	0.60 [54,18]	0.02 [1,1]	**0.77** [2,145]	0.51 [75,16]
	LGBM	0.30 [236,48]	0.21 [56,12]	0.51 [171,35]	0.06 [86,18]	0.71 [71,15]	0.28 [281,49]
	Decision Tree	0.02 [1,1]	0.21 [2,1]	**0.62** [6,3]	**0.12** [15,14]	0.74 [4,8]	0.26 [93,99]
Scenario 2	*Random Forest*	0.41 [3,5]	0.05 [3,5]	0.65 [74,16]	0.21 [6,8]	0.78 [36,5]	**0.14** [6,3]
	Gradient Boosting	**0.46** [30,19]	0.02 [16,16]	0.70 [31,46]	0.17 [48,49]	0.77 [2,49]	0.04 [6,8]
	K Nearest Neighbor	0.27 [36,18]	0.02 [18,4]	0.70 [69,15]	0.15 [41,19]	0.76 [35,2]	0.05 [36,18]
	RanSac	0.27 [291]	**0.07** [121]	0.37 [170]	**0.23** [229]	**0.80** [103]	0.03 [156]
	LGBM	0.32 [6,1]	0 [0,1]	**0.72** [2,3]	0.17 [7,2]	0.77 [1,8]	0.01 [0,1]
	Decision Tree	0.36 [3,1]	0.05 [3,1]	0.50 [10,5]	0.15 [8,4]	0.76 [7,2]	0.11 [8,9]
Scenario 3	*Random Forest*	**0.54** [5,9]	**0.48** [40,4]	**0.75** [6,4]	0.10 [6,2]	0.0 [1,0]	0.50 [8,9]
	Gradient Boosting	0.34 [78,9]	0.43 [99,9]	0.70 [1,9]	0.15 [11,7]	0.0 [0,1]	**0.60** [74,9]
	K Nearest Neighbor	0.40 [120,29]	0.46 [120,29]	0.70 [120,29]	0.0 [1,1]	0.0 [1,0]	0.58 [120,29]
	RanSac	0.51 [68]	0.40 [27]	0.60 [21]	**0.33** [39]	**0.12** [37]	0.48 [16]
	LGBM	0.25 [9,1]	0.48 [4,1]	0.73 [1,2]	0.07 [2,9]	0.0 [1,0]	0.53 [1,1]
	Decision Tree	0.53 [5,1]	0.46 [7,3]	0.73 [6,3]	0.09 [1,1]	0 [1,1]	0.44 [8,1]

Scenarios	Techniques	Ventilation	Socket Plug	Lighting	Other Electricity	Cooling	Heating
Scenario 4	*Random Forest*	0.45 [5,4]	0.47 [14,6]	0.48 [33,1]	0.01 [6,1]	0.22 [98,1]	0.52 [1,5]
	Gradient Boosting	0.45 [97,29]	**0.51** [68,19]	**0.56** [50,27]	0.01 [0,1]	0.28 [81,17]	**0.58** [4,9]
	K Nearest Neighbor	0.48 [91,19]	0.46 [56,12]	0.48 [91,19]	0.0 [1,0]	0.45 [26,6]	0.20 [91,19]
	RanSac	0.28 [74]	0.29 [35]	0.31 [74]	**0.78** [45]	**0.33** [67]	0.57 [52]
	LGBM	0.10 [3,1]	0.28 [2,1]	0.45 [2,7]	0 [1.0]	0.20 [17,9]	0.35 [19,6]
	Decision Tree	**0.53** [10,61]	0.37 [10,8]	0.52 [7,61]	0 [1,0]	0.19 [7,62]	0.50 [8,1]
Scenario 5	*Random Forest*	0.12 [6,2]	0.33 [52,5]	0.35 [136,6]	0.05 [2,1]	0.32 [71,2]	0.1 [0,0]
	Gradient Boosting	0.09 [5,19]	**0.43** [1,19]	**0.47** [7,19]	0.01 [0,1]	**0.33** [2,19]	0.13 [41,5]
	K Nearest Neighbor	0.15 [75,49]	0.40 [36,8]	0.40 [41,9]	0.0 [0,1]	0.0 [26,6]	0.0 [91,19]
	RanSac	**0.16** [81]	0.30 [93]	0.37 [34]	**0.18** [88]	0.03 [78]	**0.19** [88]
	LGBM	0.13 [9,5]	0.38 [1,1]	0.32 [1,1]	0.02 [9,8]	0 [0,1]	0 [0,1]
	Decision Tree	0.14 [6,10]	0.24 [7,19]	0.33 [7,18]	0.05 [2,60]	0 [0,1]	0 [1,0]
Scenario 6	*Random Forest*	0.02 [1,48]	0.20 [14,49]	0.37 [17,46]	0.10 [2,48]	**0.76** [11,41]	0.35 [67,44]
	Gradient Boosting	**0.12** [58,9]	**0.29** [22,66]	**0.64** [32,64]	0 [1,0]	0.73 [1,64]	0.54 [8,68]
	K Nearest Neighbor	0.07 [91,58]	0.27 [51,21]	0.56 [51,21]	0.09 [51,21]	0.74 [51,21]	0.47 [91,29]
	RanSac	0 [1]	0.06 [93]	0.29 [77]	**0.59** [91]	0 [21]	**0.60** [48]
	LGBM	0 [0,1]	0.28 [6,11]	0.58 [5,12]	0.13 [2,11]	0.70 [1,14]	0.43 [5,20]
	Decision Tree	0.02 [1,1]	0.11 [13,9]	0.24 [16,9]	0.10 [2,1]	0.72 [11,2]	0.33 [3,2]
Scenario 7	*Random Forest*	0.01 [6,68]	0 [0,0]	0.45 [18,61]	0.05 [1,61]	0.61 [57,60]	0.18 [3,62]
	Gradient Boosting	0.11 [72,60]	**0.10** [52,68]	**0.84** [46,69]	0 [0,0]	0.76 [1,67]	0.03 [4,62]
	K Nearest Neighbor	**0.12** [96,59]	0 [0,1]	0.68 [51,50]	0 [1,1]	0.48 [51,50]	0.26 [96,59]
	RanSac	0.11 [144,0]	0.04 [123,0]	0.51 [122,0]	0.**53** [148,0]	0.01 [115,0]	**0.40** [102,0]
	LGBM	0 [0,0]	0 [1,1]	0.84 [2,12]	0.05 [1,13]	**0.79** [5,20]	0.08 [4,20]
	Decision Tree	0.01 [5,1]	0 [1,1]	0.27 [7,9]	0.06 [4,3]	0.60 [12,7]	0.15 [3,1]

Scenarios	Techniques	Ventilation	Socket Plug	Lighting	Other Electricity	Cooling	Heating
Scenario 8	*Random Forest*	0.01 [1,65]	0 [1,0]	0 [0,0]	0.1 [2,66]	0.34 [79,63]	0 [0,0]
	Gradient Boosting	0 [0,0]	0.01 [45,60]	0 [0,1]	0.07 [88,69]	0.07 [21,69]	0 [0,1]
	K Nearest Neighbor	0 [0,0]	0.01 [81,56]	0 [0,0]	0.01 [51,50]	0.36 [51,50]	0 [0,1]
	RanSac	**0.03** [12,0]	**0.02** [63,0]	0 [1,0]	**0.70** [15,0]	0.01 [47,0]	**0.22** [20,0]
	LGBM	0 [0,1]	0 [1,0]	**0.14** [1,0]	0.06 [4,12]	**0.70** [2,11]	0 [1,0]
	Decision Tree	0.01 [6,1]	0 [1,0]	0 [0,0]	0.11 [2,1]	0.20 [6,8]	0 [1,1]

2.7 Bayes-Ensemble Regressor Model for NILM

In the previous section, six traditional ML algorithms, evaluated by different train-test scenarios, estimated consumption made by appliances of Grenoble NILM dataset. The models trained by the six traditional ML algorithms showed satisfactory performance in estimating the outputs of different appliances under different scenarios. It cannot be said that among six techniques, one of them is dominant in providing consumption of all appliances, rather, different models were efficient in estimating consumption of different appliances under any given train-test scenario. In this section, a novel Bayesian Optimized Ensemble regressor NILM technique has been proposed to estimate the power consumption of the appliances individually. The primary motivation behind using Ensemble Learning is to combine the efficiency of multiple regression algorithms together to estimate appliance consumptions. In order to reduce the overall computational time, different parameters of the regression techniques implemented within the Ensemble model are optimized using Bayesian Optimization, instead of Grid Search used in the previous findings. The proposed approach is trained and tested by the similar data cluster which is used for Scenario 1. The performance of the proposed technique is compared to the two other benchmarking techniques. Additionally, the performance of the proposed method is evaluated using R^2 Score and Mean Absolute Error (MAE).

2.7.1 Bayesian Optimization

Function optimization deals with finding the minimum or maximum of an objective function. Objective function takes a sample and returns a cost. Even though easy to understand, this objective function can be computationally difficult to calculate or may end up miscalculating the cost over time. Therefore, often objective function is termed as black box function. Function optimization is considered as an ML problem as most related algorithms have certain optimization of parameters (such as weight, coefficient, kernel size, etc.) in response to training data. Optimization often refers to finding the best hyperparameters that configure the training of the ML algorithm to generate model which can produce accurate results.

Bayesian optimization uses Bayes Theorem to find the maximum or minimum of an objective function that are naturally complex, noisy, and expensive to evaluate. This technique finds the suitable hyperparameters that are needed to train an efficient ML model. Therefore, it is inevitable to use Bayesian optimization in tuning of the hyperparameter for the generation of an efficient Ensemble Regressor NILM model.

Table 2.9: The best performing regression algorithm for individual appliances for each scenario.

	Ventilation	**Socket Plug**	**Lighting**	**Other Electricity**	**Cooling**	**Heating**
Scenario 1	RanSac	RanSac	Decision Tree	Decision Tree	RanSac	K-NN
Scenario 2	Gradient Boosting Regressor	RanSac	LGBM	RanSac	RanSac	Random Forest
Scenario 3	Random Forest	Random Forest	Random Forest	RanSac	RanSac	Gradient Boosting Regressor
Scenario 4	Decision Tree	Gradient Boosting Regressor	Gradient Boosting Regressor	RanSac	RanSac	Gradient Boosting Regressor
Scenario 5	RanSac	Gradient Boosting Regressor	Gradient Boosting Regressor	RanSac	Gradient Boosting Regressor	RanSac
Scenario 6	Gradient Boosting Regressor	Gradient Boosting Regressor	Gradient Boosting Regressor	RanSac	Random Forest	RanSac
Scenario 7	K-NN	Gradient Boosting Regressor	Gradient Boosting Regressor	RanSac	LGBM	RanSac
Scenario 8	RanSac	RanSac	LGBM	RanSac	LGBM	RanSac

2.7.2 Ensemble Model

Ensemble model is an umbrella term typically used in supervised ML for methods that combine multiple base models to make decision. By combining the base models, the ensemble model can draw predictions from unlabelled examples. The ensemble base model can be any type of ML technique (such as Decision Tree, k-NN, Neural Network, Linear Regression, etc.). The main motivation behind ensemble model is that when multiple models are combined, the errors caused by the output of one model is compensated by the efficient output of another ML model within the ensemble model. This result in the performance of an ensemble model to be superior to any single ML model. Moreover, when the computational cost of selected based models is low then ensemble models tend to be efficient.

Reasons why ensemble approach has improved predictive performance are:

- Ensemble models have a better fit to the data space because when different models are combined within the ensemble model, the research space is also extended.

- Often single machine learning models can get stuck in local optima so the combination of several models within the ensemble model allows to reduce the risk of overall model from getting stuck local minimum.

- When the data size is small, different ML algorithms will reach different hypothesis. During training, the individual models might fit the training data perfectly but will make poor predictions once unseen data are provided. In case of ensemble model, making a mean of different hypothesis of individual models will reduce the overall risk of poor hypothesis, resulting in an improvement of performance in prediction when unseen data are provided to the model.

- Ensemble models can mitigate class imbalance issue when ML models often develop a preference to one major class and ignore small classes.

2.7.3 Proposed Model

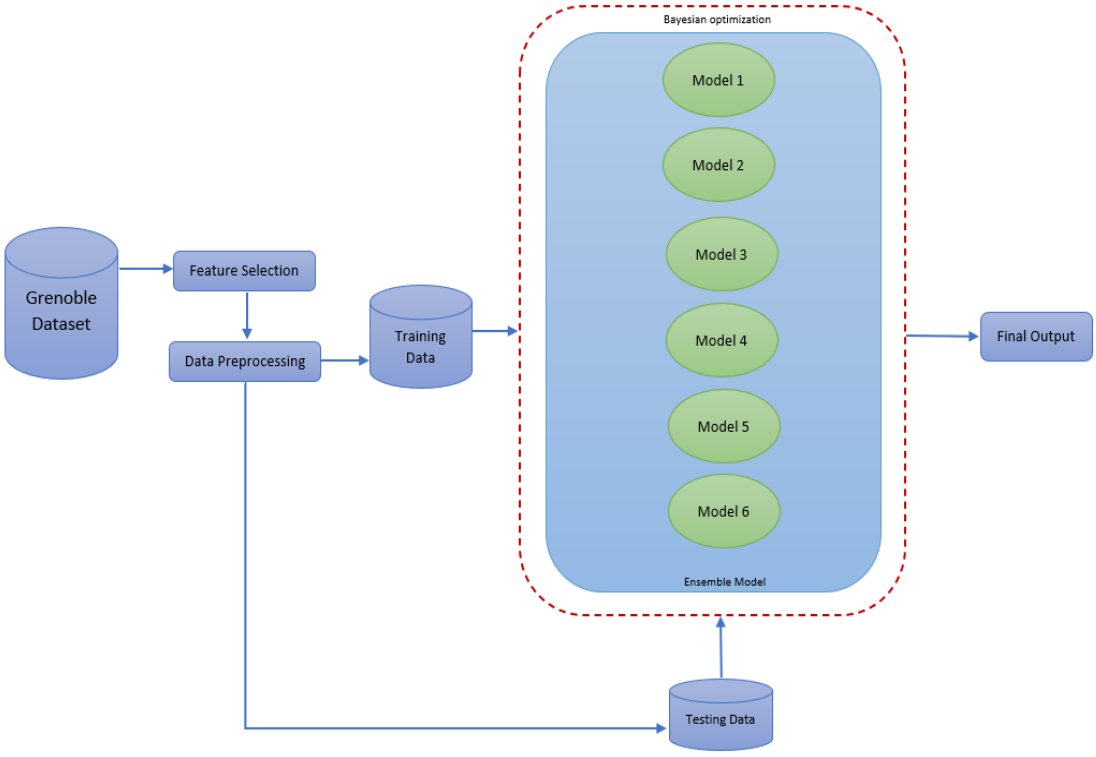


Figure 2.6: Block diagram of the Bayes-Ensemble Regression NILM model. The Ensemble model has six base models, and their averaged output will be the consumption of the appliance. There will be six ensemble models for six NILM dataset.

