Demand-side analytics for sustainable energy

By

Megha Gaur

Under the supervision of Dr. Angshul Majumdar

Computer Science and Engineering

Indraprastha Institute of Information Technology Delhi

New Delhi– 110020

October, 2019

DEMAND-SIDE ANALYTICS FOR SUSTAINABLE ENERGY

BY

MEGHA GAUR

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

COMPUTER SCIENCE AND ENGINEERING

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI– 110020

OCTOBER, 2019

Certificate

This is to certify that the thesis titled *Demand-side analytics for sustainable energy* being submitted by *Megha Gaur* to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

November, 2019

Dr. Angshul Majumdar

Indraprastha Institute of Information Technology Delhi

New Delhi 110020

Abstract

The rapidly growing demand for energy poses one of the biggest challenges in our society. This challenge is critical not just for the power utilities but also for the environment as it leads to increased carbon footprint and climate change. There have been concerted efforts towards sustainable energy. One of the main contributions towards effective energy management is the deployment of smart grid technology, in particular including demand-side management (DSM) techniques. DSM involves tasks like non-intrusive load monitoring or energy disaggregation, demand forecasting, anomaly detection, outage management, etc. These tasks empower customers to make more informed decisions about their energy consumption, adjusting both the quantity and timing of their energy usage. The goal is to reduce the overall energy consumption and also to save the cost of building an additional generation capacity to meet the critical peak demands.

In the first DSM task, the problem of energy disaggregation is explored. Energy disaggregation is a single channel blind source separation problem where the task is to estimate the consumption of each electrical appliance given the total meter reading. In this work, two issues that are often overlooked - the problem of missing data, and, the problem of outliers are addressed. The first problem arises when the smart-meter cannot transmit the readings to the server owing to the failure of wireless communication interfaces. The second problem arises from transients, surges and other non-linear effects. A modified dictionary learning based disaggregation framework is used to address these problems. A recent work [1] in this area showed that instead of employing the usual Euclidean norm cost function for dictionary learning, better results can be achieved by learning the dictionaries in a robust fashion by employing a l_1-norm cost function. This is because energy data is corrupted by large but sparse outliers. In the second work, an approach to improve robust dictionary learning is proposed. This is done by imposing a low-rank penalty on the learned

coefficients. The ensuing formulation is solved using a combination of Split Bregman and Majorization Minimization approach. In the third work, the existing work of robust dictionary learning is extended by modeling non-linear perturbations as sparse error and applying robust versions of dictionary learning for disaggregation. On top of the basic (unsupervised) robust dictionary learning formulation, two supervised variants are proposed. In the first supervision, dictionaries are learned such that they are incoherent; this ensured that the dictionaries from different appliances look different from each other. In the second formulation, discriminating sparse codes are introduced, such that the codes generated for each appliance would not look alike. In the final work in the area of energy disaggregation, a new method based on the transform learning formulation is introduced. Several recent techniques, such as discriminative sparse coding, powerlet disaggregation, and deep sparse coding, are based on the synthesis dictionary learning/sparse coding approach. The proposed method is based on its analysis equivalent. The theoretical advantage of the analysis dictionary compared to its synthesis counterpart is that the former can learn from fewer training samples - this has implications in reducing the cost of energy disaggregation.

The next contribution to DSM in this thesis is load forecasting. It is a technique used by power utilities to predict the power needed to meet the demand and supply equilibrium. In the first contribution towards this task, the problem of one-day-ahead short-term load forecasting is considered. As the effect of weather, as well as prior consumptions, are nonlinear functions, the formulation is based on non-linear Kalman filtering algorithms. In the second work, the focus is to improve the accuracy of building-level demand forecasting. For the said purpose, a regressing deep dictionary learning approach is proposed. There are two versions of the algorithm - synthesis and analysis. In this work, point forecasting as well as profile forecasting is performed.

The last contribution to DSM is to detect abnormal energy consumption behaviour in residential buildings. Understanding, identifying, and addressing abnormal energy consumption in buildings can lead to energy savings and detection of faulty appliances. This work investigates two key challenges found in energy anomaly detection research: (1) the lack of labeled ground truth, and (2) the lack of consistent performance accuracy metrics. In the first challenge, labeled ground truth is imperative for training and benchmarking algorithms to detect anomalies. In the second challenge, consistent performance accuracy

metrics are crucial to quantifying how well, algorithms perform against each other. Two approaches that help in the automatic annotation of the ground truth data from publicly available datasets are proposed: a statistical approach for short-term data, and a piecewise linear regression method for long-term data. Finally, we aim to detect anomalies that we define as power consumption during a power outage (negative anomaly) and power theft (positive anomaly). A robust principal component analysis (RPCA) technique for separating anomalies from the normal component is employed.

Dedication

To,
My loving parents and grandparents . . .

Declaration

"For all the papers, I formulated the problem with the help of the faculty member(s). I carried out all the experimentation by myself. I co-wrote the paper along with the faculty member(s)"

Acknowledgements

My Ph.D. journey at IIIT has been the most cherished and exciting time of my life. There are several people who have contributed to this journey either directly or indirectly. I would like to take this opportunity to express my deepest sense of gratitude to each one of them.

Firstly, my heartfelt gratitude goes to my advisor, Dr. Angshul Majumdar, who has been extraordinarily supportive, understanding and a great source of inspiration. I can not thank him enough for having confidence in me and especially for always being patient with me. I am very lucky to have a mentor like him, who has always been more than a family. His continuous support, motivation and encouragement helped me to reach the stage of writing this thesis. I am grateful to him for giving me enough platforms in terms of internships, conferences, seminars, etc., to collaborate and enjoy the process of learning.

I am very grateful to my external reviewers, Dr. Lina Stankovic, Dr. Behnaz Ghoraani and Dr. Aurobinda Routray for their detailed, insightful and timely feedback. I am also thankful to my internal committee members, Dr. Amarjeet, Dr. P.B Sujit and Dr. Girish Chandra (TCS) for giving me honest and timely reviews on my work. My sincere gratitude to Dr. Ivan Bajic and Dr. Stephen Makonin for hosting me during my internship program at Simon Fraser University. A special thanks to Dr. Vinayak without whom I may not have reached this stage of writing my thesis. He gave me the courage and necessary guidance to cope up with the initial setbacks in my PhD.

I am grateful to the Indraprastha Institue of Information Technology - Delhi for providing excellent infrastructure and research environment. I would also like to thank the man behind this institute, Dr. Pankaj Jalote for keeping the research standard of this institute at par with other top research institutes in the world. Many thanks to Tata Consultancy Services (TCS) for financially supporting my Ph.D. life.

This thesis would be incomplete without the mention of my support system from the Salsa lab. I thank all my friends and colleagues in this journey, Hemant, Anuprriya, Monalisa, Nipun, Shikha, Vanika, Milan, Haroon, Jyoti, Shalini, Pooja and many more. I have learned something or the other from each one of them. Special thanks to Neha for being my go-to friend, Hareesh for sending me gentle reminders to not drift away from Ph.D., Dr. Apala for being my go-to friend in Vancouver and Mehrdad for encouraging me to challenge myself while I was learning to Ski.

Finally, I would like to thank my loving family for its constant love and support. I am grateful to my parents for imparting good education and values in me, for giving me the freedom to navigate my life as per my choice, my brother, Akash for being my buddy who I used to rely on when I was learning to code, his wife, Shveta for being a loving and caring sister-in-law and most importantly for giving us a bundle of joy, Vivaan (nephew), my parents-in-law for showing great understanding in me and my brother-in-law, Tarun for always cheering me up. I can not thank enough my husband, Major Varun, for always being my pillar of strength. During all these years when we stayed away in different states and sometimes even in different time-zones, I always felt close to him. He has been strong enough to bear all my tantrums, mood swings, failures and setbacks. All these years and not once did he worry me about the turmoil at his work front. It would only be fair to say that we both have learned from our share of experiences in life and evolved together as a couple.

Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
AAL	Ambient Assisted Living
ALM	Appliance Load Monitoring
ALS	Alternating Least Squares
AD	Anomaly Detection
BCS	Blind Compressed Sensing
CS	Compressed Sensing
DSM	Demand Side Management
DSC	Deep Sparse Coding
$discSC$	Discriminative Sparse Coding
DR	Demand Response
DL	Dictionary Learning
DF	Demand Forecasting
DDL	Disaggregating Dictionary Learning
EKF	Extended Kalman Filters
FSM	Finite State Machines
$FHMM$	Factorial Hidden Markov Model
GSP	Graph Signal Processing
HAN	Home Area Network
$HEMS$	Home Energy Management Systems
HUE	The Hourly Usage of Energy Dataset
IDL	Incoherent Dictionary Learning
ILM	Intrusive Load Monitoring
IEA	International Energy Agency
$IBLEND$	Indian Buildings Energy Consumption Dataset
MMV	Multiple Measurement vector
MM	Majorization Minimization
$NILM$	Non-Intrusive Load Monitoring
$NIALM$	Non-Intrusive Appliance Load Monitoring

$NILMTK$	Non-Intrusive Load Monitoring Toolkit
PED	Powerlet Energy Disaggregation
RDL	Robust Dictionary Learning
RSC	Robust Sparse Coding
SVT	Singular Value Thresholding
$STLF$	Short Term Load Forecasting
SC	Sparse Coding
$SMPS$	Switched-Mode Power Supply
RSC	Robust Sparse Coding
$RPCA$	Robust Principal Component Analysis
$REDD$	Reference Energy Disaggregation Dataset
UKF	Unscented Kalman Filters
WAN	Wide Area Network

Chapter 1

Introduction

1.1 Background

With the growth of the economy, the demand for energy has grown substantially. This rapidly growing demand for energy poses one of the biggest challenges in our society. In India, the single largest emitter of carbon dioxide (CO2) is the power sector, but from the demand-side, the largest emissions come from the industrial sector followed by the transport sector [4]

According to the International Energy Agency (IEA) report [5], from 1990 to 2008, the average energy use per person got increased by 10% while the world population got increased by 27% but energy consumption got increased by 39%. The report also mentions the growth of average global power demand by 20% from 2008 to 2012. If the trend continues to remain the same, then most of the climate models predict that the earth's temperature will increase by at least 5 degrees. This change could cause ecological disasters on a global

scale. The high level of energy intensity in some of the sectors is a matter of serious concern. In such a case, efficient use of energy resources and their conservation assume tremendous significance. It is essential to avoid wasteful energy consumption for sustainable development. A significant reduction in the energy wastage can be achieved through fine-grained monitoring of energy consumption and feeding back this information to the customers.