In this section of the thesis, a novel Bayesian Ensemble Regressor Model for Non-Intrusive load monitoring is proposed. The base techniques of the ensemble model are the six regression algorithms discussed and evaluated in section 2.5. The Bayes-Ensemble Regressor model is trained with the entire Grenoble data of the year 2017 and 2018 and the trained model is then tested using the data of year 2019. Figure 2.6. demonstrates the simple overview of the proposed Bayes-Ensemble Regressor Model. After the data is preprocessed and a separate training and testing set has been created, the training set is passed to the proposed model which in turn send the data to each of the six base regressor techniques within the ensemble model. When the regressor models are being trained, Bayesian optimization technique is constantly tuning the hyperparameters of those models and the tuning is parallelly performed for all six regressors in the ensemble model.

Table 2.10: Parameters of the regressor algorithms within the Bayes-Ensemble Model that are turned by the Bayesian Optimization technique.

Algorithms in Ensemble Model	**Parameters tuned by Bayesian Optimization**
Random Forest Regressor	*n_estimators* *max_depth* *min_samples_leaf* *max_leaf_nodes*
Gradient Boosting Regressor	*learning_rate* *min_samples_leaf* *n_estimators* *subsample* *max_depth* *max_leaf_nodes*
K Nearest Neighbor (kNN)	*n_neighbors* *leaf_size* *p*
RanSac	*min_samples* *max_trials* *stop_probability*
LGBM	*num_leaves* *max_depth* *n_estimators* *min_split_gain* *min_child_samples* *max_leaf_nodes*
Decision Tree Regressor	*max_depth* *min_samples_split* *min_weight_fraction_leaf* *min_impurity_decrease*

Table 2.10 shows the parameters that are tuned by the Bayesian Optimization technique of the proposed model. Certain regressors within the ensemble model have similar parameters as can been seen in Table 2.10, for example, “max_leaf_node” is a common parameter of gradient boosting regressor, random forest regressor and LGBM. Therefore, tuning of this parameter for one

algorithm will not influence the tuning of the same parameter for another algorithm. The parameters for which the overall ensemble model has highest R^2 scores are considered as the ideal parameters for the proposed Bayes-Ensemble NILM Regressor. Therefore, the objective of the proposed model is in two folds. First, the Bayesian optimization technique is used to find the ideal parameters for each model within the ensemble learning technique. Second, the ensemble learning allows to combine the output of each model to generate the best overall consumption estimation for each appliance of the Grenoble dataset as well as compensate probable poor performance made by any regressor technique within the ensemble model.

Bayes-Ensemble Regressor model is trained in an environment supported by AMD Ryzen 5 5600H processor, which has a clock speed of 3.30 GHz. The system also has 16 Gigabyte of Ram and backed up by 500 Gigabyte of internal storage. The system is running Windows 11 Home operating system software. The model is implemented in Jupyter Notebook ran by Python environment.

2.7.4 Benchmarking Methods

Two state-of-the-art regression approaches proposed for NILM are considered as benchmark methods for accessing and evaluating the performance of the proposed framework. Bayesian Optimization technique was used on Bi-directional LSTM to estimate appliance load consumption which was proposed in [26] by Kaselimi et al. and is used as the first benchmarking approach. In Kaselimi's work, each node of the Bi-directional LSTM model is comprised of forget gate, input gate and output gate. This Bayes Bi-LSTM method was trained and tested using the AMPds dataset, consisting of four appliances namely dryer, dishwasher, heat pump and oven. SVM with edge analysis for estimation of unknown appliance loads was used by [41]. Rao's work [34] studied the performance of the Naïve Bayes, Artificial Neural Networks, decision trees and SVM where the researchers concluded that among all the considered techniques of the study, SVM provided the most accurate result in estimating the appliance consumption. In [41] the performance of edge-SVM is outlined in detail using a novel low frequency dataset collected by a hardware module developed by the author which comprised of three appliances, namely a heater, an electric fan, and a light bulb.

2.7.5 Results and Performance comparison with Benchmarks Approach

Performance of the Bayes Ensemble model is evaluated using R^2 score and MAE. When predicted values of the model are compared to the true values, the difference between them were minimal. To understand the performance of the proposed model, the output of the model is compared to that of Bayes Bi-LSTM approach in [26] and edge-SVM approach [41]. Both benchmarking approaches trained and tested with the same clusters of Grenoble NILM dataset which is used to train and test the proposed model. Additionally, regressors trained by Scenario 1 which is discussed back in section 2.7, had same sets of data used for training the proposed Bayes-ensemble regressor model.

Therefore, the best results of the six regression algorithms of Scenario 1 (as seen in Table 2.8) can also be compared to the results of the proposed model.

Based on the R^2 score, the proposed model showed superiority in terms of estimating the consumption made by almost all six appliances. The model had better R^2 score in terms of the appliances namely ventilation, socket plugs, lighting, other electricity, and cooling. Even though K-NN regressor trained by Scenario 1 had better score than the proposed technique in terms of estimating power of heating appliance, but the margin between the scores is significantly low.

Table 2.11: Performance comparison using R^2 score between proposed Bayesian-ensemble model against approach used in Scenario 1, Bayes Bi-LSTM and edge-SVM.

Technique	Ventilation	Socket Plugs	Lighting	Other Electricity	Cooling	Heating
Bayes-Ensemble NILM Regressor	**0.38**	**0.32**	**0.64**	**0.43**	**0.78**	0.58
Best scores of Scenario 1	0.33	0.25	0.62	0.12	0.77	**0.61**
Bayes Bi-LSTM	0.30	0.05	0.20	0.42	0.26	0.07
Edge-SVM	0.28	0.22	0.46	0.39	0.67	0.42

When compared to both approaches in [26] and [41], the proposed Bayes-ensemble regression model had far better power estimation performance for all the appliances. The performance of the proposed approach shows improvement over the approaches in [26] and [41] ranging from 2% to up to 66%. For ventilation the proposed model had a slight improvement over Bayes Bi-LSTM approach, edge-SVM approach and Scenario 1 models. For socket plugs major improvement is noticed when compared to the performance made by Bayes Bi-LSTM and slightly significant improvement from edge-SVM and scenario 1 models. Similar is the case for heating but here the scenario 1 models performed slightly better (by 5%) than proposed approach. For socket plugs and lighting and cooling the proposed approach had much higher R^2 score than the method in [26] and slightly better than edge-SVM proposed by Hernandez et al. [41]. In case of other electricity, appliances, the proposed model has much higher R^2 score than the Scenario 1 model and both of the benchmarking approaches. Table 2.11 outlines the comparison of R^2 scores between the benchmarking techniques, Scenario 1 and the proposed Bayes Ensemble Regressor Model.

Likewise, the MAE score of the proposed model is also significantly better than the two benchmarking techniques as well as Scenario 1. From Table 2.12, it can be seen that MAE score for energy consumption estimation of all six appliances by the proposed model is lower than both the benchmarking techniques and scenario 1. Score of ventilation by the proposed model is 0.42, socket plug is 0.35, lighting is 0.73, other electricity is 0.64, cooling is 0.40 and heating is 0.49. On comparing with the best scores of the three models, MAE scores generated by the proposed

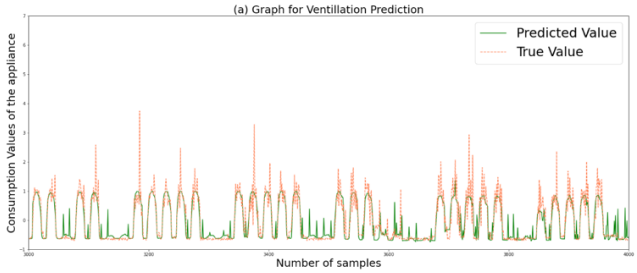
approach for ventilation, socket plugs, lighting and heating are 40%, 45%, 21% and 48%, respectively, higher against the score of scenario 1. For other electricity and cooling the MAE scores of the proposed model when compared to the best scores of the other three approaches are 21% and 46%, respectively, higher than Bayes Bi-LSTM approach. Therefore, the proposed approach shows promising result to estimate power consumptions of the six appliances in contrast to scenario 1 and two benchmark approaches.

Table 2.12: Performance comparison using MAE score between proposed Bayesian-ensemble model against approach used in Scenario 1, Bayes Bi-LSTM and edge-SVM.

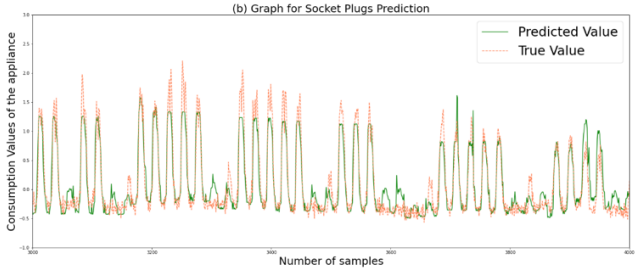
Technique	Ventilation	Socket Plugs	Lighting	Other Electricity	Cooling	Heating
Bayes-Ensemble NILM Regressor	**0.42**	**0.35**	**0.73**	**0.64**	**0.40**	**0.49**
Best scores of Scenario 1	0.73	0.65	0.92	0.87	1.21	0.97
Bayes Bi-LSTM	1.24	2.51	1.10	0.82	0.76	3.01
Edge-SVM	3.45	4.21	3.67	1.92	4.61	3.31

2.8 Discussion

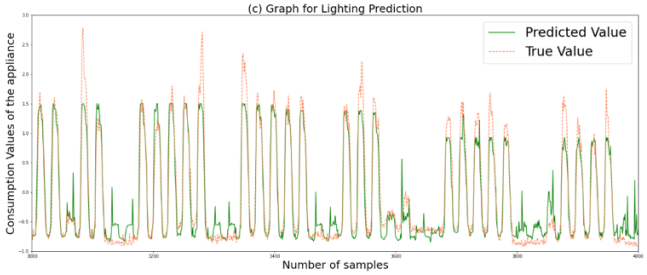
The first part of this chapter explored various regression algorithms based on multiple training-testing strategies. Six regression algorithms were trained with eight different approaches. For every scenario, different ML algorithms generated satisfactory consumption estimates for various appliances. After training models with several unique methodologies, considering different demographic parameters and analysing the results thoroughly, it can be said that even though when NILM models are trained with entire year of data in general, the model still might not be efficient for estimating appliance level power consumption. This is mainly because consumption pattern of certain household appliances depends on handful of demographic parameters such as season, working hours, weekends which is important for NILM research, and all the reviewed literature mentioned in this chapter did not take those factors into account when constructing state-of-the-art ML models for predicting appliance level consumption. Therefore, considering those parameters are important for suitable and efficient device level consumption and it should be considered that for consumption detection for each appliance, one algorithm alone might not be suitable instead different algorithms might be ideal for different appliances. The later part of this chapter proposed a novel Bayesian Optimization Ensemble regressor model technique for performing non-intrusive load monitoring. The base models used in the novel Bayes-ensemble approach are those regression models that have been trained and tested by 8 different scenarios. Based on the R^2 and MAE score, the proposed model performed well on the Grenoble NILM dataset in predicting appliance level consumption since the input features were demographic parameters and hyperparameters of the base regressor models of the ensemble model were optimized by Bayesian optimization technique.



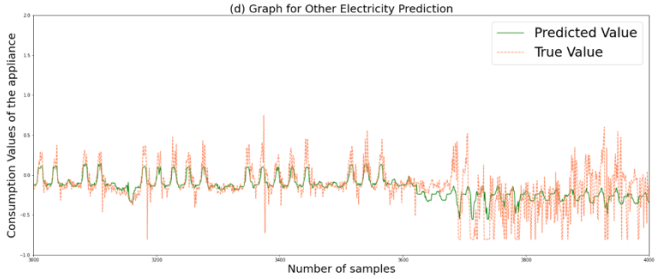
(a) Graph for Ventillation Prediction



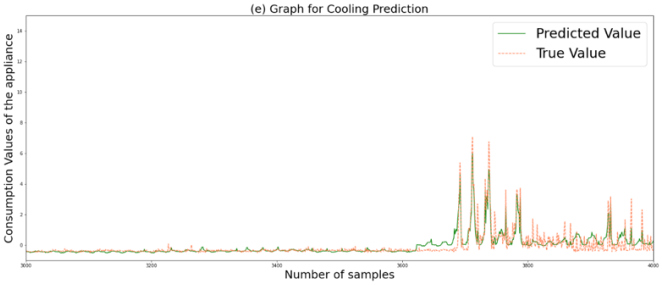
(b) Graph for Socket Plugs Prediction



(c) Graph for Lighting Prediction



(d) Graph for Other Electricity Prediction



(e) Graph for Cooling Prediction

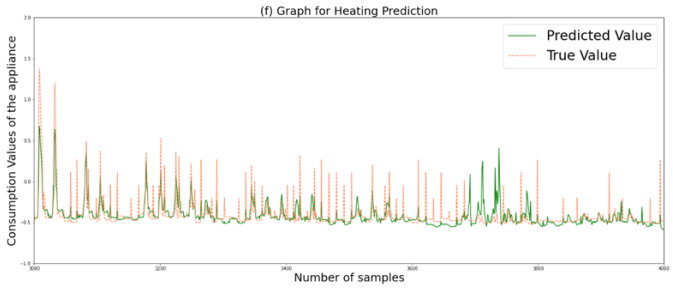


Figure 2.7: Test versus Prediction Graph of the proposed Bayes-Ensemble NILM Regressor Model.

Chapter 3

Semi-Supervised TCN – LSTM Based Deep Learning Technique with Middle-Point Thresholding Method for Non-Intrusive Load Monitoring

This chapter proposes a semi-supervised multilabel deep learning framework based on the Mean Teacher Model comprising Temporal Convolutional Network (TCN) and Long Short-term Memory (LSTM) architectures to estimate the operational states of the appliance from houses of Redd and UK-Dale and Refit datasets. The proposed model can learn the unique consumption pattern of various appliances from small number of labeled and large number of unlabeled instances. Operational state labels of the appliances from the three datasets are assigned using two thresholding techniques and impacts on the performance of the proposed model by the thresholding techniques are also explored thoroughly in this chapter.

3.1 Thresholding Techniques

NILM datasets are comprised of aggregate power load along with the consumption made by individual appliances and do not contain any explicit label for appliances' operational state. Classification NILM involves predicting the appliance state based on the aggregate signal and it is not feasible to determine whether a particular appliance is turned On or Off by simply looking at its power consumption. Thus, a prerequisite in deriving operational status for individual appliances for correct NILM classification approach is to establish a threshold $\lambda^{(a)}$ for each appliance *a* and define

$$S_t^{(a)} = H(P_t^{(a)} - \lambda^{(a)})$$

where if,

$$H(x) \geq 0\ ;\ S_t^{(a)} = 1\ (On\ State)$$

or

$$H(x) < 0\ ;\ S_t^{(a)} = 0\ (Off\ State) \tag{3.1}$$

Ideally the threshold $\lambda^{(a)}$ for appliance *a* is determined by series of power consumption data $P_t^{(a)}$ made by appliance *a.* This section discusses two thresholding approaches which are

explored in this chapter for setting individual threshold for each appliance to determine whether the appliance is On or Off from its input power signal.

3.1.1 Middle Point Thresholding (MPT)

In this thresholding approach, all the power values of appliance *a* in the training set are considered and a clustering algorithm, in this case K-Means [86], is applied to split the training set of appliance *a* into two clusters. Two centroids denoted by $m_0^{(a)}$ representing the Off state and $m_1^{(a)}$ representing the On state of appliance *a* are then derived from the two clusters. In MPT, the threshold $\lambda^{(a)}$ for appliance lies fixed between the two centroid values.

$$\lambda^{(a)} = \frac{m_0^{(a)} + m_1^{(a)}}{2} \qquad (3.2)$$

3.1.2 Variance-Sensitive Thresholding (VST)

VST was proposed in [87] and similar to MPT, this thresholding approach also uses K-Means clustering to find two centroids for each class from the power values of appliance *a.* Instead of just using mean of two centroids, the standard deviation $\sigma_k^{(a)}$ for the points in each cluster is also used for determining the threshold of appliance *a* according to the following equation

$$val = \frac{\sigma_0^{(a)}}{\sigma_0^{(a)} + \sigma_1^{(a)}} \qquad (3.3)$$

$$\lambda^{(a)} = (1 - val)\, m_0^{(a)} + (val)\, m_1^{(a)} \qquad (3.4)$$

$\sigma_0^{(a)}$ and $\sigma_1^{(a)}$ denote the standard deviation of points which belong to the clusters that represent the OFF and ON state of appliance *a* respectively. If $\sigma_1^{(a)}$ is greater than the $\sigma_0^{(a)}$ then threshold moves towards $m_0^{(a)}$ which prevents misclassifying the power values that are away from the centroid $m_1^{(a)}$ of ON state [15]. Therefore, the threshold set by VST is lower than the MPT.

3.2 Proposed Teacher-Student Semi-supervised Method based on TCN and LSTM

The proposed method is inspired by the recent development of NILM models using deep learning techniques. Deep learning NILM techniques are often not capable enough to deal with the classification of multiple appliance states (multilabel classification) using a small portion of labeled and a large portion of unlabeled data at the same time. The proposed semi-supervised Mean Teacher-Student method based on TCN, and LSTM can address this challenge.

3.2.1 Temporal Convolutional Network (TCN)

The temporal convolutional network (TCN) is a time series data processing algorithm introduced in [55]. To address the challenge of extracting long-term time series information, TCN introduces two key structures, namely dilated convolution, and residual block. A detailed explanation of these two structures are as follows.

1) Dilated Convolutions

The dilated causal convolution is considered as the primary structural component of the TCN. If input $X = (x_0, x_1, x_2, \dots, x_t, \dots, x_T)$ is a one-dimensional time series and a filter $f: \{0,1,2,3, \dots, n-1\}$, then the dilated convolution operation $L(\cdot)$ of the sequence element T can be defined as:

$$L(T) = (X *_d f)(T) = \sum_{i=0}^{n-1} f(i) \cdot x_{T-\mathrm{d}\cdot i} \qquad (3.5)$$

where n denotes the filter size, d is the dilation factor and T-$d \cdot i$ represents the direction of the past. When filter size n and dilated factor d is increased, the TCN can effectively expand the receptive field, which enables an output at the top layer to receive a wider range of input information. The computational efficiency of the whole model can also be improved by parallelly processing the same filter in each layer. Moreover, the output information of the network is only impacted by past input information, avoiding the "leakage" from future to past [45]. Figure 3.1 shows dilated casual convolution having dilation factors = 1,2,4,8.

2) Residual Blocks

Besides adjusting the filter size n and dilation factor d, another procedure to expand the receptive field size of TCN is by increasing the number of hidden layers. However, increasing hidden layers in deep networks will affect the stability of model training and cause vanishing gradients. To deal with this issue, the TCN adopts the residual block [64]. The details of the residual block for the proposed model are shown in Figure 3.2.

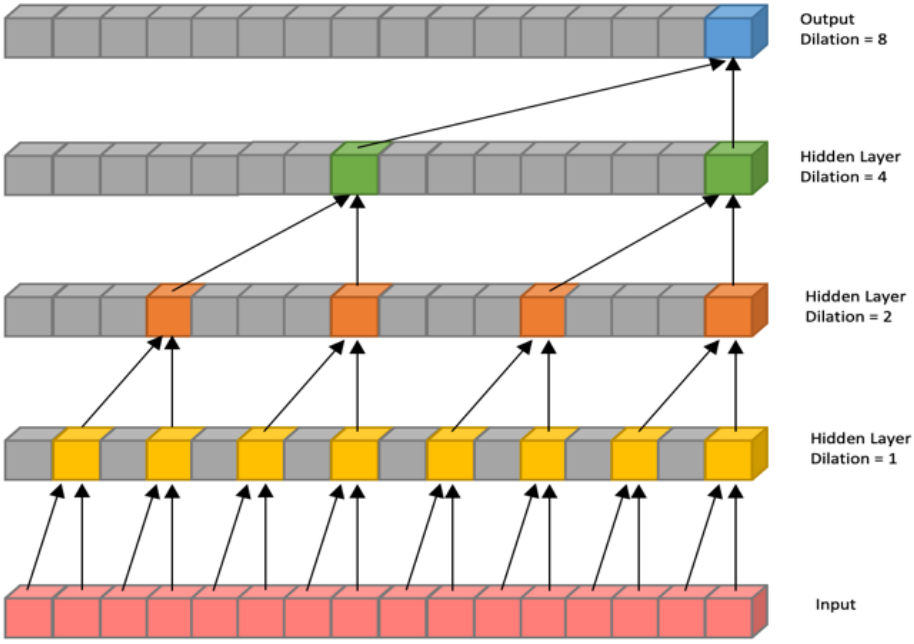


Figure 3.1: A dilated causal convolution with dilation factors = 1,2,4,8 and filter size, k=2

A residual block consists of two main branches. One branch of the residual block performs a transformation operation on the input, that is $X^{(h-1)}$. Another branch (often known as skip connection) performs a simple *1×1* Conv transformation which helps to maintain a consistent number of feature maps in parallel with the existing branch and improve gradient flow. The output of the residual block can be expressed as:

$$X^{(h)} = activation\left(F\left(X^{(h-1)}\right) + X^{(h-1)}\right) \qquad (3.6)$$

where *F(.)* is a series of transformation operations having a structure comprised of dilated causal convolution layer, the Batch Normalization, GELU as the activation layer, followed by Spatial dropout. The dilated causal convolution layer extracts the hidden features from the given input. Batch Normalization [88] is used to improve the training speed and is utilized to the convolutional filters. In this chapter, a nonlinear activation function called Gaussian Error Linear Unit (GELU) is used instead of the traditional ReLU activation function. When compared to ReLU activation function, GELU retains some of the negative information and so is better at retaining the load feature information [39]. Finally, spatial dropout is used for regularization which prevents the over-fitting issue of the deep network. The output of the network is sum of the output of two branches in the residual block.

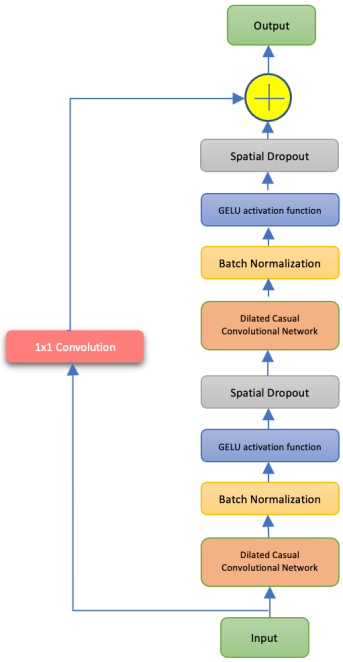


Figure 3.2: TCN Residual Block for the proposed

3.2.2 Long-short term memory (LSTM)

The recurrent neural network (RNN) has high efficiency in the prediction of time series and gets better at predicting the data based on the passage of time. However, RNN faces difficulty to recall input information that is too far apart, therefore, the long-term dependency problem is drawback of traditional RNN [89].

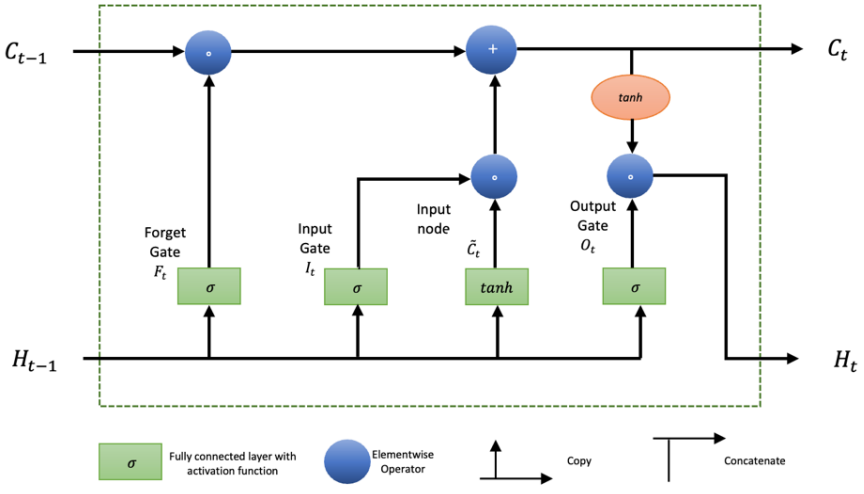


Figure 3.3: LSTM structure diagram

LSTM is a variant of RNN that mainly deals with gradient disappearance, allowing the neural network to remember the content for a prolonged period and increase network reliability [90,91]. LSTM architecture has a structure which makes it capable of reducing or increasing information to the cell state. There are three gates in LSTM namely input gate, forget gate, and output gate [92], performing the functions of read, write, and reset respectively. Figure 3.3 displays the structure diagram of a LSTM.

Initially, LSTM receives current input (X_t), output of previous module (H_{t-1}) and cell state of previous module (C_{t-1}). These received values are used by the LSTM to generate new memory whose information includes new output (C_t) and cell state (H_t). The forgetting gate acts like a valve. When the input gate is opened, a lot of information floods into the memory. During this process, a forgetting mechanism is required to remove the information which is already in the memory. This is done by the forgetting gate. It looks at H_{t-1} (previous output) and X_t (current input) and outputs a number among 0 with 1 for every digit in the cell state C_{t-1} (previous state); 1 represents completely saved, and 0 represents fully deleted. The calculation formula is:

$$F_t = sigmoid(W_f[H_{t-1}, x_t] + b_f) \tag{3.7}$$

W_f is the weight matrix, b_f is the bias term, and *F* is the output through this network whose values are in range (0, 1). The output indicates the probability of the previous cell state being forgotten. Output 1 is "Completely reserved" and 0 is "completely discarded".