Comprehensive research [6] of more than 60 feedback studies suggest that maximum energy saving can be achieved using direct feedback mechanisms (real-time appliance-level consumption information) as opposed to indirect feedback mechanisms (such as monthly bills, weekly advice on energy usage). After this study, large scale deployment of smart meters was done by the government of the UK and the USA in the residential sector. Since traditional meters could only provide data at house-level granularity, research led to the creation of appliance load monitoring (ALM) to obtain finer granularity of data for precise demand response functionality. The two main approaches for ALM are Intrusive Load Monitoring (ILM) and Non-Intrusive Load Monitoring (NILM). ILM approaches require one or more than one sensor per appliance to perform load monitoring tasks, whereas NILM requires just one smart meter per house or a building for monitoring. They are alternatively referred to as distributed sensing and single point sensing methods respectively. Even though the ILM method is more accurate in measuring appliance-level energy consumption as compared to NILM techniques but due to the practical disadvantages like high costs, multiple sensor configurations, installation complexity, privacy invasion, etc., the

use of NILM is much more favored for large scale deployment.

Non-Intrusive (Appliance) Load Monitoring (NILM or NIALM), sometimes also called load or energy disaggregation, is an area in computational sustainability that discerns what electrical loads (appliances) are running within a physical area where the power is supplied by mains. By using the information about the different types of active loads, their running times, duration and the amount of power consumption, the user can make informed decisions leading to a reduction in power consumption. NILM serves as one of the techniques that empower power utilities to gain better insight into the breakdown of energy consumption for each household. With this availability of finer granularity of energy consumption data, utilities can efficiently formulate the demand-side activities. Our contributions to the field of NILM has been covered in chapter 2.

Amongst other techniques that enable utilities to manage energy requirements are load forecasting and anomaly detection. Power utilities have the ability to operate and manage supply to the end users with accurate demand forecasting [7, 8]. It increases the efficiency and revenue for the electrical generation and distribution companies by planning ahead of time. It also helps in the planning the future in terms of size, location and type of additional generation capacity in order to provide a reliable supply to the customers. By understanding demand, utilities can decide when to carry out maintenance tasks in a way that its impact on the customers is minimal. The contribution to this important field of load forecasting has been given in chapter 3.

Anomaly detection is another technique that provides utilities with information on faulty appliances running at customer's end. Anomaly is defined when the end-users energy consumption is abnormally high or abnormally low, that is it does not conform to the regular pattern. One way to achieve appliance specific consumption is through NILM. The other way is to use anomaly detection algorithms on meter-level energy data. A use-case scenario for this approach is when utilities want to minimize the number of customer complaint calls. In this scenario, customers call to complaint when their energy bill is unexpectedly high. To avoid these calls, utilities segregate customers based on their anomalous energy consumption patterns. The customers with high degree of anomalous activities or with high anomaly scores are contacted in advance to inform them about their high next month energy bill. This feedback given to end-users could lead to better energy utilization. The identification of abnormal energy usage patterns can lead to not just saving of energy but also increase in revenue for both customers and utilities. Upon identification of appliances running in the state of disrepair or used improperly, utilities can create alerts to either repair an appliance or suggest a more optimal use of it to its customers. A detailed discussion on this problem and the challenges that it poses can be found in chapter 4.

Let us now review what demand-side management (DSM) means and see how it can bring about a change in the end user's energy usage behavior.

1.2 Demand-side Management

Energy demand management or demand-side management (DSM) or demand response (DR) was introduced by the Electric Power Research Institute (EPRI) in the 1980s as a consequence of the energy crisis in 1973 and 1979. DSM is a global term that includes activities like load management, energy efficiency, energy savings, etc. According to the Federal Energy Regulatory Commission, demand response [9] is defined as, " Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of energy over time or to incentive payments designed to induce lower energy use at times of high wholesale market prices or when system reliability is jeopardized."

DSM is a set of interconnected programs that enable customers to make more informed decisions about their energy consumption, adjusting both the quantity and timing of their energy usage [10]. The objective of these programs is to reduce the overall energy consumption and also to save the cost of building an additional generation capacity to meet the peak demands.

1.2.1 Building blocks of demand-side programs

In the smart grid infrastructure, high-quality demand-side techniques have become indispensable to control energy consumption at the user-side. DSM comprises of the following principal programs -

- **Energy efficiency and conservation**: This program encourages customers to use less power to perform the same task in return for saving money. It involves a permanent reduction in power demand by replacing old with more efficient load intensive appliances like a water heater, air conditioners, refrigerators, etc., or by turning up the thermostat a few degrees to reduce air conditioning. NILM or load disaggregation is called the holy grail of energy efficiency [11]. It is the task of separating the whole energy signal of residential, commercial or industrial buildings into energy signals of individual appliances [12]. Apart from providing feedback to the customers, it also helps to detect malfunctioning of electrical devices, design energy incentives, forecasts demands. Studies [13] have shown that appliance specific information leads to much larger gains (reduces 12% average consumption in the residential sector) than the whole home data.

  In the Indian context, a study done by the Energy and Resource Institute (TERI) [14] discusses the potential of energy efficiency in various sectors such as irrigation in agricultural the sector, retrofitting in the existing industrial sector, lighting and space conditioning in residential and commercial sectors. With the changing consumption patterns, demand-side techniques need to be refocussed as well. For example, in Delhi, the change in the peak from evening or morning hours to midnight during August to September months is contributed to the heavy use of air conditioners in the households. One of the ways to manage such peaks is to produce extremely efficient air conditioners. Thus, with changing end-uses leading to energy

growth, the DSM plans too require modifications.

In order to maximize the gains from energy efficiency, we can also learn outcomes on energy efficiency measures from other countries. For example, Japan has developed a carbon reduction reporting system for small and medium enterprises (SME) involving both voluntary and mandatory reporting to the Tokyo Municipal Government (TMG). Based on the inputs, TMG educates the SMEs on energy efficiency to drive them in the right direction [15]. Similarly, in Curitiba, Brazil, an improved transport system has led to one of the lowest per capita gasoline consumption despite having the second highest car ownership rate [16].

- **Demand Response (DR) or load shifting**: It is a strategy used by power utilities to reduce or shift the energy consumption from peak hours of the day to off-peak hours in response to time-based rates or other financial incentives. It allows end-users to play an important role in the operation of the electric grid by choosing non-essential loads which can be shed by them or utilities directly, at peak times. It comprises of any reactive or preventive method to reduce, flatten or shift the demand curve. There are two common ways in which demand response events are executed by utilities including:

  Dynamic pricing: It uses variable energy rates to encourage customers' voluntary curtailment of energy usage during peak hours. Power utilities use a variety of pricing schemes including peak time rebates, critical peak pricing, and time of use rates to reduce usage. These actions are taken at the demand-side in response to particular conditions within the energy

system (such as peak period, network congestion or high prices). This also saves the cost of building additional generation capacity to meet demands at peak periods [17].

Let us see an example to check how demand response works. Let us assume the demand is at a peak in the afternoon from 2 pm to 6 pm. In order to maintain the uninterrupted power supply, utility companies buy power from service providers at a very high rate. As a result, they incur a loss.

Utilities buy power at rate = 11$/kWh

Utilities sell power to customers at rate = 2$/kWh

The loss incurred by utilities = 9$/kWh

To avoid or minimize this loss, utilities conduct events based on certain conditions like supply during peak demand, failure of power supply, natural disasters like earthquake, tornadoes, cyclones and other types of severe storms, emergency situations. On anticipating a power supply discontinuity or a peak demand in the near future, the utilities come up with Demand Response program and create an event. They will inform the customers about this event and also about how much power consumption each customer can reduce, and in case the customer participates in the event and reduces the power consumption to a certain amount, he/she gets an incentive from the utility company, say, 1$/kWh. Now, if the customer reduces the usage from 100kW to 80 kW for the period.

Utilities buy power = 80*11 = 880$

Utilities sell power to customer = 2*80 = 160$

Utilities gives incentive to customer = 1*(100-80) = 20$

So, loss incurred = [880-160+20] $ = 740$

The previous loss without the Demand Response program = (11-2) $/kW * 100kW = 900$

So, amount of reduction in loss = (900-740) = 160$ for 100kW and the percentage reduction in loss is 17.8%.

Dynamic demand (DD) or Direct load control: It is similar to demand response mechanisms to manage energy consumption in response to supply conditions. It involves the remote interruption of the customers' energy usage in which distributors cycle loads like heating, cooling, washing, elevators, etc., on and off at varying time intervals at peak hours. They do so to balance the overall system load with the generation, reducing critical power mismatches. Load control switches enable direct remote control over AC units or heating systems. Smart thermostats allow utilities to adjust temperature settings remotely. In a dynamic demand mechanism, devices are passively shut off when the stress in the grid is sensed whereas, in demand response programs, customers respond to transmitted requests to shut off the devices. These programs also hold interesting prospects for the grid to vehicle (G2V) and vehicle to grid (V2G) technologies in electric vehicles and solar rooftop with batteries.

1.2.2 Advantages of DSM programs

A meticulously planned demand-side program promises to improve the efficiency of the system in several ways. Some of them are listed below.

- DSM activities benefit the end users by giving them financial incentives in response to their load shifting, thus reducing their energy bills.
- These programs have the potential to help energy providers save money through reductions in peak demand and deferring the construction of new power plants.
- These programs help in reducing the frequency of blackouts and brownouts.
- These programs not only help in balancing the supply and demand but they also help in lowering the energy prices in the wholesale and retail markets.
- These activities help in the reduction of energy usage which in turn decreases the carbon emissions preventing the environment from climate change.

1.2.3 Challenges

Some of the roadblocks in creating an effective demand-side program are -

- For end-users participating in a demand response event can be burdensome. This is mainly because it would require curbing heating or cooling systems on days of extreme temperature. The challenge for utilities is to find appropriate incentives to keep the customers motivated.

- There is a lack of availability of data on load patterns of different consumer categories at a granular level in a digitized form. This acts as a hindrance to formulate DSM strategies.
- The need to ensure that the DSM events lead to improvement of the financial health of the utilities.
- Some DR events can be uncomfortable for some section of population like the elderly. In this case, utility companies need to have provisions to allow this group of DR users to override the event.
- Utilities use different technologies to communicate DR events with the users. Thus, coordinating DR events with IT and communication technologies require selection, testing and implementation.

Successful demand-side programs would be a win-win for utilities, governments, regulators, manufacturers, and consumers. With the numerous facets to the energy problem, there is a growing consensus that energy and sustainability problems are mainly informatics problems where machine learning techniques can play a dominant role. [18]

1.3 Datasets

In this section, all the datasets used in this thesis are described. Experiments have been conducted on 4 publicly available datasets. A brief description of each of these dataset is given below.

Table 1.1: Description of appliances used in houses in REDD dataset

Houses	Appliances
1	Electronics, Lighting, Refrigerator, Disposal, Dishwasher, Furnace, Washer Dryer, Smoke Alarms, Bathroom GFI, Kitchen Outlets, Microwave
2	Lighting, Refrigerator, Dishwasher, Washer Dryer, Bathroom GFI, Kitchen Outlets, Oven, Microwave, Electric Heat, Stove
3	Electronics, Lighting, Refrigerator, Disposal, Dishwasher, Furnace, Washer Dryer, Bathroom GFI, Kitchen Outlets, Microwave, Electric Heat, Outdoor Outlets
4	Lighting, Dishwasher, Furnace, Washer Dryer, Smoke Alarms, Bathroom GFI, Kitchen Outlets, Stove, Disposal, Air Conditioning
6	Lighting, Refrigerator, Disposal, Dishwasher, Washer Dryer, Kitchen Outlets, Microwave, Stove

- **REDD:** This dataset [19] is of moderate size. It comprises of power consumptions from six different houses. These houses are located in greater Boston area. Table 1.1 shows appliances in different houses. For each house, the total/aggregate energy consumption as well as individual consumptions of about twenty different household devices are recorded. The data is available for a period of two weeks with a very high frequency sampling rate of 15kHz. Such a high sampling rate is impractical; to emulate real-life scenario the training and testing data is aggregated over a time period of 10 minutes. A standard evaluation protocol was followed where the 5^{th} house is omitted owing to very limited availability of samples.