After current input circulates the "forgets" part of the neural network, it also enters the input gate in LSTM. The input gate always requires the newest memory. This gate is comprised of two parts. First part of the gate is a sigmoid layer named as the "input threshold layer" that decides which values are needed to be renewed. The second part of the input gate is a *tanh* layer that establishes a vector $\tilde{C}_t$ of values between -1 and 1. The formula is as follows:

$$I_t = sigmoid(W_n \cdot [H_{t-1}, X_t] + b_n) \tag{3.8}$$

$$\tilde{C}_t = tanh(W_m \cdot [H_{t-1}, X_t] + b_m) \tag{3.9}$$

$$C_t = F_t * C_{t-1} + I_t * \tilde{C}_t \tag{3.10}$$

Here, W_n is the weight matrix, b_n is the bias item, W_m and b_m is the weight matrix and bias item, respectively, that needed for updating the state of the unit [93]. C_t represents the state of the updated memory unit. In equation (3.10), first and second part of the input gate (I_t and $\tilde{C}_t$, respectively) are multiplied element-wise to decide whether to update the state of time-step memory unit. Dot product function is also performed by forgetting gate F_t with C_{t-1} to decide whether the original state of the time-step memory unit should be retained or not.

The output gate allows the LSTM to select output relevant information while suppressing irrelevant information. This gate controls the flow of information out of the memory cell by enabling the LSTM to remember important long-term dependencies in the input sequence while filtering out irrelevant noise and information. Calculation formula for the output gate is as follows:

$$O_t = sigmoid(W_0[H_{t-1}, X_t] + b_0) \quad (3.11)$$

$$H_t = O_t * \tanh(C_t) \quad (3.12)$$

First the sigmoid activation function which takes the previous hidden state (H_{t-1}) and current input (X_t) to generate O_t, whose value lie in the interval [0,1]. Then the memory cell state C_t in the *tanh* activation function is multiplied with O_t to generate output of H_t. H_t is not only related to the input X_t under the time step *t* and the activation value H_{t-1} of the hidden layer in the previous time step. It is related to the memory unit state C_t under the current time step.

Bidirectional LSTM (BiLSTM) [94] is an extension of the described LSTM architecture, where two LSTMs are applied to the input data. Firstly, an LSTM is applied on the input sequence (i.e., forward layer). Then, LSTM is applied to a reverse form of the input sequence (i.e., backward layer). Typically, in power system, at a certain instance, the power load data are not only influenced by factors such as holidays and social environment, but also affected by the past input features, as well as the future input features to some extent can also reflect the present load features [95]. BiLSTM network is capable of capturing and extracting the characteristics and features of the information before and after [96]. Thus, such capability of the BiLSTM architecture makes it an ideal solution for NILM problems and also act as the main motivation behind using it in this chapter.

3.2.3 Mean Teacher Model

The semi-supervised objective for NILM is achieved through the Mean Teacher-Student model. The Mean Teacher model utilizes both labeled and unlabeled data to improve the performance of a model. It maintains two copies of the model, a "student" model, and a "teacher" model [97]. The student model is trained on the labeled data, while the teacher model weight $\boldsymbol{\phi'}$ is updated as an exponential moving average (EMA) of the student model's weights $\boldsymbol{\phi}$ which is expressed as

$$\phi'_t = \alpha\phi'_{t-1} + (1-\alpha)\phi_t \quad (3.13)$$

where *t* is the training steps and α is the smoothing hyperparameter. The final output of the model is determined by the student network. The student model is updated during training with the supervised loss, here is termed as multilabel classification loss.

$$\mathcal{L}_{classification}(y, \hat{y}) = \frac{1}{A}\sum_i y_i \log\big(sigmoid(\hat{y}_i)\big) + (1 - y_i)\log\,(1 - sigmoid(\hat{y}_i)) \quad (3.14)$$

where y is the actual state of appliance, $\hat{y}$ is the output of student network and A is the number of appliances. The teacher model is updated with the unsupervised loss, termed as multilabel consistency loss:

$$\mathcal{L}_{consistency}(\hat{y}, \bar{y}) = \frac{1}{A}\sum_i (sigmoid(\hat{y}) - sigmoid(\bar{y}))^2 \qquad (3.15)$$

where $\bar{y}$ is the output of the teacher network. Since the appliance states are in binary form, the multilabel classification loss is calculated using average binary cross entropy. The multilabel consistency loss is calculated by comparing the student model's predictions on the unlabeled data with the teacher model's predictions. Then the classification and consistency loss are combined to generate the semi-supervised loss:

$$\mathcal{L}_{semi-supervised} = \mathcal{L}_{classification} + \omega * \mathcal{L}_{consistency} \qquad (3.16)$$

where, ω is the weight ramp-up function. Usually, the value of ω is set to a minimum (typically 0) and is increased to maximum value which is 1 over the training epochs. Back propagation is used to minimize the overall composite loss through Adam optimization algorithm [98]. Using the teacher model as a reference, the student model can learn from labeled and unlabeled data, resulting in improved performance.

3.2.4 Model Implementation Details

The input layer of the neural networks of both student and teacher framework consists of a Gaussian noise layer which is usually activated during the time of training the semi-supervised model. The Gaussian input layer injects Gaussian noise into the input data that enters the neural networks which improves the generalization of the entire network and make the training procedure of the model more robust [99]. The receptive field of TCN network should be large for the output of the respective network to receive a wider range of input information. A large receptive field of TCN is commonly achieved by increasing the number of hidden layers but this will increase the overall computational complexity, resulting in more consumption of resources such as memory. This research aims to avoid such consequences from occurring so the alternate approach of choosing right filter size and dilation factor is performed here to increase the size of the TCN receptive field. For the proposed approach, the filter size is set to 2 and dilation factor is set to 1,2,4 and 8. The TCN network of the proposed model comprised of Six TCN residual blocks with each block having 128 filters for hidden layers. The spatial dropout rate is set to 0.2.

Alongside TCN layers, both teacher and student frameworks also have two LSTM layers. The first LSTM layer consists of 64 filters with kernel size of 1*5 and the second LSTM layer is comprised of 120 filers with similar kernel size as the first LSTM layer. A fully connected layer is attached at the end of the final layer of both teacher and student networks. The smoothing hyperparameter of the EMA is set to 0.99. The classification loss is calculated through ignoring instances labeled by -1 and consistency loss is calculated through the comparison of predictions made by the teacher and student framework. The semi-supervised loss is then calculated using the classification and consistency losses. The semi-supervised loss is then used to update the student network with Adam optimizer. The value of the learning rate with Adam optimizer is set to 0.001. The teacher network is then updated using EMA. A total of 200 epochs was used during training with option for early stopping to avoid overfitting. Once the training is over, final results are obtained from the student network.

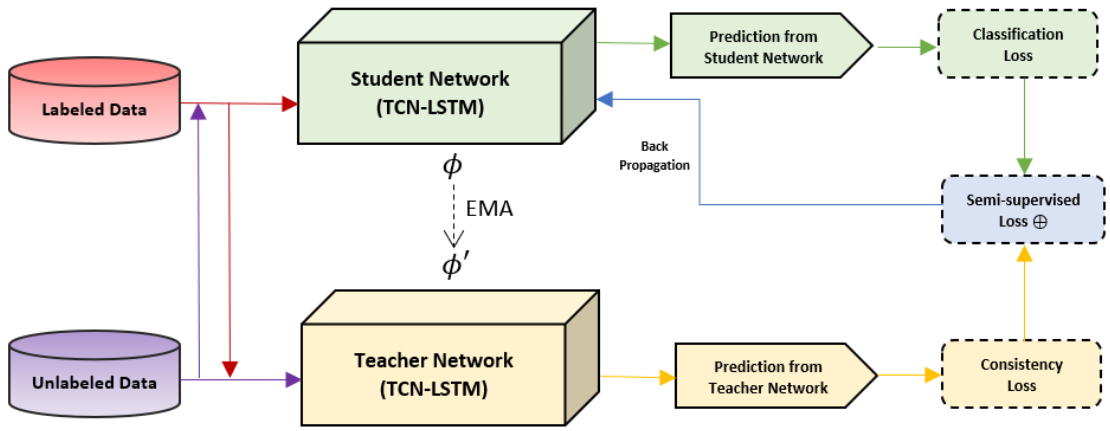


Figure 3.4: Architecture of the proposed semi-supervised TCN-LSTM technique for appliance states classification in NILM

3.3 Datasets Description and Preparation

Three real-world publicly accessible NILM datasets were used in finding the suitable threshold technique and then training and evaluating the proposed semi-supervised technique. These datasets are comprised of consumption readings made by multiple appliances over different ranges of time. The time of the dataset is in Unix time stamp and the power consumption values are in watts. This section describes the three NILM datasets and steps regarding their preparation to train and evaluate the proposed model.

3.3.1 Dataset Description

1) REDD Dataset

The Reference Energy Disaggregation Data Set (REDD) [23] is a publicly available dataset specifically designed for energy disaggregation. It is one of the popular datasets for evaluating energy disaggregation algorithms. It comprises both aggregate and sub-metered power data from six distinct homes in Massachusetts, USA. The data were collected from approximately forty homes in Boston and San Francisco over a range of two weeks to one month from a total of 48 circuit breakers. The researchers monitored the entire home voltage and current at high frequencies (16 kHz) to capture the true AC waveforms of the total electrical energy signatures in the houses. Circuits are labeled with clear descriptions along with the major loads presented on the recorded circuits.

2) Uk-Dale Dataset

The UK Domestic Appliance-Level Electricity dataset [100] contains information on power consumption gathered from five residential buildings in the UK ranging from the year 2013 to 2015. The dataset includes data on over 10 different types of household appliances. The frequency of aggregate consumption varies across households, with some having a low frequency of 1 Hz and others having a high frequency of 16 kHz. All the appliances of this dataset are sub-metered at 1/6 Hz.

3) REFIT Dataset

The REFIT dataset [101] contains records of both aggregate and appliance-level power usage gathered from 20 homes in United Kingdom from October 2013 to June 2015 by researchers from the University of Southampton. Each instance of the dataset is recorded every 8 seconds. Unlike the REDD dataset, REFIT provides fine-grained information about different areas within a house, such as the power usage of a "Computer site," which includes multiple appliances such as desktop computers, laptops, charging stations, and printers. Moreover, the dataset also comprises additional metadata about each recorded home, such as the number of occupants, the year the building was constructed, the size of the home, and the total number of appliances.

3.3.2 Preparation

The proposed semi-supervised NILM method predicts and classifies the states of four appliances commonly used in each selected house of three datasets. These appliances are microwave (MW), washing machine (WM), dishwasher (DW). From REDD dataset, Houses 1 and 3 are selected, from

UK-Dale Houses 1 and 2, and for Refit Dataset Houses 2,3 are selected. These selected data of houses also had other appliances, but the four appliances of interest were common between all the selected houses of the three datasets. At each instance, the total consumption for each dataset is equivalent to the sum of consumption made by the four selected appliances. For the accurate analysis of the proposed model training and evaluation, Unix time is converted to datetime, and the original series of data are resampled to a larger sampling interval of 60 seconds. For instance, in UK-dale dataset for houses 1 and 2, aggregate and individual power consumptions are measured by the smart meter at a constant rate of 6 seconds and for this research these series of power readings are resampled at intervals of 60 seconds. The missing values were handled using linear interpolation. Two sets of threshold values of the four appliances for each house of three datasets are then obtained using MPT and VST methods.

Table 3.1: Threshold Values for determining operation status for four appliances of different houses in REDD, UK-Dale and REFIT dataset.

Thresholding Technique	Dataset	House Number	MW	WM	DW	FD
Middle Point Thresholding (MPT)	REDD	House 1	442	925	772	504
		House 3	418	931	751	510
	UK-Dale	House 1	498	887	692	541
		House 2	452	962	704	575
	REFIT	House 2	519	1022	788	584
		House 3	522	1007	772	576
Variance-Sensitive Thresholding (VST)	REDD	House 1	57	195	104	62
		House 3	62	207	142	71
	UK-Dale	House 1	54	182	98	59
		House 2	49	103	85	62
	REFIT	House 2	71	218	171	88
		House 3	68	109	92	82

In case of MPT, for any particular appliance, two clusters are formed based on consumption values using the K-Means clustering algorithm. Next, the centroid values of the two clusters are derived. The lower centroid value indicates consumption values for the OFF state, and consumption values of the ON state surround the larger centroid value. Then, a mean value of the two centroids is calculated. This mean value is the threshold value of the appliance of interest. Now, the threshold value is subtracted from each consumption value of the appliance. If the value after subtraction is greater than or equal to 0, then the appliance is in an ON state for that specific consumption value. Otherwise, if the subtracted value is less than 0, the appliance for that consumption value is in the OFF state. These steps are repeated for each appliance of the dataset.

In the case of the VST method, instead of the mean, the standard deviation of the centroids of two clusters is derived for the consumption value of an appliance. Now, the standard deviation value of the ON cluster is divided by the sum of the ON and OFF clusters' standard deviation values. For simplicity in explanation, the newly derived value can be termed "val". The threshold value then is the product of the centroid value for the OFF cluster and val subtracted from 1, added to the product of val with centroid value for the ON cluster. Such calculations in obtaining thresholds help to prevent miss classification of consumption values that indicates ON but are furthest from the ON cluster. This results in consumption values of the OFF cluster having less variance, and thus for the exact same appliance in the same dataset, the threshold value by the VST method is lower than the MPT method (see Table 3.1). Then deriving operational states for each consumption value of an appliance is similar to that of the MPT method where based on the outcome of threshold value subtracted from the power, it is determined whether an appliance is ON or OFF.

The appliance operation status can be deduced from the threshold values of Table 3.1. Since the semi-supervised learning technique is being studied here, a separate set of instances of each house from each dataset were selected randomly where operation status of individual appliances is set to -1, indicating that the operation status of the appliance is not labeled. The final form of dataset will comprise aggregate power load along with operation status of individual appliances (-1,0 and 1) over time. The proposed semi-supervised model is then trained, and the performance of the model is evaluated a total of six times separately for two thresholding methods based on three datasets. For REDD dataset, House 1 is used for training and House 3 for testing. When UK-Dale dataset is considered, House 1 is used for training and House 2 is used for testing. Finally for REFIT, House 2 is used towards training and House 3 for testing.

3.4 Evaluation Metrices and Benchmarks

3.4.1 Evaluation Metrices

The performance of the proposed SSL TCN-LSTM model is evaluated based on three traditional evaluation metrics. The first one is F1 score which is ideal for multilabel classification [102].

$$F1 = \frac{2*\mathrm{TP}}{2*\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \qquad (3.17)$$

where, TP is termed as true positive, FP is false positive, and FN is false negative. F1 score ranges between 0 and 1. The more the F1 score is closer to 1 the better the model is. F1 micro is the second evaluation metric that is derived from F1 score and is used to assess the quality of multilabel binary classification problems that deals with imbalanced data [103]. NILM datasets are mostly imbalanced, meaning there are inequal distributions of ON (value 1) and OFF (value 0)

states within the dataset which makes F1 micro a good choice for accessing the overall performance of the proposed SSL TCN-LSTM method. The F1 micro score is as follows:

$$F1_{micro} = F1(\sum_{i=1}^{A} TP_i, \sum_{i=1}^{A} FP_i, \sum_{i=1}^{A} FN_i) \qquad (3.18)$$

where, A is the total number of appliances in the dataset. Like F1 score, the value of F1 micro score also ranges from 0 to 1. The final evaluation metric is the Hamming loss (HL). This metric represents the fraction of labels which have been incorrectly predicted by the model [33] and is given by

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{A}|f(x_i)\Delta y_i| \qquad (3.19)$$

where N is the total number of test instances and $f(.)$ is the multilabel classifier. y_i is the actual label set for the input sample x_i and Δ indicates the symmetrical difference between the actual label set (y_i), A is the number of labels and the label set generated when input sample is provided to the classifier ($f(x_i)$). Hamming loss score also ranges between 0 and 1 but this is a negatively oriented score, meaning that the model that generates HL score closer to 0 is better compared to the one that generates a score closer to value 1.

3.4.2 Benchmarking Approaches

The performance of the proposed SSL TCN-LSTM technique based on MPT and VST thresholding approach is compared to two other state-of-the-art semi-supervised methods for NILM. A semi-supervised TCN (SSL TCN) approach was proposed in [56], where TCN within similar mean teacher-student architecture was developed. ReLU was used as the activation function within all the residual blocks of TCN of the model in [56]. The filter size was kept at 3 and the dilation factor was 2. Instead of using any thresholding technique, the SSL TCN model was trained based on the theory that for each appliance, if the consumption value is more than 50% of its highest recorded consumption value within the dataset then the operational state of that appliance is determined to be ON.

A BiLSTM NILM technique for appliance operational status classification was proposed in [64] based on supervised multilabel classification loss. The model in [64] has four LSTM layers, each having 64 hidden units. This BiLSTM model was implemented as a semi-supervised BiLSTM (SSL BILSTM) by Yang *et al.* using the same loss function and architecture as their proposed SSL TCN. For the fair performance comparison of the proposed SSL method presented in this chapter with both benchmarking approaches, SSL BILSTM and SSL TCN, were implemented using the same network structures, loss functions, parameters, and appliance thresholding mentioned in [56].

3.5 Result Discussion

The proposed SSL TCN LSTM model is trained by three datasets. Appliance threshold from each of the three datasets is obtained separately by using MPT and VST thresholding techniques. Therefore, for each datasets the proposed model is trained and evaluated twice based on the thresholding approaches. Overall model performance is analyzed by the F1-micro scores. For models' individual appliance state prediction capability, Hamming Loss and F1 score are used. The best score in each category is highlighted in italic-bold style. For the Redd dataset, house 1 is used for training and house 3 for testing (redH1→ redH3). For the UK-Dale dataset, House 1 is used for training and House 2 for testing (daleH1→ daleH2) and in the case of the Refit dataset, house 2 is for training and house 3 is for testing (refH2→ refH3).

Table 3.2 shows the average overall F1-micro score of the proposed model and the benchmarking approaches based on all three datasets. In Table 3.2, the proposed model using MPT approach outperformed the other three methods. The proposed model using VST technique also had a satisfactory overall F1 micro scores when compared to the two benchmarking methods. While SSL TCN had a balanced score, the BILSTM based benchmarking method performed poorly compared to the benchmarking TCN and the other two proposed methods. Overall, the proposed SSL TCN LSTM technique using MPT had a 6% higher F1 micro score compared to the SSL TCN which is the best performing benchmarking approach based on REDD dataset. For Uk-Dale dataset, MPT-based approach achieved 0.970 F1 micro score which is 3% higher than the second-best score obtained by SSL TCN benchmarking technique. Similarly, for the Refit dataset, the same model also achieved a higher F1 micro score of 0.985, followed by the proposed approach using VST, having F1 micro score of 0.966. Therefore, when considering the overall performance, the proposed SSL TCN LSTM model based on MPT approach, according to F1 micro score, outperformed the other proposed model using VST and the two benchmarking models in estimating appliance states. This is reflected from the high F1 micro scores of the MPT based proposed method for all three datasets as seen in Table 3.2.

Table 3.2: Overall $F1_{micro}$ score comparison between the proposed and benchmarking models using all three datasets

Methods	***REDD*** ***(redH1→ redH3)***	***UK-DALE*** ***(daleH1 → daleH2)***	***REFIT*** ***(refH2 → refH3)***
SSL TCN LSTM using MPT	***0.986***	***0.970***	***0.985***
SSL TCN LSTM using VST	0.961	0.931	0.966
SSL TCN	0.922	0.952	0.927
SSL BILSTM	0.896	0.878	0.885

The performance of the proposed method in classifying the states of four appliances namely – microwave (MW), washing machine (WM), dishwasher (DW) and Fridge (FD), is accessed and compared to the benchmarking methods through the Hamming Loss and the F1 scores. Tables 3.3 to 5 represent the Hamming loss scores based on the operational states of four appliances for the Redd, Uk-Dale and Refit dataset, respectively. From Table 3.3, proposed SSL TCN LSTM model using the MPT approach achieved good hamming loss scores in predicting appliance states for MW, DW and FD appliances of REDD dataset. Hamming loss score for MW of the proposed method using MPT is 0.064 which represents 35% improvement over model proposed using VST approach. For DW and FD appliances, the hamming loss scores of proposed MTP based model are 0.122 and 0.059 respectively, which are significantly better than the remaining three methods. Only for WM appliance, the hamming loss score from the proposed model using VST approach is 0.081 which is lower than the MPT based model and the other two benchmarking approaches. Washing machine is an appliance which is occasionally (usually twice or thrice each week) used at homes. The VST method causes the OFF clusters to have less variance which reduces the overall threshold value. Therefore, there are more ON instances available by the VST method than the MPT, making washing consumption data more balanced by the VST method for the REDD dataset. Table 3.4 highlights the hamming loss score of the four appliances derived from the two proposed models and the benchmarking approaches for UK-dale dataset. Here, for all the four appliances, the MPT based proposed model showed superiority in predicting the states. MW appliance obtained a hamming loss score of 0.145 which is about 9% improvement over the score of next best performing model. For WM appliance the score is 0.116 which represents an improvement of 11%, score of DW appliance is 0.131 indicating improvement of 12% and score of FD is 0.174 with an improvement of 4% over the score of the second best performing model. The hamming loss scores of the appliances obtained from the proposed and benchmarking models are shown in Table 3.5. Here, MW appliance obtained a score of 0.077, WM and FD appliance obtained 0.068 and 0.122 scores, respectively, by the proposed SSL TCN LSTM model developed using MPT approach. For these three appliances – MW, WM and FD, the improvement over the second best performing model, that is, proposed model trained using VST approach, is by 62%, 64% and 9%, respectively. For the remaining DW appliance, score by the VST based proposed method is 0.084 which outperforms the MPT based method by 8%. Overall, the hamming loss scores obtained for the REDD and Refit datasets are better than the scores for the UK-Dale dataset. It can also be concluded that on comparing hamming loss scores for each dataset, proposed MPT based SSL TCN LSTM model performs significantly better than the remaining proposed method and the two benchmarking techniques.

Table 3.3: Hamming Loss Score comparison of the classification performance between the proposed and benchmarking models for four appliances using REDD Dataset

REDD				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.064***	0.110	***0.122***	***0.059***
SSL TCN LSTM using VST	0.087	***0.081***	0.149	0.062
SSL TCN	0.102	0.098	0.137	0.076
SSL BILSTM	0.136	0.113	0.151	0.082

Table 3.4: Hamming Loss Score comparison of the classification performance between the proposed and benchmarking models for four appliances using Uk-Dale Dataset

UK-DALE				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.145***	***0.116***	***0.131***	***0.174***
SSL TCN LSTM using VST	0.158	0.129	0.147	0.181
SSL TCN	0.174	0.184	0.158	0.200
SSL BILSTM	0.163	0.202	0.155	0.194

Table 3.5: Hamming Loss Score comparison of the classification performance between the proposed and benchmarking models for four appliances using Refit Dataset

REFIT				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.077***	***0.068***	0.091	***0.122***
SSL TCN LSTM using VST	0.125	0.112	***0.084***	0.134
SSL TCN	0.136	0.159	0.117	0.165
SSL BILSTM	0.144	0.170	0.126	0.188

For further evaluation of the proposed models, F1 scores for all four appliances are also obtained for all three datasets as can be seen from Table 3.6 to Table 3.8. When the proposed model is trained with the REDD dataset (Table 3.6), model using the MPT approach showed efficiency through the highest F1 score among the proposed model with VST approach and the two benchmarking approaches for all four appliances. The F1 score of MW is 0.638, for WM is 0.857, DW is 0.941 and FD is 0.958. The improvement of F1 scores of MPT-based proposed approach by all four appliances range between 2% to 31% over the other three methods. The proposed SSL TCN LSTM using VST approach had slight superiority in terms of F1 scores generated by the semi-supervised TCN as well as LSTM benchmarking approach where the latter benchmarking approach performed poorly among the three other methods. Table 3.7 represents performance of the proposed and benchmarking approaches based on the UK-Dale dataset. The benchmarking TCN approach shows slightly better performance for classifying states of all four appliances when compared to the proposed method that used VST for finding the appliance threshold. For all four appliances, the

benchmarking TCN approach, however, could not outperform the proposed SSL TCN LSTM model which used the MPT technique. Here, in case of MW, WM, DW and FD appliances, the proposed MPT-based method produced F1 scores of 0.918, 0.587, 0.988 and 0.879, respectively. In case of models trained with Refit dataset, for WM and FD appliances the benchmarking TCN generated higher F1 scores, and the proposed model based on the VST technique provided better operational state classification for MW and DW appliances. But, the proposed SSL TCN LSTM using MPT thresholding technique had higher F1 scores and the benchmarking BILSTM approach had lower F1 scores for all four appliances when compared to the other two remaining methods. The F1 score of the MW appliance for the proposed MPT based model is 0.883, WM is 0.906, DW is 0.852 and FD is 0.978. Therefore, for all four appliances the SSL TCN LSTM model using MPT outperformed other methods in terms of the F1 score for all three datasets.

Table 3.6: F1 score comparison of the classification performance between the proposed and benchmarking models for four appliances using REDD Dataset

REDD				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.638***	***0.857***	***0.941***	***0.958***
SSL TCN LSTM using VST	0.485	0.821	0.907	0.933
SSL TCN	0.422	0.732	0.922	0.896
SSL BILSTM	0.380	0.694	0.885	0.772

Table 3.7: F1 score comparison of the classification performance between the proposed and benchmarking models for four appliances using UK-Dale Dataset

UK-DALE				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.918***	***0.587***	***0.988***	***0.879***
SSL TCN LSTM using VST	0.879	0.516	0.951	0.842
SSL TCN	0.884	0.562	0.973	0.865
SSL BILSTM	0.835	0.389	0.914	0.803

Table 3.8: F1 score comparison of the classification performance between the proposed and benchmarking models for four appliances using REFIT Dataset

REFIT				
Methods	MW	WM	DW	FD
SSL TCN LSTM using MPT	***0.883***	***0.906***	***0.852***	***0.978***
SSL TCN LSTM using VST	0.829	0.881	0.823	0.952
SSL TCN	0.800	0.896	0.819	0.961
SSL BILSTM	0.764	0.812	0.768	0.915

The primary reason behind which benchmarking BILSTM performed poorly is due to the number of BILSTM layers used in the model since complexity in deep learning training increases with a large number of layers. Furthermore, the VST thresholding technique generates threshold values which are lower than MPT, resulting in formation of a more imbalanced NILM dataset comprised of appliance operational status. Therefore, the performance of the proposed model trained with datasets that used VST techniques is less than the model trained with datasets which used the MPT technique. From the above case studies, it is well understood that semi-supervised deep neural network architecture comprised of TCN, and LSTM layers is ideal for multilabel classification of appliance operational states. The satisfactory performance of the proposed SSL TCN LSTM model is backed by i) Mean Teacher-student model which is used as the semi-supervised learning technique and ii) training the model using dataset which prepares appliance operational status using Middle Point Thresholding instead of choosing threshold values arbitrarily.