- **Weather:** The weather dataset used here is retrieved from the weather underground website[1]. The dataset contains 14 attributes consisting of timestamps, temperate, humidity, dew point, sea level pressure, visibility,

[1] https://www.wunderground.com/

wind direction, wind speed, precipitation, gust speed, events and conditions. Weather data was used corresponding to the energy data in REDD dataset. REDD recorded data during the months of April, May and June. The weather corresponding to these months is relatively colder (highest temperature being 13°C, 19°C and 24°C respectively), hence temperature and wind speed were selected as the exogenous input to the system of load forecasting.

- **Dataport:** The Pecan Street dataset is either obtained by registering on their website[2] or via the open source NILM toolkit (NILMTK) [20]. This dataset contains 1 minute circuit level and building level electricity data from 240 houses. The data set contains per minute readings from 18 different appliances: air conditioner, kitchen appliances, electric vehicle, and electric hot tub heater, electric water heating appliance, dish washer, spin dryer, freezer, furnace, microwave, oven, electric pool heater, refrigerator, sockets, electric stove, waste disposal unit, security alarm and washer dryer. For the problem of NILM, 165 homes were assigned for training and 72 for testing. The remaining 3 homes did not have aggregated data so they were not used in the experiments.

  For the problem of load forecasting, continuous data for more than 3 years was required. Since there were only a few houses with more than 3 years of continuous data; only those are being considered (house ids: $1589, 3310, 3369$).

  In case of anomaly detection, two months (April and May) of meter-level

[2] https://dataport.pecanstreet.org/

data from nine houses was used. The average temperature in these two months was 19°C and 23°C, respectively.

- **HUE:** The Hourly Usage of Energy (HUE) is collected from different residential houses located in Burnaby in British Columbia, Canada [21]. The residential customers of BC Hydro, a provincial power utility have donated this data. It has meter-level energy consumption values which are sampled at each hour. The data is collected over a period of three years, ranging from January 2015 to January 2018. Five houses with house ids 3, 4, 5, 6, 7 from this dataset were used. However, there are 22 houses in this dataset. Hourly temperature data was included in the dataset to detect abnormal energy consumption readings.

- **I-BLEND:** Indian Buildings Energy consumption Dataset (I-BLEND) [22] is collected for 52 months at the Indraprastha Institute of Information Technology, New Delhi, India. The dataset contains both residential (student accommodation) and academic buildings. The data is available for all the buildings at a sampling rate of once per ten minutes. For demand forecasting problem, data collected on our campus from lecture halls, academic buildings, facilities, girls hostel and boys hostel was used. It consists of meter-level data for all the buildings.

The description of each dataset is given in the Table 1.2. This table gives the name of the dataset, reference to the dataset, the application (Non-Intrusive Load Monitoring (NILM), Demand Forecasting (DF) and Anomaly Detection

Table 1.2: Description of datasets used in this thesis

Name	Reference	Applications	Section
REDD	[19]	NILM, DF	(2.3, 2.4, 2.5, 2.6), 3.2
Weather	https://www.wunderground.com/	DF, AD	(3.2 , 3.3), 4.2
Dataport	[20]	NILM, DF, AD	2.6, 3.3, 4.2
HUE	[21]	DF, AD	3.3, 4.2
I-BLEND	[22]	DF, AD	3.3, 4.3

(AD)) in which it has been used and the thesis section where this dataset has been referred.

1.4 Research Contributions

Our work aims at improving the accuracy by either modifying existing algorithms or by building new algorithms for solving various problems in the area of sustainable computation. These problems include energy disaggregation, load forecasting and detecting abnormal energy consumption patterns.

The research contributions of this thesis are summarized below:

- **Developing improved frameworks and algorithms for non-intrusive load monitoring**

  - Two main issues commonly observed in the energy disaggregation dataset were addressed. The first is the problem of missing data occurring due to the inability to transmit the readings wirelessly to the remote server. This issue was handled by modifying the mathematical framework for disaggregation to accommodate the information re-

garding missing readings. The other is the issue of data corruption (outliers) happening due to several reasons like voltage/ current surges, transients, etc. To solve this problem, mathematical formulation was modified by changing the l_2 norm on the data fidelity term to a more robust estimation using l_1 norm. This work has been discussed in detail in section 2.3.1.

  - An approach was proposed to improve the robust dictionary learning by employing the low-rank penalty (nuclear norm) instead of the l_1 norm on the learned coefficients. More details of this work can be found in section 2.4.1.
  - Unsupervised robust dictionary learning algorithm was modified by proposing two supervised models on top of it to improve the disaggregation accuracy. This work has been covered in detail in section 2.5.1.
  - A new approach for solving load disaggregation problem was introduced. This is done using analysis version of dictionary learning approach - transform learning. In practical scenarios of low training data regime, this method always excels over the state-of-the-art techniques. An elaborate discussion on this work can be found in section 2.6.1.

- **Developing algorithms to improve the accuracy of building level demand forecasting**

  - An approach to solve the problem of short-term load forecasting using Kalman filtering algorithms was proposed. Owing to the problem's non-linearity and non-stationarity, nonlinear variants of the Kalman

filter, that is extended Kalman filters (EKF) and unscented Kalman filters (UKF) were used. In this work, historical energy data, temperature and wind speed were used. The full details of this work can be found in section 3.2.

  - An approach to improve upon a sparse coding based forecasting approach was proposed. It used a deeper version of dictionary learning for point and profile load forecasting. An improvement upon the prior work was proposed in two ways. First, a deeper non-linear extension of shallow dictionary learning was invoked. Second, regression was incorporated into the deep dictionary learning process. This work has been discussed in detail in section 3.3.1.

- **Developing algorithms to generate ground truth, evaluate performance accuracy and improve the accuracy of anomaly detection in buildings**

  - The lack of ground truth has hampered the development of advanced algorithms as there is no clear way of testing their performance accuracy. To address the challenges faced by testing anomaly detection algorithms our work provides:

    1. two novel methods to generate labeled (i.e., ground truth) data for abnormal energy consumption in buildings for both short-range and long-range datasets;
    2. the source code used to generate labeled data in a standard way;
    3. a publicly available dataset of anomalies found in our experiments,

so researchers can use this data directly;

4. a comprehensive review of all the different accuracy measures used; and
5. a framework and discussion on how accuracy methods work when compared to each other and what performance metrics to use.

An elaborate discussion on this work is given in section 4.2.2.

– A robust principal component analysis (RPCA) technique was employed to separate abnormal energy consumption patterns from normal energy usage. An improvement in accuracy over existing methods in cases of real and injected anomalies was observed. Section 4.3.2 has covered this work in detail.

1.5 Publications

The work done in this thesis has resulted in several publications. This section lists the publication categorized by type of publication venue.

Journals

1. **Gaur M.**, Majumdar A., “Regressing deep dictionary learning for building level short term load forecasting", IEEE Transactions on Smart Grid, 2019 (Submitted) [3.3]
2. **Gaur M.**, Makonin S., Bajic I., Majumdar A., “Performance evaluation of techniques for identifying abnormal energy consumption in buildings",

IEEE Access, 2019 [4.2]

3. **Gaur M.**, Majumdar A., “Disaggregating Transform Learning for Non-intrusive Load Monitoring", IEEE Access, 2018 [2.6]

International Conferences

4. **Gaur M.**, Majumdar A., “Robust Supervised Sparse Coding for Non-Intrusive Load Monitoring", IJCNN, Rio, Brazil, 2018 [2.5]

5. **Gupta M.**, Majumdar A., “Nuclear Norm Regularized Robust Dictionary Learning for Energy Disaggregation", EUSIPCO, Budapest, 2016 [2.4]

6. **Gaur M.**, Majumdar A., “Short-Term Load Forecasting using non-linear Kalman Filters", Elsevier Energy Procedia, Hyderabad, 2016 [3.2]

7. **Gupta M.**, Majumdar A., “Handling Missing data and Outliers in Energy Disaggregation", Elsevier Energy Procedia, Hyderabad, 2016 [2.3]

Letters

8. **Gaur M.**, Majumdar A., “Robust PCA for Anomaly detection", IEEE Communication Letters, 2019 (Submitted) [4.3]

1.6 Outline of Thesis

The structure of this thesis is as follows -

In Chapter 1, demand-side management for sustainable energy is introduced. A discussion on various activities adopted by power utilities to formulate effec-

tive demand-side strategies is given. This is followed by a discussion on how these activities work to modify consumer's demand for energy. Finally, our contributions in this area are presented.

In Chapter 2, the problem of non-intrusive load monitoring or load disaggregation is introduced. This chapter is divided into four sections, each of which focuses on improving the disaggregation accuracy of the appliances. In section 2.3, a review of existing works in the area of NILM is presented. The challenges that energy data poses to the research community are discussed. A description of methods proposed to address the challenges is given which is followed by the empirical evaluation. The proposed methods are shown to improve the disaggregation accuracy when compared with the benchmarks. In section 2.4, the work on learning dictionaries in robust fashion is extended to deal with the problem of data corruption by large and sparse outliers. All prior works are based on the assumption that energy consumption follows a linear mixing model, that is the total energy consumed is the linear sum of the energy consumed by individual appliances. They assume the modeling error to be small and normally distributed. The problem with this assumption is explained and an approach that combines learning robust dictionaries along with learning rank deficient coefficients is proposed. This is followed by the empirical evaluation showing our approach outperforming the state-of-the-art. In the next section 2.5, dictionaries and sparse codes are learnt in a way that they look as dissimilar as possible for each appliance. To ensure this, two supervised models on top of the unsupervised robust dictionary learning are proposed to improve the disaggregation

accuracy. The proposed models are shown to outperform the existing works. In the last section 2.6 of this chapter, a new method for solving energy disaggregation problem based on transform learning is proposed. For supervised NILM approaches, data acquisition at the training phase is expensive owing to the cost of buying/ renting plug-level sensors. A brief literature review on the advantages of transform learning over synthesis dictionary learning is given. The proposed method is followed by the empirical analysis on how it outperforms the benchmark techniques in limited training data regime, thus reducing the cost of NILM.

In Chapter 3, the problem of short-term load forecasting is addressed. In section 3.2, the focus is on using nonlinear Kalman filtering algorithms to perform a day-ahead demand forecasting in residential buildings. The mathematical formulation of two nonlinear variants of Kalman filters is presented followed by the empirical evaluation; proposed method is shown to outperform the existing works. In section 3.3, a deeper extension and generalization of the sparse coding based forecasting approach is proposed for point and profile load forecasting. It is based on the synthesis and analysis versions of deep dictionary learning regression. The literature review of the same is presented followed by the proposed works. It is shown that proposed work outperforms the existing techniques.

In Chapter 4, the emphasis is on identifying abnormal energy consumption patterns in buildings. In section 4.2, two problems commonly faced in the anomaly detection research community are discussed. First problem is the lack

of availability of ground truth to test the algorithms and the second is the lack of a unified performance accuracy metric. The literature review of different techniques used to detect anomalies in building energy is presented. This is followed by the proposed methodologies. Given the user-defined threshold, two approaches to generate the ground truth based on the range of the available dataset are proposed. A detailed analysis of a list of performance metrics used in the literature is presented. An evaluation of the existing works against the ground truth generated by the proposed method is performed using different metrics. Each metric is examined for the problem at hand. In section 4.3, an approach for unsupervised anomaly detection using robust principal component analysis (RPCA) is presented. Anomalies injected in the dataset are assumed to be positive (power outage) and negative (power theft). The results obtained from the proposed method outperform the existing techniques.

In Chapter 5, the contributions of this thesis are summarized and the future direction of this work is highlighted.

Chapter 2

Non-Intrusive Load monitoring

2.1 Introduction

NILM [12] is the task of separating the whole energy signal of residential, commercial or industrial buildings into energy signals of individual appliances. It is called non-intrusive to contrast it with previous techniques that required installing sensors on individual appliances to collect appliance load data. Currently, residential and commercial buildings account for 40% of the total energy consumption [11]. Studies have estimated that 20% of this consumption could be avoided with changes in user behavior [23]. Energy disaggregation is the task of segregating the combined energy signal of a building into the energy consumption of individual appliances. The information regarding consumption pattern is fed back to consumers with the goal of increasing their awareness about energy usage and its wastage. Studies have shown that such precise and detailed feedback to consumers can be quite effective in improving energy conservation [24].

The approach towards energy disaggregation is broadly based on the nature of the targeted household and commercial appliances. These appliances can be broadly categorized as simple two-state (on/off) appliances such as electrical toasters and lights; more complex multistate appliances like refrigerators and washing machines; and continuously varying appliances such as IT loads (printers, modems, laptops, etc.). The earliest techniques were based on using real and reactive power measured by residential smart meters. The appliances' power consumption patterns were modeled as finite state machines [12]. These techniques were successful in disaggregating simple two state and multistate appliances, but they performed poorly in the case of time-varying appliances which do not show a marked step increase in the power. The techniques based on stochastic finite state machines used Hidden Markov Models and their variants [25, 18, 26], have improved upon the prior approach. More recent approaches have focussed on unsupervised energy disaggregation, i.e. without sub-metering or additional hardware requirements [27, 28]. NILM not only provides information about activities within the home but it has also been used for healthcare applications to assist independent living for elderly. The application of NILM specific to the areas of Home Energy Management System (HEMS) and Ambient Assisted Living (AAL) in smart homes have been reviewed and discussed in [29, 30]. A review of different ILM and NILM approaches have been covered in [31]. Dictionary learning and sparse coding [32, 33] based approaches have also been extensively used in energy disaggregation. Such dictionary learning based methods are not limited by the assumptions of stochas-

tic finite state machines and hence are capable of handling all kinds of loads - multi-state and continuously varying. There is yet another class of methods that is gaining popularity in recent times based on the multi-label classification approach [34, 35, 36]. These do not model the electrical appliance in any way but want to predict the state of the appliance (ON / OFF) given the aggregate power signal. Since multiple appliances can be ON at the same time, this turns out to be a multi-class classification problem. However these techniques cannot estimate the consumption accurately.

2.2 Literature Review

The literature on energy disaggregation can be divided into two categories. The first category is event based classification and the other category directly addresses the disaggregation problem.

1. **Event based techniques** The first group of algorithms focuses on classifying electrical events present in the signal. The earliest techniques in this group were based on using real and reactive power measured by the residential smart meters. The appliances' power consumption patterns were modelled as finite state machines [12] and sharp edges were detected in real and reactive power signals. These techniques were successful in segregating two state (on/off) appliances and multistate appliances but performed poorly in the case of time-varying appliances which do not show a marked step increase in the power. Besides, such techniques required high reso-

lution data (at least once a second) but smart-meters sample once every 10-15 minutes. With such low frequency data, the sharp edges required by the HMM to learn the model are smoothed out and hence the aforesaid techniques do not yield desirable results.

Later, the devices were clustered based on their consumption changes but the drawback was that the clusters associated with several low power different devices became indistinguishable. In order to improve the distinguishability of the devices, later research used transient and harmonic information using very high sampling [37]. However, high frequency sampling requires costly hardware and installation of monitoring devices in the building.

2. **Non-event based techniques** The other category directly addresses the disaggregation problem by decomposing the total energy signal into its component appliances over time [38, 18, 32, 39]. This group can further be divided into supervised and unsupervised learning methods.

    - **Supervised Algorithms** This class of algorithms require labelled data sets to train the classifier so that appliances can be recognized from the aggregated load measurement. They can be broadly divided into optimization or pattern recognition based methods.

        – Optimization Methods: These methods pose the disaggregation problem as an optimization problem. The features extracted from an unknown load are matched to the known load present in the pool of appliance database and the closest possible match is found

to minimize the error between them. However, this problem becomes complex in case of composite load disaggregation where combination of appliances are to be matched instead of one-to-one matching. Several optimization techniques [40, 41, 42] like integer programming, genetic algorithms have been tried to tackle this problem. The challenges involved with these methods are to reduce the complexity in case any unknown load and appliances with overlapping load signatures are found in the aggregated load data.

  - Pattern Recognition Methods: These are the most frequently used approaches for load disaggregation. In [43], Bayesian approach is used to detect most likely states of the appliances. A naive bayes classifier is trained for each appliance and accordingly a set of trained classifier is used to recognize the appliance specific states from the aggregated load measurements. This study made an assumption that the states of appliances are independent of each other which seems to be false as in a residential environment, operation of consumer appliances are correlated to each other. In another research [44, 45], temporal information along with real power values are used to facilitate disaggregation task. This also encouraged the researchers to use Artificial Neural Network (ANN) [46] and Hidden Markov Models (HMM) [47] due to their ability to learn temporal and appliance state transition information.

- **Unsupervised Algorithms** One of the main challenges of NILM is

the lack of *a-priori* training information. It is highly required for the NILM system to be installed with minimal cost setup as the training requirement for the supervised methods are expensive and laborious. Therefore, unsupervised learning methods [48] are needed for wider application of NILM techniques. The genetic K-means and agglomerative clustering approaches have been investigated to automatically determine the total number of appliance clusters from the aggregated load data. Each cluster is considered to be a combination of multiple appliances which are further reduced to individual sources. Matching Pursuit (MP) algorithm is used for source reconstruction. However, this approach also has several drawbacks like presence of smaller appliances having similar consumption level, presence of multistate appliances that form several clusters resulting in mixing of events. Some of the recent unsupervised approaches [49] have used graph based signal processing (GSP) for load disaggregation. These methods have shown to perform equally well when compared to supervised approaches that employ GSP for data classification only [50].

Another work [51], uses motif mining approach for unsupervised disaggregation task. It uses power change events instead of power consumption. [38] uses probabilistic models of appliance behavior using variants of FHMM. The non-power features like time of usage, duration of usage along with the real power consumption is used to model the device specific HMMs. FHMMs are well suited to model appliances con-

tributing independently to the aggregated load data. Other models like Conditional Factorial Hidden Semi-Markov Model (CFHSMM) gave the best unsupervised disaggregation performance achieving an accuracy of 83% in comparison to FHMMs. The analysis showed that the on-state occupancy distribution can be best represented using gamma distribution and the inclusion of non-power features along with the additional information like correlation between usage of appliances has shown to improve the performance. At the same time, the drawback of FHMMs is that the existing techniques for hidden state estimation are susceptible to local optima. To address this, [18] proposed a new inference algorithm with convex formulation, Additive Factorial Approximate MAP (AFMAP). They have also used frequently occurring appliance patterns [51] as the load signatures. A more detailed and comprehensive survey of the algorithms can be found here [52].

Recently, sparse coding and dictionary learning based approaches like [32, 33, 53] have been used to address the said problem. These techniques do not suffer from the same pitfalls as HMM, i.e. they can work with low frequency data on time varying appliances. There are many other approaches to address the same problem ranging from pure rule based methods to completely data driven techniques (neural networks, multi-label classification etc.) [52].

Supervised energy disaggregation works in two phases, training phase and disaggregation phase. In the training phase, the power consumption data for

each appliance is collected separately over time, and the model for the appliance is built. In the disaggregation phase, the composite data from multiple appliances is available and the task is to estimate the power consumed by each appliance. Given the success of such dictionary learning based techniques, the proposed work is formulated on the same approach.

Dictionary learning based methods learn a codebook that can represent the sub-metered active power measurements (X_i) for appliance i. This is expressed as:

$$X_i = D_i Z_i + \epsilon, i = 1 \dots N \tag{2.1}$$

Here N is the total number of appliances, D_i and Z_i are codebook/ dictionary and loading coefficient of i^{th} appliance. Both D and Z need to be learnt.

All prior studies assumed that the modelling error (ϵ) is small and Normally distributed. Therefore, a Euclidean norm based minimization technique was employed. In a generic fashion, the learning can be expressed as:

$$\min_{D,Z} \|X_i - D_i Z_i\|_F^2 + R(D_i, Z_i), \tag{2.2}$$

where R is some penalty on D and Z. The main cost function is Euclidean hinged on the assumption $\epsilon \sim N(0, \sigma^2)$. However, the assumption that the modelling error is Normally distributed is incorrect. This can be seen via the plots of probability density functions in fig. 4.1 that best fit the histograms of residential energy consumption from 4 different houses. Sub-metered or training data

always show spikes which are not typical to the appliance under study. They arise out of power surges or from transients. In such a case the modelling error does not follow a Normal distribution. It is large in magnitude but sparse. In this work, a more realistic noise model is addressed.

During disaggregation, all prior studies assume that the total power logged by the smart-meter is a sum of the power consumed by individual appliances that are turned on. This is expressed as,

$$X = \sum_i X_i + \epsilon, \tag{2.3}$$

Here too, the assumption is that the modelling error is small and approximately follows a Normal distribution. There are two reasons why this assumption is wrong. The first one has already been discussed. There are unforeseen spikes owing to uncontrollable reasons. The other reason is owing to nonlinear effects. The assumption that total power is a sum of individual power consumed by different appliances only holds for passive loads. The non-linearity can arise in two ways-

1. Today, most of our appliances such as refrigerators, AC's, washers, microwaves, laptops, printers etc. are quite sophisticated and cannot be modeled as passive loads. They have internal sources of electromagnetic emission, e.g. the switched mode power supplies (SMPS) in a desktop or a laptop adapter. These secondary sources of emission can interfere with the

loads on the power lines depending on the proximity of the loads as well as their frequency response. In such cases, where the appliance needs to be modeled as a combination of a source and a load, the linear mixing model does not hold [54].