Chapter 4

Deep Learning Based Solution for Appliance Operational State Detection and Power Estimation in Non-Intrusive Load Monitoring

This chapter introduces a novel NILM algorithm that utilizes deep learning Temporal Convolutional Networks (TCN) for the regression and classification NILM tasks. The deep TCN layers in the proposed architecture extract complex patterns in the data and estimate the power consumption and the operational state of individual appliances. The proposed model is evaluated using real-world household power usage data. The results show the effectiveness of the proposed method in detecting the appliance states and estimating individual appliance loads when compared to a benchmarking approach.

4.1 Dataset Preparation

For training and evaluating the proposed novel TCN based NILM regression and classification technique, a publicly accessible NILM dataset called REFIT [101] is used. The REFIT dataset was created as part of the REFIT (Real-world Experiments for Future Internet of Things) project and is intended for use by researchers and practitioners in the field of NILM. The REFIT dataset includes power measurements taken over a period of several months from 20 households. The dataset contains both aggregate power measurements as well as ground truth power consumption data for a range of appliances, including lighting, heating, cooling, and household appliances.

Five appliances are selected to perform the NILM regression and classification task using the proposed technique. These appliances are Fridge-Freezer (F), Washing machine (WM), Dishwasher (DW), Microwave (MW) and Kettle (K). Among the twenty household consumption records, two houses – house 3 and house 11 were selected. These five appliances were common between the selected two houses. The aggregated and the individual appliance consumption of the two houses were recorded either between 6 or 8 second intervals. The consumption record of these two houses were down sampled to 60 seconds. The missing values were filled up using linear interpolation. Once the consumption records were down sampled, house 3 had a total of 885095 instances and house 11 had 564697 instances. Consumption data of each house is used to separately

train and evaluate the proposed model, i.e., the proposed model is trained and tested twice. 80% of the data from house 3 is split for training and remaining 20% for testing. Similar train-test split is done for house 11.

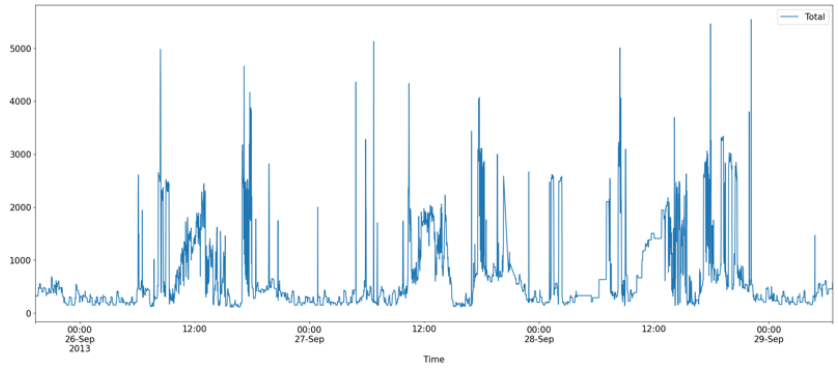
Table 4.2: Training and Testing instances of house 3 and house 11.

House Number	Training Instances	Testing Instances	Total Instances
House 3	708076	177019	885095
House 11	451757	112939	564697

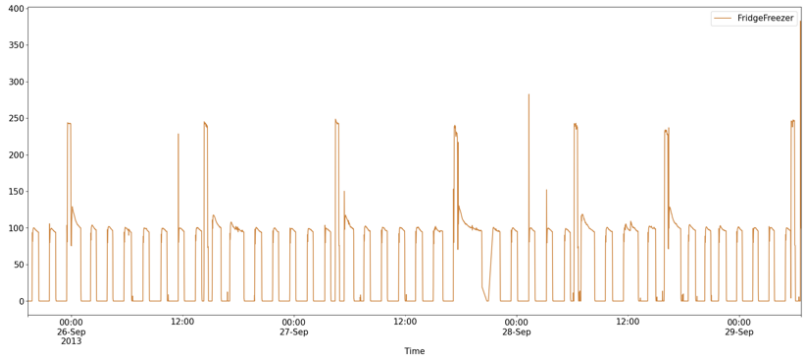
Table 4.2: Threshold values of the five different appliances of house 3 and 11. If the consumption value of an appliance is equal or greater than their respective threshold value, then that appliance is considered as ON otherwise it is OFF.

House Number	F	WM	DW	MW	K
House 3	50	892	1042	554	801
House 11	41	1000	1098	436	782

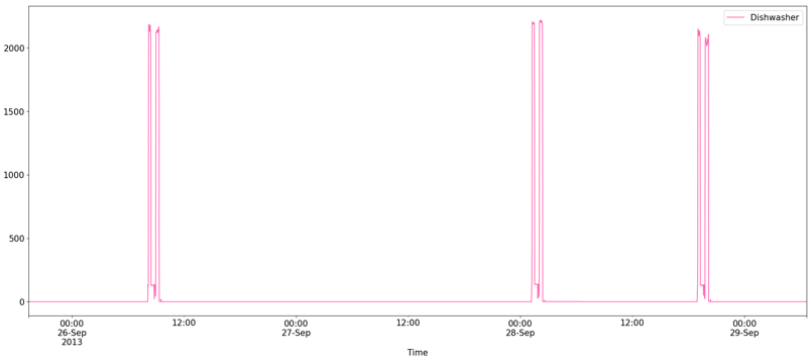
Naturally, the REFIT dataset contains only the consumption information of the five appliances. The operational states of the appliances are derived from the consumption values of appliance by using the Middle Point Thresholding technique (MPT) [18]. Once the threshold values for each appliance is obtained, for each instance if the consumption of an appliance for a house is equal or greater than the respective threshold value then that appliance is set to ON state (value 1 is assigned), otherwise the appliance is set to OFF state (value 0 is assigned). Table 4.2 presents the threshold values of five appliances from house 3 and 11.



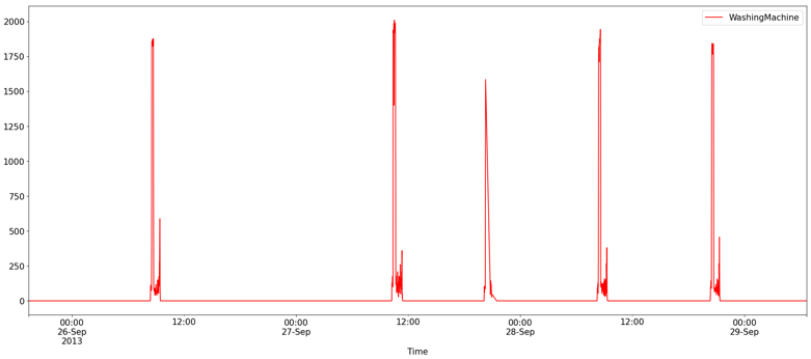
(a)


FridgeFreezer
400
350
300
250
200
150
100
50
0
00:00
26-Sep
2013
12:00
00:00
27-Sep
12:00
00:00
28-Sep
12:00
00:00
29-Sep
Time

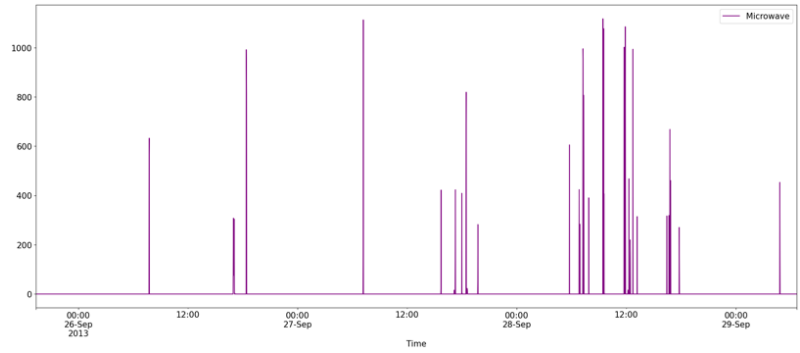
(b)


Dishwasher
2000
1500
1000
500
0
00:00
26-Sep
2013
12:00
00:00
27-Sep
12:00
00:00
28-Sep
12:00
00:00
29-Sep
Time

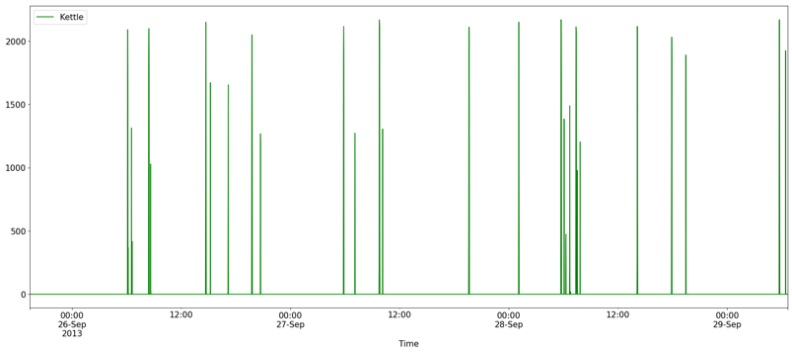
(c)


WashingMachine
2000
1750
1500
1250
1000
750
500
250
0
00:00
26-Sep
2013
12:00
00:00
27-Sep
12:00
00:00
28-Sep
12:00
00:00
29-Sep
Time

(d)


Microwave
1000
800
600
400
200
0
00:00
26-Sep
2013
12:00
00:00
27-Sep
12:00
00:00
28-Sep
12:00
00:00
29-Sep
Time

(e)



(f)

Figure 4.1: Aggregated power load along with consumption value of five appliances for 48 hours of House 3 data. (a) represents the aggregate power load, (b) shows consumption value of fridge-freezer, (c) is for dishwasher, (d) is for washing machine, (e) for microwave and (f) for kettle.

4.2 Experimental Setup

The Receptive field of TCN network should be large for the output of the respective network to receive wider range of input information. Large receptive field of TCN is commonly achieved by increasing the number of hidden layers but this will increase the overall computational complexity, resulting in more consumption of resources such as memory. This research aims to avoid such consequences from occurring so the alternate approach of choosing right filter size and dilation factor is performed here to increase the size of the TCN receptive field. For the proposed approach, the filter size is set to 3 and dilation factor for each residual block is set to 2^i where i is the residual block number. The TCN network of the proposed model comprised Seven TCN residual blocks with each block having 128 filters for hidden layers. The final layer is a fully connected layer from which the final appliance states and power consumption are obtained. The spatial dropout rate is set to 0.2. The experiment is performed in Mac Mini M1 device having a Ram capacity of 8 GB and a storage of 512 GB. The device is composed of 8 core CPU and 8 core GPU.

4.3 Result Discussion

After the proposed model is trained and evaluated twice by data of house 3 and 11 separately, the regression task of the model is evaluated using MAE score and the classification task is evaluated using F1-score. The performance of the proposed model is also compared to the performance of the method proposed in [18] where a model comprised of CNN and LSTM architecture is used to obtain the appliance state and power consumption from the aggregated load signal. The benchmarking model is trained with the same set of data (house 3 and 11 of Refit dataset) which is used towards the training of the proposed model.

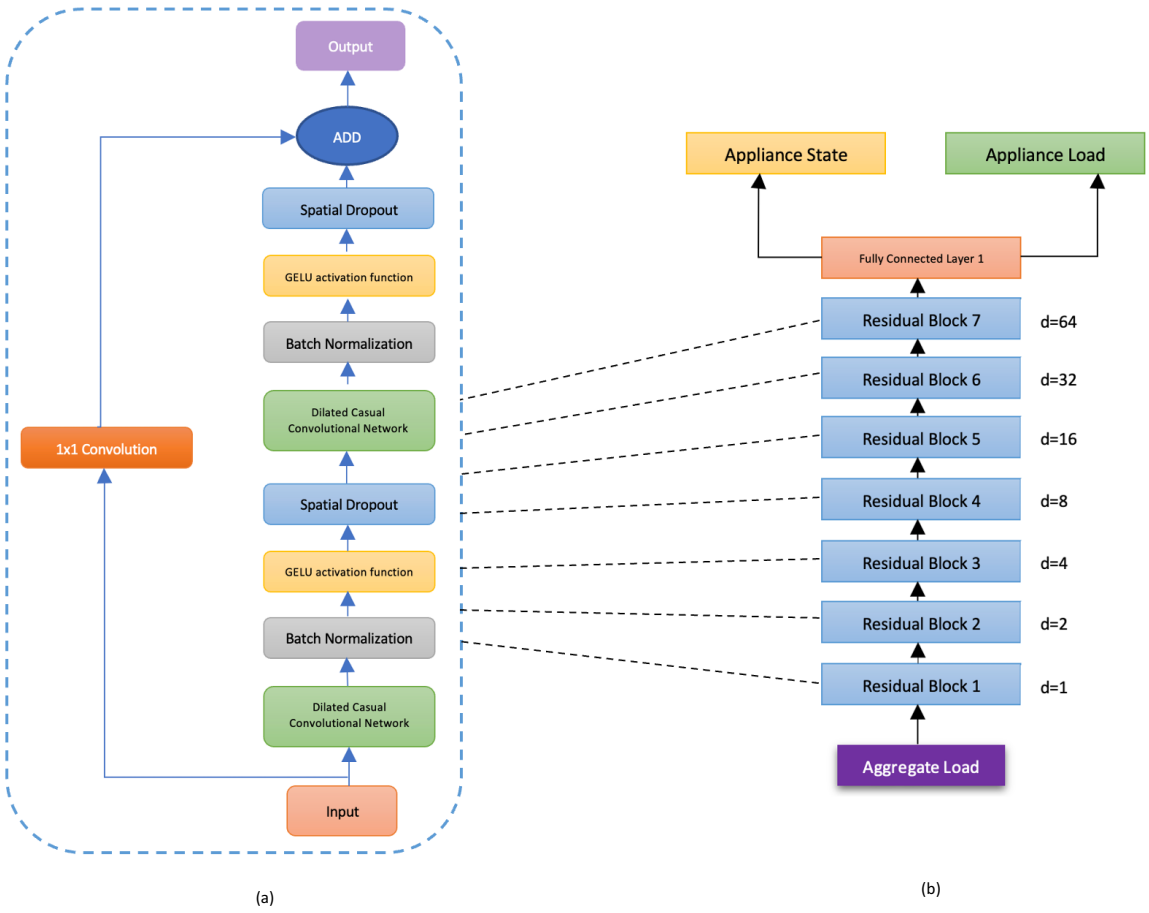


Figure 4.2: (a) Residual Block of the proposed TCN NILM model. (b) The architecture of the proposed TCN NILM model.