2. Secondly, the reactive components of the loads (transformers, magnetic and capacitive elements within the power supplies and AC to DC converters) exhibit nonlinear behavior depending on the frequency of operation.

In a recent work [1], a similar approach is followed - it assumed that the linear mixing model for energy holds in most cases (since this model is known to yield good results for simple loads), but that there are a few large perturbations arising out of the inherent non-linearity inside the load. The reasoning here is that linearity holds at the power line frequency (50/60Hz) while the non-linearity arises from the electromagnetic emission within the appliances at higher frequencies. In order to meet federal regulations on emissions, some of these appliances are fitted with good quality filters that restrict the emission. Secondly, these emissions are likely to decay with the length of the transmission cables. Therefore, a perturbation based model of the load non-linearities may be suitable in this context.

In short, it can be argued that both during training and during actual disaggregation, the assumption that the modelling error is small (Normally distributed) is not true. In both cases the actual noise is sparse but of large magnitude. During training, the sparse noise arises out of power surges; during disaggregation

it arises from power surges and other nonlinear effects. Since prior studies assumed the noise to be small, they employed an l_2-norm cost function. It is well known that the Euclidean norm is not robust to outliers (large and sparse perturbations). Therefore, the proposed work employs an l_1-norm cost function for both training and disaggregation. This makes the cost function robust to outliers.

A typical dictionary learning problem with sparse coefficients is shown in (2.4). In [32] various penalties were proposed on the dictionaries and the coefficients. In the simplest formulation the problem is solved via:

$$\min_{D,Z} \|X_i - D_i Z_i\|_F^2 + \lambda \|Z_i\|_1, \tag{2.4}$$

This is bi-linear problem; it is usually solved via alternating minimization, i.e. the coefficients are estimated assuming the dictionary is fixed; and the dictionary/ codebook is updated assuming that the coefficients are fixed. Off-the-shelf algorithms exist for each of the two problems. Usually, the atoms of the dictionary are normalized to prevent degenerate solutions.

Learning the dictionary constitutes the training phase. During actual operation, several appliances are likely to be in use simultaneously. They [32] make the assumption that the aggregate reading by the smart-meter is a sum of the powers for individual appliances. Thus if X is the total power from N appliances (where the columns indicate smart-meter readings over the same period of time as in training) the aggregate power is modeled by (2.3). By imputing

(2.1) in (2.3), one can express (2.3) as -

$$X = \begin{bmatrix} D_1 \dots D_n \end{bmatrix} \begin{bmatrix} Z_1 \\ \vdots \\ Z_n \end{bmatrix}. \tag{2.5}$$

The loading coefficients can be solved using l_1-norm minimization. Once the loading coefficients are estimated, the consumption for each appliance is obtained by:

$$\hat{X}_i = D_i Z_i, i = 1 \dots N \tag{2.6}$$

Prior studies in dictionary learning are based on minimizing an l_2-norm data mismatch - the underlying assumption being that the noise (ϵ) is approximately Normally distributed. This does not hold for this problem; the reasons have been explained above. The noise is large but sparse. A more appropriate model would be one where ϵ is sparse (modelling nonlinear perturbations). If there are such large outliers the estimate from Euclidean norm minimization is skewed towards the outlier. For a robust estimate [1] proposed to replace the l_2-norm by an l_1-norm data mismatch (since ϵ is sparse),

$$\min_{D,Z} \|X_i - D_i Z_i\|_1 + \lambda \|Z_i\|_1, \tag{2.7}$$

Both [32, 1] imposed sparsity constraint on the coefficients. However, there

is no clear motivation behind the requirement of enforcing sparsity on the loading coefficients.

Once the dictionary is learnt, [1] follows the procedure similar to [32]. The aggregate consumption (X) is assumed to be the sum of consumptions from individual appliances (X_i's). However, [1] acknowledges that the modelling error ϵ is large but sparse. Thus they estimated loading coefficients (Z_i's) by solving,

$$\min_{Z} \|X - \sum_{i} D_i Z_i\|_1 + \lambda \sum_{i} \|Z_i\|_1, \tag{2.8}$$

The problem with minimizing the l_1-norm is computational. However, over the years various techniques have been developed. The earliest known method is based on Simplex [55]; Iterative Reweighted Least Squares [56] used to be another simple yet approximate technique. Other approaches include descent based method introduced by [57] and Maximum Likelihood approach [58].

In [1] a more modern approach is followed for solving the l_1-norm cost function. It is based on variable splitting and augmented Lagrangian. It decomposes the difficult problem of l_1-norm data mismatch into several easier sub-problems whose solutions exist.

2.3 Handing Imperfection in Energy Disaggregation

The proposed work addresses two commonly overlooked imperfections in energy data. The first one is the problem of missing readings and the second one is regarding large yet sparse outliers.

In a typical experimental set-up, the data (training and testing) is collected by attaching smart-meters to individual appliances, which acquire the reading at periodic intervals and transmit these to a remote server wirelessly. Many a time the home area network (HAN) interface such as WiFi, Zigbee, Bluetooth, etc or the wide area network (WAN) interface do not work thus leading to loss of data packets. These HAN and WAN interfaces are important for enabling the communication between the smart meters itself with the in-home consumer devices and energy utility center respectively. The malfunctioning of interfaces leads to missing data; it is a common phenomenon in almost all energy disaggregation datasets. The feasibility of NILM is when the aggregate readings from the smart meter are needed for the purpose of training and testing and submetering is performed purely for validation of results. In cases like these, the packet losses would only affect the validation of results. This is the first issue that is addressed in this work.

The second issue is regarding outliers. Energy datasets are often corrupted by large but sparse noise (outlier). This may arise in several ways. It can occur from transients, voltage/ current surges. In all such cases, the anomaly is of large magnitude but for a very short duration. Such spikes are uncontrollable

and do not show any characteristics of the appliance under study.

The issues of data fidelity were first discussed in [59]; the author (of [59]) also released a code preprocessing such datasets. This work is different from the approach presented in [59]; the proposed work does not try to 'cure' the data by preprocessing, rather model them in the mathematical framework thereby minimizing the error caused during preprocessing.

The first problem of missing data is usually handled in a heuristic fashion by the NILM research community. The missing values are either imputed by prior readings or at best by nearest neighbor interpolation [60, 61, 62]. In this work, missing readings are not interpolated, rather accommodated into the mathematical framework for disaggregation.

The second problem, i.e. the problem of spiky outliers have been largely ignored by NILM researchers until recently [1, 63]. Various signal processing techniques that have been proposed for NILM to remove spikes, noise and to smooth the signal [28]. Some of them are smoothing filters including median filters, mean filters, kernel weighted average filters and their possible combinations [26, 64]. Total variation regularization has been used prior to the additive Factorial HMM based NILM to remove outliers and to minimize the influence of rarely used appliances [18]. [63] used a graph based signal processing approach to improve low-rate supervised and unsupervised event-based NILM classification. Prior studies assumed that the noise in the system is small (Normally distributed) and hence employed a Euclidean data fidelity term. A recent

study [1] argued about the existence of sparse but large outliers and following studies in robust estimation proposed a robust data fidelity term based on absolute deviations. A similar approach is followed in the proposed work.

2.3.1 Proposed Approach

2.3.1.1 Handling Missing Readings

The problem of solving an under-determined linear inverse problem where the solution is known to be sparse is studied by Compressed Sensing (CS).

$$y_{m\times 1} = A_{m\times n}x_{n\times 1} + \eta_{m\times 1}; m < n \tag{2.9}$$

(2.9) is an under-determined system of linear equations as the number of constraints (m) in A is less than the number of unknowns (n), that is $m < n$. Here, y is a m x 1 dependent variable, A is the m x n matrix with coefficient values, x is the n x 1 unknown vector and η is the vector of intercepts. In general there are infinitely many solutions to (2.9). CS is interested in the case where the solution is s-sparse, i.e. x has only s non-zero values, the rest n-s being zeroes. Donoho's seminal work [65] showed that a sparse solution is in most cases unique. CS literature shows that such a sparse solution can be recovered by l_1-minimization [65].

$$\min_x \|y - Ax\|_2^2 + \lambda\|x\|_1 \tag{2.10}$$

where λ is the regularization parameter. For most practical problems the signal is not sparse in itself but has an approximately sparse representation in the transform domain. Orthogonal and tight-frame transforms are useful in this case since the synthesis (2.11) and analysis (2.12) hold.

$$synthesis : x = W\alpha \tag{2.11}$$

$$analysis : \alpha = W^T x \tag{2.12}$$

where W is the sparsifying transform (wavelet / DCT for image, wavelet for EEG / ECG, STFT for speech etc.) and α the sparse transform coefficients. This allows inverse problems arising from sparsifiable natural signals to be expressed in the following form,

$$y = AW\alpha + \eta \tag{2.13}$$

The sparse transform coefficients are solved using l_1-minimization (2.10) from which the signal of interest is obtained by applying the synthesis equation (2.11). In CS lingo, A is called the measurement operator and W is the sparsifying transform.

Dictionary learning gained popularity with the advent of K-SVD [66]. It was showed that instead of using fixed basis (like wavelet, DCT etc.) better solutions can be achieved by learning the sparsity basis from data. For dictionary learning, one needs training data (say X) from which a dictionary (D) is learnt such that the coefficients (Z) are sparse. In the training phase, X is the matrix with

column i representing submetered active power measurements from appliance i; D and Z has learned atoms/ basis and coefficients for appliance i in column i. The synthesis dictionary learning problem is framed as,

$$X = DZ \tag{2.14}$$

The learning can be framed in various ways. The basic constraints Z should be sparse and there should not be degenerate solutions, i.e. very large D and small Z or vice versa. This can be prevented in several ways. K-SVD proposes an elegant (albeit slow) solution based on rank-1 updates. Others propose a normalization constraint on the columns of the dictionary. But the easiest formulation is to have a simple Frobenius norm penalty on the dictionary as a whole. This leads to the following formulation-

$$\min_{D,Z} \|X - DZ\|_F^2 + \lambda_1 \|D\|_F^2 + \lambda_1 \|Z\|_1 \tag{2.15}$$

This (2.15) constitutes the training phase. The learnt dictionary is used as the sparsifying basis in the testing phase for solving the sparse inverse problem.

Blind Compressed Sensing (BCS) [67] marries CS with dictionary learning. Instead of learning the dictionary in an offline fashion and then using it for solving the inverse problem, it learns the dictionary on the go. Obviously, it can be only used for multiple measurement vector (MMV) problems; this is because many samples would be needed to robustly estimate the dictionary.

Say the problem is to solve $Y = AX$; according to the dictionary learning

formulation one assumes $X = DZ$; incorporating one into the other leads to -

$$Y = AX = ADZ \tag{2.16}$$

The solution to equation (2.16) is formulated as follows,

$$\min_{D,Z} \|X - ADZ\|_F^2 + \lambda_1 \|D\|_F^2 + \lambda_1 \|Z\|_1 \tag{2.17}$$

Kolter et. al. [32], assumed that there is training data collected over time, where the smart-meter logs consumption from every appliance individually. This can be expressed as X_i where i is the index for an appliance, the columns of X_i are the readings over a period of time. For each appliance, a codebook is learnt; this assumption is expressed in (2.18).

$$X_i = D_i Z_i, \qquad i = 1 \dots N \tag{2.18}$$

where D_i represents the codebook/dictionary and Z_i are the coefficients, assumed to be sparse. This is a typical dictionary learning problem with sparse coefficients.

In this work, the missing readings are not interpolated rather they are modeled into the mathematical framework.

$$Y_i = R_i \bigodot X_i \tag{2.19}$$

where R_i is the binary sampling mask; it is 1 when the reading has been obtained and 0 when the reading is missing, $\bigodot$ indicates element-wise product and Y_i is

the data that is actually obtained (at the server). The estimation problem (for dictionary and coefficients) is expressed as,

$$\min_{D_i, Z_i} \|Y_i - R_i \bigodot D_i Z_i\|_F^2 + \lambda_1 \|D_i\|_F^2 + \lambda_2 \|Z_1\|_1 \tag{2.20}$$

This is akin to the BCS formulation. The algorithm for solving this problem is available at [68]. The missing data problem exists also in the test/ disaggregation phase. The aggregate reading is expressed as the sum of the power readings from individual appliances.

$$X = \sum_i X_i = \begin{bmatrix} D_1 \dots D_n \end{bmatrix} \begin{bmatrix} Z_1 \\ \dots \\ Z_n \end{bmatrix} \tag{2.21}$$

The dictionaries are obtained from training, the task at disaggregation is to estimate the coefficients Z_i's; this is done by simple l_1-minimization.

$$\min_Z \|X - DZ\|_F^2 + \lambda_2 \|Z\|_1 \tag{2.22}$$

where $D = \begin{bmatrix} D_1 \dots D_n \end{bmatrix}$ and $Z = \begin{bmatrix} Z_1 \\ \dots \\ Z_n \end{bmatrix}$

Once the loading coefficients are estimated, the consumption for each appliance is obtained by:

$$\hat{X}_i = D_i Z_i, \qquad i = 1 \dots N \tag{2.23}$$

The issue of missing reading arises during disaggregation as well. A more appropriate model (compared to prior studies) would be to express the problem as $Y = R \bigodot X$. Here, X is the aggregate active power readings which contains missing values and Y is the aggregate active power readings obtained at the server. Thus the recovery is posed as a simple CS problem.

$$\min_{Z} \|Y - R \bigodot DZ\|_F^2 + \lambda_2 \|Z\|_1 \tag{2.24}$$

2.3.1.2 *Handling Outliers*

In the previous subsection, it was assumed that the noise in the system, if any, is small. Hence the l_2-norm data fidelity term is justifiable. However, as discussed in the introduction, this is not the case. Training data is corrupted by sparse but large outliers arising from electrical surges and transients. The test data is corrupted not only by such surges and transients but also effects arising out of nonlinear mixing. In such cases, where the noise appears as large and sparse outliers, the l_2-norm is not optimal. There is a large body of literature in robust statistics that argues against the usage of l_2-norm minimization; it works when the deviations are small - approximately Normally distributed; but fail when there are large outliers (as in this case). The Huber function [69] has been in use for more than half a century in this respect. The Huber function is an approximation of the more recent absolute distance based measures (l_1-norm). Recent studies in robust estimation prefer minimizing the l_1-norm instead of the Huber function [69, 70, 71]. The l_1-norm does not bloat the distance between

the estimate and the outliers and hence is robust.

Following such studies in robust estimation, l_1-norm data fidelity term is employed in place of the l_2-norm. For the scenario where the data is assumed to be complete (by simple interpolation) has been addressed in [59]. In this work we look at a more challenging problem - problem of outliers and missing readings. Therefore the corresponding formulation for the training using sub-metering active power data (modifying equation (2.20)) and test/disaggregation using aggregate active power data (modifying equation (2.24)) phases are:

$$\min_{D_i Z_i} \|Y_i - R_i \bigodot D_i Z_i\|_1 + \lambda_1 \|D_i\|_F^2 + \lambda_2 \|Z\|_1 \tag{2.25}$$

$$\min_{Z} \|Y - R \bigodot DZ\|_1 + \lambda_2 \|Z\|_1 \tag{2.26}$$

The solution for (2.26) is a standard one. Here, YALL1 algorithm [72] is used to solve the optimization problem. However, solving (2.25) is not so straightforward; no off-the-shelf algorithm exists for (2.25). An algorithm is derived to solve (2.25) in the following sub-section.

2.3.1.3 *Deriving a solution for (2.25)*

Dropping the subscript 'i' from (2.25) (for the sake of simplicity) the task is to solve:

$$\min_{D,Z} \|Y - R \bigodot DZ\|_1 + \lambda_1 \|D\|_F^2 + \lambda_2 \|Z\|_1 \tag{2.27}$$

An elegant way to solve this problem is via Split Bregman technique. We substitute $P = Y - R \bigodot DZ$, and introduce the Bregman relaxation variable (B) leading to,

$$\min_{P,D,Z} \|P\|_1 + +\lambda_1 \|D\|_F^2 + \lambda_2 \|Z\|_1 + \mu \|P - (Y - R \bigodot DZ) - B\|_F^2 \tag{2.28}$$

This can be recast into the alternating minimization of the following subproblems:

$$P1 : \min_{P} \|P\|_1 + \mu \|P - (Y - R \bigodot DZ) - B\|_F^2 \tag{2.29}$$

$$P2 : \min_{D} \lambda_1 \|D\|_2^F + \mu \|P - (Y - R \bigodot DZ) - B\|_F^2 \tag{2.30}$$

$$P3 : \min_{Z} \lambda_2 \|Z\|_1 + \mu \|P - (Y - R \bigodot DZ) - B\|_F^2 \tag{2.31}$$

P2 is the easiest to solve; it is a least squared problem having a closed form solution. P1 has a closed form solution via soft thresholding. P3 needs to be solved iteratively via iterative soft thresholding. The usual constraint about positivity is enforced on the coefficient Z after every update.

The final step is to update the Bregman relaxation variable,

$B \leftarrow P - (Y - R \bigodot DZ) - B$

There are two stopping criteria for the Split Bregman algorithm. Iterations continue till the objective function converges (to a local minima) or if the maximum number of iterations reach 200.

2.3.2 Experimental Results

The proposed work is evaluated on REDD dataset [19]. The details of this dataset are given in section 1.3. The evaluation metric used in this work is same as that defined in [19], also known as 'disaggregation accuracy' and formally defined in eq(2.32).

$$Acc = 1 - \frac{\sum_t \sum_i |\hat{y_t}^i - y_t^i|}{2 \sum_t \bar{y}_t} \tag{2.32}$$

where $\hat{y_t}^i$ denotes the algorithm's prediction for the *i*th device at the *t*th time step; the 2 factor in the denominator is to discount the fact that the absolute value will "double count" errors.

In [1] it was shown that robust dictionary learning (RDL) with sparse coefficients yields better results than the standard simple sparse coding (SC) proposed in [32] and the Factorial Hidden Model (FHMM) [18]. Therefore the proposed approach is compared only with [1] and powerlet based energy disaggregation (PED) technique [33].

Same testing procedure is followed as outlined in [19]. There are two modes of testing, mode I (training mode) and mode II (testing mode). In the training mode, a portion of the data from every household is used as training samples and rest (from those households) is used for prediction. In the testing mode, the data from four households are used for training and the held-out one is used for prediction; this is a more challenging problem.

Table 2.1: Results after improving missing data and outliers problem(in %). Comparison of proposed approach has been done with RDL and PED methods.

Houses	Mode I			Mode II		
	RDL [1]	PED [33]	Prop	RDL [1]	PED [33]	Prop
1	75.5	81.6	77.0	53.0	46.0	54.5
2	66.7	79.0	69.1	56.3	49.2	58.0
3	65.2	61.8	67.0	43.9	31.7	45.7
4	63.7	58.5	65.9	60.1	50.9	61.6
6	68.5	79.1	70.2	60.2	54.5	62.0

The proposed algorithm requires specifying two parameters (λ_1, λ_2) and one hyperparameter (μ). Some recent studies have shown that in a Split Bregman based technique, one can put the parameters to be unity and only tune the μ. We use the simple L-curve method to find out the hyperparameter; the value we obtained is $\mu = 0.01$.

The results in table 2.1 show the disaggregation accuracies using different baseline algorithms. On comparing with [1], we get an insight regarding the importance of modelling missing readings. We show that instead of imputing the readings in a naive fashion, better disaggregation results can be achieved if the missing values are modelled into the formulation.

We see that the in training mode (mode I), powerlet based method [33] outperforms the simple yet robust learning techniques ([1] and proposed). In this case, the training data is scant but the test conditions are simple. The results show a significant drop from the mode I to mode II as in the latter, prediction is done on the previously unseen set of devices. The powerlet based method shows poor performance when the training data volume increases but at the same time the problem becomes more challenging.

2.3.3 Summary

In this work, two often ignored problems that are ever-present in non-intrusive load monitoring are addressed. The first problem is of missing readings in the dataset - arising due to malfunctioning of the HAN and WAN interfaces. The second problem is that of large and sparse outliers occurring out of transients, surges and non-linearities in the load. The second problem has been handled in a recent work [1]. In this work, the missing data problem is combined with the outlier removal problem in a single combined framework. It was shown in the prior study [1] that better results are indeed obtained when outliers are removed.

A simplest possible formulation for dictionary learning is used. In [32] it was shown that better disaggregation results can be achieved when disaggregating terms are appended to the formulation in the learning phase.

2.4 Proposed nuclear norm regularised Robust Dictionary Learning for NILM

2.4.1 Proposed Approach

In prior studies [32, 1] the loading coefficients were assumed to be sparse. However, sparsity does not model any reasonable aspect of the disaggregation problem; it only learns a dictionary to express the smart-meter signals in a sparse fashion.

Let us take a closer look at the problem; especially the training phase. The x_i's basically tell us the power consumption of individual devices across time. Ideally they are non-zero only when they are ON and zero when OFF. Consider a utopian situation where the devices are turned on exactly at the same time every day; in that case the matrix formed by stacking the x_i's will be of rank-1; since all the columns x_i's will be the same. In general this assumption will never hold in practice, each of the x_i's will be time shifted versions of each other. Here we propose to learn a dictionary that approximately aligns the input signals (x_i's) so that the resultant output (coefficients z_i's) are approximately aligned. If the z_i's are approximately aligned the coefficients matrix Z_i (formed by stacking z_i's as columns) will be of low-rank. Following this assumption, we learn the dictionaries such that the resultant coefficients will be of low-rank. This is formally expressed as,

$$\min_{D_i, Z_i} \|X_i - D_i Z_i\|_1 + \lambda \|Z_i\|_{NN}, \tag{2.33}$$

The nuclear norm is the convex surrogate of the rank of a matrix; it enforces rank deficiency on the variable.