From Table 4.3 it is observed that the MAE scores based on the regression task of proposed TCN model for estimating power consumed by all five appliances of house 3 dataset is significantly better than the benchmarking LSTM-CNN model. The model scored an average MAE score of 14.35 for all the five appliances with overall MAE score of 15.36. The washing machine had the lowest MAE score of 8.67. The benchmarking method had an average MAE score of 18.17, with overall score of 25.67 for house 3. In terms of model trained and tested by house 11, similar observation is made where again the proposed TCN model outperformed the benchmarking approach in terms of MAE score for all the five appliances. The average MAE score of all the appliances for proposed model is 15.44 while the benchmarking method had MAE score of 18.86. Overall, the MAE score of the proposed model is 16.42 which is better than the benchmarking MAE score of 27.28.

Table 4.3: MAE score comparison between the proposed TCN model and the LSTM-CNN model

	Model	F	WM	DW	MW	K	Overall Score	Average Score
MAE score for House 3	TCN [proposed]	**10.95**	**8.76**	**15.97**	**14.43**	**21.68**	**15.36**	**14.35**
	LSTM-CNN	17.66	11.90	17.42	18.56	25.35	25.67	18.17
MAE score for House 11	TCN [proposed]	**8.41**	**10.38**	**18.52**	**12.93**	**26.78**	**16.42**	**15.44**
	LSTM-CNN	9.88	11.67	22.60	15.32	31.46	27.28	18.86

Table 4.4: F1-score comparison between the proposed TCN model and the LSTM-CNN model

	Model	F	WM	DW	MW	K	Overall Score	Average Score
F1-score for House 3	TCN [proposed]	**0.92**	**0.95**	**0.95**	**0.97**	**0.98**	**0.97**	**0.96**
	LSTM-CNN	0.83	0.91	0.94	0.88	0.92	0.91	0.89
F1-score for House 11	TCN [proposed]	**0.94**	**0.97**	**0.98**	**0.98**	**0.96**	**0.97**	**0.97**
	LSTM-CNN	0.88	0.92	0.89	0.93	0.95	0.90	0.92

When considering the classification NILM task, the proposed model again outperformed the benchmarking method in F1-score for classifying appliance operational state. For house 3, the average F1-score of the proposed model for all five appliance is 0.96 which is significantly higher than the average score of the benchmark LSTM-CNN model which is 0.89. Kettle had the highest F1-score of 0.98. Likewise for house 11 data, similar performance of the proposed model is observed where the TCN model outperformed the LSTM-CNN benchmark model for classifying appliance states. The proposed model had an overall score of 0.97 with an average 0.97 F1-score for all five appliances. Meanwhile the benchmark method overall scored 0.90 with average F1-score of 0.92 for all five appliances. The main reason of superiority of TCN model over LSTM-CNN model is for the option of having large receptive field which allows output to receive more input information, which makes the proposed model significantly better for the NILM regression and classification task.

Chapter 5

Conclusion

This thesis elaborates three different machine and deep learning models for Non-intrusive load monitoring tasks. The three proposed models fall under the category of supervised and semi-supervised learning in predicting power and estimating operational states of various appliances of various datasets. The models were evaluated using different evaluation metrics.

In chapter 2, a set of demographic parameters are extracted from a novel NILM data set from Grenoble INP in France. Instead of just active power reading, these demographic parameters are used for model training for NILM. The initial part of the research used six traditional regression algorithms to train model for estimating appliance power consumption. Eight train-test scenarios were used to observe how the selected ML algorithms estimated power consumption under different situations. The parameters of the models were tuned by grid search technique. The result of the evaluation of the different ML models constructed by the eight train-test scenarios aided in understanding instead of one algorithm generating favourable outcomes for multiple appliance consumption estimation, different algorithms are more suitable for different appliances at the same time. Then, a novel Bayesian Optimized Ensemble regressor model for non-intrusive load monitoring is proposed for estimating appliance level power consumption using active power and demographic parameters as input features. Based on R^2 and MAE score the proposed model outperformed the two benchmark approaches as well as the model generated by scenario 1 of the former part of this research. The proposed model showed dominance in estimating accurate device level consumption for Grenoble NILM dataset in contrast to the other benchmarking models. The idea of using demographic parameters for estimating appliance power consumption, understanding that instead of one algorithm, different algorithms might be suitable for different appliances, adopting various train-testing scenarios to understand appliance usage and finally using the proposed novel Bayes-ensemble regressor model for non-intrusive load monitoring will help towards the implementation of more efficient load monitoring systems.

Chapter 3 introduces a novel framework that utilizes deep learning and semi-supervised multi-label method for Non-Intrusive Load Monitoring. Temporal Convolution Neural Networks and Long-Short Term Memory are used as deep learning techniques. Mean Teacher model is used as the semi-supervised architecture. NILM datasets come with the individual appliance consumption record and for classification NILM tasks, the individual power load records are changed to operational status empirically without properly analyzing the data. Therefore, the effects of two appliance thresholding techniques namely – Middle Point thresholding and Variance Sensitive thresholding on the proposed model is thoroughly examined. The proposed methods were tested with three public NILM datasets. Multiple case studies and performance comparison helped

to conclude that proposed semi-supervised TCN LSTM model trained with dataset having operational status derived from Middle Point thresholding technique, performed better than the two state-of-the-art benchmarking approaches. Moreover, the proposed framework allows for the use of both labeled and unlabeled data to improve power consumption disaggregation and mitigate the challenges in availability of labeled NILM datasets.

Chapter 4 presented a novel NILM approach which combined classification and regression to predict power consumption along with on/off state of five different appliances of two houses in Refit dataset. The proposed model used TCN architecture to perform the NILM regression and classification task. The proposed approach is shown to have high accuracy when tested against the performance of the benchmarking technique which was comprised of LSTM-CNN architectures. Even on a generic computer with limited ram capacity, the approach is trained and evaluated smoothly.

Potential future works could be devoted to developing dedicated semi-supervised NILM methods for detecting appliances which have more than two operational states. Novel NILM techniques could be introduced to deal with high frequency load data. Finally, researchers can also focus on improving the accuracy of NILM algorithms using the integration of audio and vibration data: In addition to power data, appliances also emit audio and vibration signals that can provide additional information about their operation. Researchers could explore how these signals could be used in combination with power data to improve the accuracy of NILM algorithms. For example, audio signals could be used to identify specific sounds associated with different appliances, while vibration data could be used to detect changes in appliance operation.

Bibliography

[1] Looney, B. (2020). "Statistical Review of World Energy 2020, 69th," ed: Edition.

[2] Faustine, A., Mvungi, N. H., Kaijage, S., & Michael, K. (2017). A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem. arXiv preprint arXiv:1703.00785.

[3] Dong, K., Dong, X., & Jiang, Q. (2020). How renewable energy consumption lower global CO2 emissions? Evidence from countries with different income levels. The World Economy, 43(6), 1665-1698.

[4] Armel, K. C., Gupta, A., Shrimali, G., & Albert, A. (2013). Is disaggregation the holy grail of energy efficiency? The case of electricity. Energy policy, 52, 213-234.

[5] Darby, S., Liddell, C., Hills, D., & Drabble, D. (2015). Smart metering early learning project: synthesis report.

[6] Wagner, L., Ross, I., Foster, J., & Hankamer, B. (2016). Trading off global fuel supply, CO2 emissions and sustainable development. PloS one, 11(3), e0149406.

[7] Yang, Y., Yuan, J., Xiao, Z., Yi, H., Zhang, C., Gang, W., & Hu, H. (2021). Energy consumption characteristics and adaptive electricity pricing strategies for college dormitories based on historical monitored data. Energy and Buildings, 245, 111041.

[8] Batra, N., Singh, A., & Whitehouse, K. (2015, November). If you measure it, can you improve it? exploring the value of energy disaggregation. In Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments (pp. 191-200).

[9] Hart, G. W. (1992). Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12), 1870-1891.

[10] Shin, C., Joo, S., Yim, J., Lee, H., Moon, T., & Rhee, W. (2019, July). Subtask gated networks for non-intrusive load monitoring. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 1150-1157).

[11] Revuelta Herrero, J., Lozano Murciego, Á., López Barriuso, A., Hernández de la Iglesia, D., Villarrubia González, G., Corchado Rodríguez, J. M., & Carreira, R. (2018). Non intrusive load monitoring (nilm): A state of the art. In Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection-15th International Conference, PAAMS 2017 15 (pp. 125-138). Springer International Publishing.

[12] Bonfigli, R., Squartini, S., Fagiani, M., & Piazza, F. (2015, June). Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview. In 2015 IEEE 15th international conference on environment and electrical engineering (EEEIC) (pp. 1175-1180). IEEE.

[13] Barsim, K. S., & Yang, B. (2015, December). Toward a semi-supervised non-intrusive load monitoring system for event-based energy disaggregation. In *2015 IEEE global conference on signal and information processing (GlobalSIP)* (pp. 58-62). IEEE. doi: 10.1109/GlobalSIP.2015.7418156.

[14] Li, D., & Dick, S. (2017, July). A graph-based semi-supervised learning approach towards household energy disaggregation. In *2017 IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE) (pp. 1-7). IEEE. doi: 10.1109/FUZZ-IEEE.2017.8015650

[15] Precioso, D., & Gómez-Ullate, D. (2022). Thresholding Methods in Non-Intrusive Load Monitoring to Estimate Appliance Status. doi: 10.21203/rs.3.rs-1923023/v1

[16] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[17] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. doi: 10.48550/arXiv.1803.01271.

[18] Precioso, D., & Gómez-Ullate, D.: NILM as a regression versus classification problem: the importance of thresholding. arXiv preprint arXiv:2010.16050 (2020).

[19] Saraswat, G., Lundstrom, B., & Salapaka, M. V.: Scalable Hybrid Classification-Regression Solution for High-Frequency Nonintrusive Load Monitoring. arXiv preprint arXiv:2208.10638 (2022).

[20] Naderian, S.: A Novel Hybrid Deep Learning Approach for Non-Intrusive Load Monitoring of Residential Appliance Based on Long Short Term Memory and Convolutional Neural Networks. arXiv preprint arXiv:2104.07809 (2021).

[21] Faustine, A., Pereira, L., Bousbiat, H., & Kulkarni, S. (2020, November). UNet-NILM: A deep neural network for multi-tasks appliances state detection and power estimation in NILM. In Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring (pp. 84-88).

[22] Kim, H., Marwah, M., Arlitt, M., Lyon, G., & Han, J. (2011, April). Unsupervised disaggregation of low frequency power measurements. In Proceedings of the 2011 SIAM international conference on data mining (pp. 747-758). Society for Industrial and Applied Mathematics.