For disaggregation, we follow the standard superposition model -

$$X = \sum_i X_i + \epsilon = \sum_i D_i Z_i + \epsilon, \tag{2.34}$$

The dictionaries are learnt in the training phase; during disaggregation, we propose solving the following problem,

$$\min_{Z_i} \left\| X - \begin{bmatrix} D_1 \dots D_n \end{bmatrix} \begin{bmatrix} Z_1 \\ \vdots \\ Z_n \end{bmatrix} \right\|_1 + \lambda \sum_i \|Z_i\|_{NN}, \tag{2.35}$$

A. Deriving a solution for Training Phase

Solving the robust dictionary learning problem, subject to nuclear norm penalty is new. Here we adopt the Split Bregman approach. We substitute (dropping the subscripts for notational simplicity) and introduce a Bregman relaxation variable, B. This leads to the following formulation of (2.33),

$$\min_{D,Z,P} \|P\|_1 + \lambda \|Z\|_{NN} + \mu \|P - X + DZ - B\|_F^2, \tag{2.36}$$

Alternating minimization of (2.36) leads to the following sub-problems:

$$P1 : \min_{D} \|P - X + DZ - B\|_F^2, \tag{2.37}$$

$$P2 : \min_{Z} \lambda\|Z\|_{NN} + \mu\|P - X + DZ - B\|_F^2, \tag{2.38}$$

$$P3 : \min_{P} \|P\|_1 + \mu\|P - X + DZ - B\|_F^2, \tag{2.39}$$

Solving P1 is straightforward, it is a least squares problem with analytic solution. P2 is a nuclear norm regularized least squares minimization problem. This is efficiently solved using singular value shrinkage [73, 74]. The last sub-problem P3, also has a closed form solution in the form of soft thresholding [75].

B. Deriving a solution for Disaggregation Phase

We follow the Split Bregman approach here as well. We make the substitution $P = X - DZ$ where $D = \begin{bmatrix} D_1 & \dots & D_n \end{bmatrix}$ and $Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_n \end{bmatrix}$. After introducing the Bregman relaxation variable, the problem 2.35 takes the form,

$$\min_{Z,P} \|P\|_1 + \mu\|P - X + DZ - B\|_F^2 + \lambda \sum_i \|Z_i\|_{NN}, \tag{2.40}$$

As before this can be decomposed into the following sub-problems:

$$P1 : \min_{P} \|P\|_1 + \mu\|P - X + DZ - B\|_F^2, \tag{2.41}$$

$$P2 : \min_{Z} \mu\|P - X + DZ - B\|_F^2 + \lambda \sum_i \|Z_i\|_{NN}, \tag{2.42}$$

We have already discussed the solution for P1; it is solved via soft thresholding. Solving P2 is however tricky. To solve this, we need to decouple the problem. This can be done via Majorization-Minimization (MM) [75]. We represent P2 in a simpler fashion.

$$\|Y - DZ\|_F^2 + \eta \sum_i \|Z_i\|_{NN}, \tag{2.43}$$

where $Y = X + B - P$ and $\eta = \lambda/\mu$.

In every iteration, MM of (2.43) leads to decoupling -

$$\|T - Z\|_F^2 + \lambda \sum_i \|Z_i\|_{NN}, \tag{2.44}$$

where $T = Z_{k-1} + \frac{1}{\alpha} D^T(Y - DZ_{k-1})$; α is the maximum eigenvalue of $D^T D$.

The decoupled problem is easy to solve; it can be segregated for every appliance 'i' in the following way,

$$\min_{Z_i} \|T - Z\|_F^2 + \lambda\|Z_i\|_{NN}, \tag{2.45}$$

This is solved by one step of singular value shrinkage [73, 74].

C. Updating the Bregman Relaxation Variable

The final step is to update the relaxation variable B for all the problems. This is done by simple gradient descent.

$$B \Leftarrow X + DZ - B \tag{2.46}$$

There are two stopping criteria for the Split Bregman algorithm. Iterations continue till the objective function converges (to a local minima). The other stopping criterion is a limit on the maximum number of iterations. We have kept it to be 200.

2.4.2 Experimental Results

The proposed algorithm requires specifying the parameter λ and the hyperparameter μ. Some recent studies have shown that in a Split Bregman based technique, one can put λ =1 and only tune the μ. We use the simple L-curve method [76] for tuning the hyper-parameter. The number of dictionary atoms used is 144. The dictionary atoms are initialized by randomly picking up columns from the training data.

For energy disaggregation, we report results on the popular REDD dataset. The disaggregation accuracy is defined as follows [19] -

$$Acc = 1 - \frac{\sum_t \sum_i |\hat{y_t}^i - y_t^i|}{2\sum_t \bar{y}_t} \tag{2.47}$$

Table 2.2: Energy Disaggregation Results (in %)

Houses	Mode I			Mode II		
	RDL	PED	Prop	RDL	PED	Prop
1	70.1	81.6	82.8	52.1	46.0	54.2
2	61.9	79.0	79.7	55.7	49.2	58.4
3	61.0	61.8	65.9	43.3	31.7	48.9
4	71.0	58.5	73.5	59.8	50.9	63.8
6	64.7	79.1	79.9	60.0	54.5	65.9

where $\hat{y_t}^i$ denotes the algorithm's prediction for the *i*th device at the *t*th time step; the 2 factor in the denominator is to discount the fact that the absolute value will "double count" errors.

In the previous work [1] it was already shown that simple robust dictionary learning yields better results than the well known Factorial Hidden Markov Model (FHMM) [18] and sophisticated methods like discriminative sparse coding [32]. In this work, we show that the results from the proposed work are better than Robust Dictionary Learning (RDL) [1]. The results from proposed work were also compared with PED [33]. The comparative results are shown in Table 2.2.

As outlined in [19], there are two modes of testing. The first mode is simple, a portion of the data from every household is used as training samples and rest (from those households) is used for prediction. The second mode is more challenging, the data from four households are used for training and the remaining one is used for prediction; this is a more challenging problem.

Robust dictionary learning [1] is worse than PED [33] for mode 1 and better than PED for mode 2. The proposed method outperforms both the methods

[1, 33] for both modes.

2.4.3 Summary

In a previous work [1] it was shown that instead of using the standard Euclidean cost function for dictionary learning, better results are obtained with the more robust l_1-norm cost function. The reason has been discussed in the introduction. In this work, we extend the robust dictionary learning approach; we add a nuclear norm penalty on the coefficients. The reasoning behind this penalty is discussed in Section 2.4.1.

The resulting formulation is solved using a combination of Split Bregman and Majorization Minimization. The proposed technique is compared with robust dictionary learning [1] and powerlet energy disaggregation [33] and it outperforms both.

2.5 Proposed Robust Supervised Sparse Coding for NILM

As discussed in previous sections, a simple linear model does not hold in general. A seemingly unrelated research, in hyperspectral unmixing, faces a similar issue. A recent study [77], showed that the non-linear mixing problem can be approximated as a sum of linear mixing and non-linear perturbations. The perturbations can be assumed to be sparse, i.e. the effect is localized but may be of relatively large magnitude.

In this work we follow a similar approach - we assume that the linear mixing model for energy disaggregation holds in most cases (since this model is known to yield good results for simple loads), but that there are a few large perturbations arising out of the inherent non-linearity inside the load. The reasoning in favour of sparse non-linearities has already been discussed in section 2.3.

We assume an energy disaggregation paradigm defined in [32]; i.e. energy readings of individual appliances are available for training. In dictionary learning based techniques, the basis for representing each appliance is learnt separately, with the assumption that the noise is small/Normally distributed. Here we have argued that the noise/model error is actually not small - it is more likely to be large but sparse. In [1] it was shown that in such cases, solving the more robust l_1-norm instead improves the results significantly.

This work builds upon robust sparse coding approach [1]. To improve the results further we will incorporate supervision. In the first supervision, we will learn dictionaries such that they are incoherent; this ensures that the dictionaries

from different appliances look different from each other. In the second formulation, we introduce discriminating sparse codes, such that the codes generated for each appliance would not look alike.

2.5.0.1 Sparse coding based disaggregation

Kolter et. al [32], assumed that there is training data collected over time, where the smart meter logs only consumption from a single device. This can be expressed as X_i where i is the index for an appliance, the columns of X_i are the readings over a period of time.

For each appliance, a basis was learnt as expressed in eq.(2.48):

$$X_i = D_i Z_i, \;\; i = 1 \ldots N \tag{2.48}$$

where D_i represents the basis/ dictionary and Z_i are the loading coefficients, assumed to be sparse.

This is a typical dictionary learning problem with sparse coefficients - there are several ways to solve (2.48). All of them are variants of the following:

$$\min_{D_i, Z_i} \|X_i - D_i Z_i\|_F^2 + \lambda \|Z_i\|_1, \tag{2.49}$$

The dictionary (D) and the sparse codes (Z) are updated alternately.

During actual operation, several appliances are likely to be in use simultaneously. Like all prior studies, [54] makes the assumption that the aggregate power read by the smart meter is a sum of the powers for individual appliances.

Thus if X is the total power from N appliances (where the columns indicate smart meter readings over the same period of time as in training) the aggregate power can be modeled as:

$$X = \begin{bmatrix} D_1 | \ldots | D_N \end{bmatrix} \begin{bmatrix} Z_1 \\ \vdots \\ Z_N \end{bmatrix} = \sum_i D_i Z_i, \tag{2.50}$$

The dictionaries are already available during disaggregation operation. Therefore the problem is of estimating the sparse codes in (2.50). This is achieved by solving,

$$\min_{Z_1 \ldots Z_N} \|X - \sum_i D_i Z_i\|_F^2 + \lambda \|Z_i\|_1, \tag{2.51}$$

Once the Z_i's are obtained from (2.51) one can find out the power consumption of individual devices by:

$$\hat{X} = D_i Z_i, \;\; i = 1 \ldots N \tag{2.52}$$

Several modifications to the basic formulation has been proposed in [32, 33]. But the basic idea remains the same. In [1], the non-linear perturbation model was assumed and instead of minimizing the Euclidean norm for the data fidelity terms, the l_1-norm was used instead.

2.5.0.2 Robust non-linear hyperspectral unmixing

In hyperspectral imaging a scene is acquired at a large number wavelengths. Such an acquisition has fine spectral resolution but poor spatial resolution. Hence pixel values of an hyperspectral image has contributions from multiple sources. The task is to find out the composition of the scenery, i.e. distribution of the constituent elements. Mathematically this is represented as follows:

$$X = MA \tag{2.53}$$

Here X is the hyperspectral datacube, its columns are the images at different wavelengths. The rows are the pixel values at different wavelengths. M is called the endmember matrix which consists of the signature of different elements at all the wavelengths. A is the abundance map - it shows how much of each element is present at the given pixel location. In the most general case, neither M nor A is known. They are estimated by solving the following -

$$\min_{M,A} \|X - MA\|_F^2 + \lambda \|A\|_1, \tag{2.54}$$

The sparsity in the abundance map follows from the fact that only a few end members are present at each location.