[23] Kolter, J. Z., & Johnson, M. J. (2011, August). REDD: A public data set for energy disaggregation research. In Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA (Vol. 25, No. Citeseer, pp. 59-62).
[24] Buddhahai, B., & Makonin, S. (2021). A nonintrusive load monitoring based on multi-target regression approach. IEEE Access, 9, 163033-163042.
[25] Struyf, J., & Dzeroski, S. (2006). Constraint based induction of multi-objective regression trees. Lecture notes in computer science, 3933, 222-233.
[26] Kaselimi, M., Doulamis, N., Doulamis, A., Voulodimos, A., & Protopapadakis, E. (2019, May). Bayesian-optimized bidirectional LSTM regression model for non-intrusive load monitoring. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2747-2751). IEEE.
[27] Makonin, S., Popowich, F., Bartram, L., Gill, B., & Bajić, I. V. (2013, August). AMPds: A public dataset for load disaggregation and eco-feedback research. In 2013 IEEE electrical power & energy conference (pp. 1-6). IEEE.
[28] Lin, J., Ma, J., Zhu, J., & Liang, H. (2021). Deep domain adaptation for non-intrusive load monitoring based on a knowledge transfer learning network. IEEE Transactions on Smart Grid, 13(1), 280-292.
[29] Hadi, M. U., Suhaimi, N. H. N., & Basit, A. (2022). Efficient Supervised Machine Learning Network for Non-Intrusive Load Monitoring. Technologies, 10(4), 85.
[30] Timplalexis, C., Krinidis, S., Ioannidis, D., & Tzovaras, D. EMD and Gradient Boosting Regression for NILM at Residential Houses.
[31] Schirmer, P. A., Mporas, I., & Paraskevas, M. (2019, July). Evaluation of regression algorithms and features on the energy disaggregation task. In 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA) (pp. 1-4). IEEE.
[32] Piccialli, V., & Sudoso, A. M. (2021). Improving non-intrusive load disaggregation through an attention-based deep neural network. Energies, 14(4), 847.
[33] Konstantopoulos, C., Sioutas, S., & Tsichlas, K. (2022, June). Machine Learning Techniques for Regression in Energy Disaggregation. In Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings, Part I (pp. 356-366). Cham: Springer International Publishing.
[34] Rao, K. M., Ravichandran, D., & Mahesh, K. (2016). Non-intrusive load monitoring and analytics for device prediction. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, pp. 132-136).
[35] Laouali, I., Ruano, A., Ruano, M. D. G., Bennani, S. D., & Fadili, H. E. (2022). Non-intrusive load monitoring of household devices using a hybrid deep learning model through convex hull-based data selection. Energies, 15(3), 1215.
[36] Li, C., R. Yang, and H. Wang. (2022). Non-intrusive Load Monitoring in Industry Based on Gradient Boosting Algorithm. In 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 1523-1528). IEEE. doi:10.1109/CSCWD54268.2022.9776262.
[37] Precioso, D., & Gómez-Ullate, D. (2021, June). Non-Intrusive Load Monitoring using Multi-Output CNNs. In 2021 IEEE Madrid PowerTech (pp. 1-6). IEEE.
[38] Chen, K., Zhang, Y., Wang, Q., Hu, J., Fan, H., & He, J. (2019). Scale-and context-aware convolutional non-intrusive load monitoring. IEEE Transactions on Power Systems, 35(3), 2362-2373.
[39] Yang, W., Pang, C., Huang, J., & Zeng, X. (2021). Sequence-to-point learning based on temporal convolutional networks for nonintrusive load monitoring. IEEE Transactions on Instrumentation and Measurement, 70, 1-10.
[40] Mollel, R. S., Stankovic, L., & Stankovic, V. (2022, November). Using explainability tools to inform NILM algorithm performance: a decision tree approach. In Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (pp. 368-372).
[41] Hernandez, A. S., Ballado, A. H., & Heredia, A. P. D. (2021, June). Development of a non-intrusive load monitoring (nilm) with unknown loads using support vector machine. In 2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS) (pp. 203-207). IEEE.
[42] Yang, C. C., Soh, C. S., & Yap, V. V. (2018). A systematic approach in appliance disaggregation using k-nearest neighbours and naive Bayes classifiers for energy efficiency. Energy Efficiency, 11, 239-259.
[43] Hock, D., M. Kappes, and B. Ghita. (2018). Non-intrusive appliance load monitoring using genetic algorithms. In IOP Conference Series: Materials Science and Engineering (Vol. 366, No. 1, p. 012003). IOP Publishing. doi:10.1088/1757-899X/366/1/012003.
[44] Zhao, B., L. Stankovic, and V. Stankovic. (2016). On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. IEEE Access, 4, 1784-1799. doi:10.1109/ACCESS.2016.2557460.
[45] Jia, Z., Yang, L., Zhang, Z., Liu, H., & Kong, F. (2021). Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring. International Journal of Electrical Power & Energy Systems, 129, 106837. doi: 10.1016/j.ijepes.2021.106837.

[46] Alami, M., Decock, J., Kaddah, R., & Read, J. (2022, September). Conv-NILM-Net, a causal and multi-appliance model for energy source separation. In European Conference on Machine Learning (ECML), MLBEM Workshop.
[47] Qian, Y., Yang, Q., Li, D., An, D., & Zhou, S. (2021, May). An Improved Temporal Convolutional Network for Non-intrusive Load Monitoring. In *2021 33rd Chinese Control and Decision Conference (CCDC)* (pp. 2557-2562). IEEE. doi: 10.1109/CCDC52312.2021.9601611.
[48] Liu, Y., Qiu, J., Lu, J., Wang, W., & Ma, J. (2021). A Single-to-Multi Network for Latency-Free Non-Intrusive Load Monitoring. *IEEE Transactions on Network Science and Engineering, 9*(2), 755-768. doi: 10.1109/TNSE.2021.3132309.
[49] Zhang, Z., Li, Y., Duan, J., Guo, Y., Hou, Z., Duan, Y., ... & Rehtanz, C. (2022). A Multi-State Load State Identification Model Based on Time Convolutional Networks and Conditional Random Fields. *IEEE Transactions on Artificial Intelligence*. doi:10.1109/TAI.2022.3203685.
[50] Zhou, X., Li, S., Liu, C., Zhu, H., Dong, N., & Xiao, T. (2021). Non-intrusive load monitoring using a cnn-lstm-rf model considering label correlation and class-imbalance. *IEEE Access, 9, 84306-84315*. doi:10.1109/ACCESS.2021.3087696.
[51] Kim, J. G., & Lee, B. (2019). Appliance classification by power signal analysis based on multi-feature combination multi-layer LSTM. *Energies, 12*(14), 2804.
[52] de Diego-Otón, L., Fuentes-Jimenez, D., Hernández, Á., & Nieto, R. (2021, May). Recurrent lstm architecture for appliance identification in non-intrusive load monitoring. In *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 1-6). IEEE. doi:10.1109/I2MTC50364.2021.9460046.
[53] Li, D., Sawyer, K., & Dick, S. (2015, August). Disaggregating household loads via semi-supervised multi-label classification. In 2015 *Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)* (pp. 1-5). IEEE.
[54] Gillis, J. M., & Morsi, W. G. (2016). Non-intrusive load monitoring using semi-supervised machine learning and wavelet design. *IEEE Transactions on Smart Grid*, 8(6), 2648-2655. doi: 10.1109/TSG.2016.2532885.
[55] Iwayemi, A., & Zhou, C. (2015). SARAA: Semi-supervised learning for automated residential appliance annotation. *IEEE Transactions on Smart Grid, 8*(2), 779-786. doi: 10.1109/TSG.2015.2498642.
[56] Yang, Y., Zhong, J., Li, W., Gulliver, T. A., & Li, S. (2019). Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids. *IEEE Transactions on Industrial Informatics, 16*(11), 6892-6902. doi: 10.1109/TII.2019.2955470.
[57] Hur, C. H., Lee, H. E., Kim, Y. J., & Kang, S. G. (2022). Semi-Supervised Domain Adaptation for Multi-Label Classification on Nonintrusive Load Monitoring. *Sensors, 22*(15), 5838. doi:10.3390/s22155838.
[58] Fatouh, A. M., Nasr, O. A., & Eissa, M. M. (2018, August). New semi-supervised and active learning combination technique for non-intrusive load monitoring. In *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)* (pp. 181-185). IEEE. doi:10.1109/SEGE.2018.8499498.
[59] da Silva Nolasco, L., Lazzaretti, A. E., & Mulinari, B. M. (2021). DeepDFML-NILM: A new CNN-based architecture for detection, feature extraction and multi-label classification in NILM signals. IEEE Sensors Journal, 22(1), 501-509.
[60] Devlin, M., & Hayes, B. P. (2019, August). Non-intrusive load monitoring using electricity smart meter data: A deep learning approach. In 2019 IEEE Power & Energy Society General Meeting (PESGM) (pp. 1-5). IEEE.
[61] Xiao, P., & Cheng, S. (2019). Neural network for nilm based on operational state change classification. arXiv preprint arXiv:1902.02675.
[62] Kim, J., & Kim, H. (2016, July). Classification performance using gated recurrent unit recurrent neural network on energy disaggregation. In 2016 international conference on machine learning and cybernetics (ICMLC) (Vol. 1, pp. 105-110). IEEE.
[63] Zhang, C., Zhong, M., Wang, Z., Goddard, N., & Sutton, C. (2018, April). Sequence-to-point learning with neural networks for non-intrusive load monitoring. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).
[64] Kelly, J., & Knottenbelt, W. (2015, November). Neural nilm: Deep neural networks applied to energy disaggregation. In Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments (pp. 55-64).
[65] Serafini, L., Tanoni, G., Principi, E., Spinsante, S., & Squartini, S. (2022, August). A Multiple Instance Regression Approach to Electrical Load Disaggregation. In 2022 30th European Signal Processing Conference (EUSIPCO) (pp. 1666-1670). IEEE.
[66] Lu, Z., Cheng, Y., Zhong, M., Luan, W., Ye, Y., & Wang, G.: LightNILM: lightweight neural network methods for non-intrusive load monitoring. In Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (pp. 383-387) (2022, November).

[67] Commercial Buildings Energy Consumption Survey, Consumption and Expenditures Highlights, U.S. Energy Information Administration, December 2022, https://www.eia.gov/cbecs.
[68] Delinchant, B., Martin, G., Laranjeira, T., Muhammad, S., & Wurtz, F. (2021, July). Machine Learning on Buildings Data for Future Energy Community Services. In SGE 2021-Symposium de Génie Electrique.
[69] Martin Nascimento, G.F. (2022). Optimization of resources and consumption of smart buildings with a view to energy efficiency, (PhD Thesis, UGA/UFSC). https://thares.univ-grenoble-alpes.fr/2022GRALT078.pdf.
[70] Martin Nascimento, G. F., Delinchant, B., Wurtz, F., Kuo-Peng, P., Jhoe Batistela, N., & Laranjeira, T. G. E. (2020). Electricity Consumption Data of a Tertiary Building. Mendeley Data, 1.
[71] GreEn-ER API, available online: https://mhi-srv.g2elab.grenoble-inp.fr/django/API.
[72] Hodencq, S., Delinchant, B., & Wurtz, F. (2021, September). Open and Reproducible Use Cases for Energy (ORUCE) methodology in systems design and operation: a dwelling photovoltaic self-consumption example. In Building Simulation 2021.
[73] McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. Python for high performance and scientific computing, 14(9), 1-9.
[74] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.
[75] Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.
[76] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
[77] Brownlee, J. (2016). A gentle introduction to the gradient boosting algorithm for machine learning. Machine Learning Mastery, 21.
[78] Shi, H. (2007). Best-first decision tree learning (Doctoral dissertation, The University of Waikato).
[79] Ahamed, B. S. (2021). Prediction of type-2 diabetes using the LGBM classifier methods and techniques. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 223-231.
[80] Bahmani, MJ. (2022). Understanding LightGBM Parameters (and How to Tune Them). The MLOps Blog. https://neptune.ai/blog/lightgbm-parameters-guide.
[81] Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. Image Rochester NY, 4(1), 2-3.
[82] Brachmann, E., & Rother, C. (2019). Neural-guided RANSAC: Learning where to sample model hypotheses. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4322-4331).
[83] Reinhardt, A., Baumann, P., Burgstahler, D., Hollick, M., Chonov, H., Werner, M., & Steinmetz, R. (2012, October). On the accuracy of appliance identification based on distributed load metering data. In 2012 Sustainable Internet and ICT for Sustainability (SustainIT) (pp. 1-9). IEEE.
[84] Maasoumy, M., Sanandaji, B., Poolla, K., & Vincentelli, A. S. (2013, December). Berds-berkeley energy disaggregation data set. In Proceedings of the Workshop on Big Learning at the Conference on Neural Information Processing Systems (NIPS) (Vol. 7).
[85] Batra, N., Parson, O., Berges, M., Singh, A., & Rogers, A. (2014). A comparison of non-intrusive load monitoring methods for commercial and residential buildings. arXiv preprint arXiv:1408.6595.
[86] MacQueen, I. (1967). Some methods for classifiction and analysis of multivariate observations. In *Proceedings 5th Berkeley Symposium on Mathematical Statistics Problems* (pp. 281-297).
[87] Desai, S., Alhadad, R., Mahmood, A., Chilamkurti, N., & Rho, S. (2019). Multi-state energy classifier to evaluate the performance of the nilm algorithm. *Sensors, 19*(23), 5236. doi: 10.3390/s19235236.
[88] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.
[89] Srivastava, S., & Lessmann, S. (2018). A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Solar Energy, 162*, 232-247. doi: 10.1016/j.solener.2018.01.005.
[90] Zhang, M., Li, J., Li, Y., & Xu, R. (2021). Deep learning for short-term voltage stability assessment of power systems. *IEEE Access, 9*, 29711-29718. doi: 10.1109/ACCESS.2021.3057659.
[91] Ko, M. S., Lee, K., Kim, J. K., Hong, C. W., Dong, Z. Y., & Hur, K. (2020). Deep concatenated residual network with bidirectional LSTM for one-hour-ahead wind power forecasting. *IEEE Transactions on Sustainable Energy, 12*(2), 1321-1335. doi: 10.1109/TSTE.2020.3043884.
[92] Wang, K., Qi, X., & Liu, H. (2019). Photovoltaic power forecasting based LSTM-Convolutional Network. *Energy, 189*, 116225. doi: 10.1016/j.energy.2019.116225.

[93] Wang, W., Hong, T., Xu, X., Chen, J., Liu, Z., & Xu, N. (2019). Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm. *Applied Energy, 248*, 217-230. doi: 10.1016/j.apenergy.2019.04.085.

[94] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing, 45*(11), 2673-2681. doi: 10.1109/78.650093.

[95] Liu, S., Lee, K., & Lee, I. (2020). Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. *Knowledge-Based Systems, 197*, 105918. doi: 10.1016/j.knosys.2020.105918.

[96] Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing, 337*, 325-338. doi: 10.1016/j.neucom.2019.01.078.

[97] Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems, 30*.

[98] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[99] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

[100] Kelly, Jason P. MD; James, Michelle A. MD. Radiographic Outcomes of Hemiepiphyseal Stapling for Distal Radius Deformity Due to Multiple Hereditary Exostoses. *Journal of Pediatric Orthopaedics 36(1)*:p 42-47, January 2016. | DOI: 10.1097/BPO.0000000000000394.

[101] Murray, D., Liao, J., Stankovic, L., Stankovic, V., Hauxwell-Baldwin, R., Wilson, C., ... & Firth, S. (2015). A data management platform for personalised real-time energy feedback.

[102] Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. *arXiv preprint arXiv:1402.1892*.

[103] Harbecke, D., Chen, Y., Hennig, L., & Alt, C. (2022). Why only Micro-F1? Class Weighting of Measures for Relation Classification. *arXiv preprint arXiv:2205.09460*.