As can be seen, (2.53) is a linear mixing model. In reality, the contribution from different elements do not mix linearly. It is not possible to form the exact non-linear mixing model; in most cases the linear mixing model performs

reasonably well. Keeping in mind the success of linear mixing, [77] postulated that for most pixels the linear model holds but in certain areas the non-linearity becomes more pronounced - these few pixels can be treated as sparse perturbations where the linearity breaks down. Thus [77] proposed an improved model:

$$X = MA + E \tag{2.55}$$

where E is the sparse non-linear modelling error. Following this argument, instead of solving (2.54), [77] proposed to solve the robust version of it.

$$\min_{M,A} \|X - MA\|_1 + \lambda \|A\|_1, \tag{2.56}$$

This simple modification led to significant improvement in unmixing performance.

2.5.1 Proposed Supervised Models

2.5.1.1 Incoherent Dictionary Learning

In the basic robust sparse coding formulation proposed in [1] the data fidelity terms for both learning and testing are changed from the l_2-norms (2.48) and (2.50) to l_1-norms. In there it is just assumed that learnt dictionaries are distinct enough to disaggregate the appliances. However, the previous naive formulation does not have any in-built mechanism to learn dictionaries that look separate from each appliance. We address this shortcoming in the supervised

formulation.

$$\min_{D_i,Z_i} \|X_i - D_i Z_i\|_1 + \lambda \|Z_i\|_1 + \lambda_2 \|D_{i^c}^T D_i\|_F^2, \tag{2.57}$$

The first term is employs the l_1-norm data fidelity for robustness. The second term is for usual sparse coding. In the third term $D_{i^c}^T$ consists of all the dictionaries not including D_i. The third term is the penalty for coherence - we want to minimize the coherence; i.e. we want to make the dictionaries look as dissimilar as possible for each appliance. Such an incoherence term has been used previously in [78] for face recognition problems.

We follow the Split Bregman technique [79] is used here; this is in turn based on the proximal splitting approach [80]. We introduce a proxy variable, $P_i = X_i - D_i Z_i$ and a corresponding Bregman relaxation variable B_i. This leads to the following formulation:

$$\begin{aligned} \min_{P_i,D_i,Z_i} \|P_i\|_1 + \lambda \|Z_i\|_1 + \mu \|P_i - (X_i - D_i Z_i) \\ - B_i\|_F^2 + \lambda_2 \|D_{i^c}^T D_i\|_{F_2}, \end{aligned} \tag{2.58}$$

One can use alternating direction method of multipliers (ADMM) [81, 82] to update each of the variables separately. This leads to the following sub-

problems for the expression above.

$$P1 : \min_{P_i} \|P_i\|_1 + \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2, \tag{2.59}$$

$$P2 : \min_{D_i} \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2 + \lambda_2\|D_{i^c}^T D_i\|_F^2, \tag{2.60}$$

$$P3 : \min_{Z_i} +\lambda\|Z_i\|_1 + \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2, \tag{2.61}$$

P1 is a simple denoising problem; it has a closed form solution in the form of soft thresholding [83]. P3 is an l_1-norm minimization problem. It can be solved using iterative soft-thresholding [84]. Updating the dictionary is seemingly difficult. Here we minimize the coherence between the dictionary D_i (the one we are currently updating) against the dictionaries from all other classes obtained in the previous iteration. With this approximation, $D_{i^c}^T$ becomes a constant. Thus solving P2 becomes easy - it is just a ridge regression.

2.5.1.2 Disaggregating Dictionary Learning

In the previous formulation, we make the dictionaries look different from each other (incoherent), but it does not guarantee that the disaggregation results will be good - that is what is of more importance to us. In order to ensure that, we will add a term to the basic dictionary learning framework that will promote smaller disaggregation error.

$$\min_{D_i, Z_i} \|X_i - D_i Z_i\|_1 + \lambda_1\|Z_i\|_1 - \lambda_2\|X_{i^c} - D_i Z_{i^c}\|_F^2, \tag{2.62}$$

The disaggregation penalty (final term) says that the learnt dictionary should be such that the dictionary for the i^{th} appliance does not synthesize data for classes other than i. Here Z_{i^c} consists of all the coefficients not including i^{th} appliance and X_{i^c} contains all the samples not including the i^{th} appliance. Notice the negative sign, it means that we want to maximize the term so that disaggregation error is minimized.

This problem (2.62) is solved using the Split Bregman approach. As before, we introduce a proxy variable, $P_i = X_i - D_i Z_i$ and a corresponding Bregman relaxation variable B_i. This leads to the following formulation:

$$\min_{P_i, D_i, Z_i} \|P_i\|_1 + \lambda \|Z_i\|_1 + \mu \|P_i - (X_i - D_i Z_i) - B_i\|_F^2 \qquad (2.63)$$
$$-\lambda_2 \|X_{i^c} - D_i Z_{i^c}\|_F^2,$$

Ideally we need to update both the dictionary as well as the coefficients based on the disaggregation penalty; but here we are more interested in the dictionary so we only use the penalty to update D_i assuming Z_{i^c} to be a constant. While learning the coefficients, we do not consider the disaggregation penalty (mainly because the coefficients are not useful after the dictionaries are learnt for the actual disaggregation problem). This small change, makes the algorithm simpler.

Via alternating direction method of multipliers, we separate the expression

above into the following (simpler) sub-problems:

$$P1 : \min_{P_i} \|P_i\|_1 + \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2, \tag{2.64}$$

$$P2 : \min_{D_i} \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2 + \lambda_2\|X_{i^c} - D_i Z_{i^c}\|_F^2, \tag{2.65}$$

$$P3 : \min_{Z_i} +\lambda\|Z_i\|_1 + \mu\|P_i - (X_i - D_i Z_i) - B_i\|_F^2, \tag{2.66}$$

From the previous sub-section we have already learnt how to solve P1 and P3. Solving P2 is also straight-forward; it is just a least squares minimization problem having an analytical solution.

2.5.1.3 Disaggregation

So far we have discussed the various dictionary learning techniques. Once we learn the dictionaries we use it for disaggregation. Prior works assumed a linear mixing model, i.e. the total power consumed by all appliances (as read by the meter) is a sum of the individual power consumptions, i.e.

$$X = \sum_i X_i \tag{2.67}$$

However, we argued that the simple linear mixing model is not always correct. In such a scenario we allow for sparse non-linear perturbations (E). Following this, instead of (2.67) we model total energy consumption as:

$$X = \sum_i X_i + E \tag{2.68}$$

As in [3, 4] the individual appliances are modeled as: $X_i = D_i Z_i$ .Therefore (2.68) can be expressed as follows,

$$X = \sum_i D_i Z_i + E \tag{2.69}$$

The dictionaries have been learnt before. The only task is to estimate the loading coefficients. This is achieved by minimizing the least absolute deviations.

$$\min_{Z_i} \left\| X - \left[D_1 | \dots | D_N \right] \begin{bmatrix} Z_1 \\ \vdots \\ Z_N \end{bmatrix} \right\|_1 + \lambda \sum_i \|Z\|_1, \tag{2.70}$$

This is efficiently solved using YALL1 algorithm [72].

2.5.2 Experimental Results

2.5.2.1 REDD Dataset

We evaluate the proposed energy disaggregation framework on the real-world REDD dataset [19], a large publicly available dataset for energy disaggregation. The details of this dataset has been given in section 1.3. This evaluation protocol has been outlined in [32] and [33].

The disaggregation accuracy is defined as follows [19] -

$$Acc = 1 - \frac{\sum_t \sum_i |\hat{y}_t^{\,i} - y_t^i|}{2 \sum_t \bar{y}_t} \tag{2.71}$$

where t denotes time instant and n denotes a device; the 2 factor in the denominator is to discount the fact that the absolute value will "double count" errors. Here y_t denotes the actual (measured) power, $\hat{y}_t$ the estimated power and $\bar{y_t}$ the mean of the actual.

The performance of the proposed method is compared with the Factorial HMM (FHMM) based technique [19], the Powerlet based Energy Disaggregation (PED) [33], the discriminative sparse coding (DiscSC) method outlined in [32], and robust sparse coding (RSC) proposed in [1]. As outlined in [19], there are two modes of testing. In the 'Training' mode, a portion of the data from every household is used as training samples and rest (from the same households) is used for prediction. In the 'Testing' mode, the data from four households are used for training and the remaining one is used for prediction; this is a more challenging problem.

The methods are dubbed incoherent dictionary learning (IDL) and disaggregating dictionary learning (DDL). For both the methods the parameters λ_1 and λ_2 need to be tuned. The value of λ_1 has been fixed at 0.1. The value of λ_2 has been fixed at 0.25 for IDL and 0.2 for DDL. These values were tuned by greedy L-curve method. The value of the hyper-parameter μ in both cases is fixed to 0.1; we found that the algorithm is robust to the value of μ and it does

Table 2.3: Comparative results on REDD

House	Training Mode						Testing Mode					
	FHMM	DiscSC	RSC	PED	IDL	DDL	FHMM	DiscSC	RSC	PED	IDL	DDL
1	71.5	81.6	75.8	74.5	75.9	85.5	46.6	46.0	44.2	43.8	45.0	50.2
2	59.6	79.0	69.7	70.2	72.1	82.1	50.8	49.2	48.7	48.5	50.6	53.4
3	59.6	61.8	61.9	62.4	64.4	69.2	33.3	31.7	30.1	31.0	32.1	38.9
4	69.0	58.5	67.5	66.9	67.2	71.9	52.0	50.9	46.3	48.2	49.6	56.8
6	62.9	79.1	69.9	63.4	66.0	81.3	55.7	54.5	50.4	51.6	52.1	59.0
Total	64.5	72.0	66.4	67.5	**68.9**	**78.0**	47.7	46.5	43.9	44.6	**45.9**	**51.7**

not require much tuning.

- FHMM is a relatively old technique and yields the worst results. Amongst all the prior techniques PED yields the best results.
- We find that just by modeling for non-linearities in the basic RSC, results improve over sophisticated approaches like PED and DiscSC which does not consider such perturbations.
- There is only slight improvement from the basic robust dictionary learning and the incoherent dictionary learning. This is because the incoherent dictionaries only look different from each other, but they produce almost the same features.
- The proposed robust dictionary learning technique with disaggregation penalty yields the best results. It is considerably superior even compared to the recently proposed powerlet energy disaggregation (PED) as can be seen in Table 2.3

H.No.	PED	RSC	IDL	DDL
1.	94.92	94.04	95.69	98.64
2.	64.75	68.50	71.34	74.65
3.	83.60	83.31	84.92	87.91
4.	43.67	47.68	51.33	54.01
5.	63.21	66.05	68.81	71.78
6.	43.23	51.24	56.66	59.53
7.	58.89	64.81	70.31	73.33
8.	51.41	57.32	59.81	63.10
9.	58.39	62.10	65.40	68.38
10.	62.96	64.85	66.00	69.43
11.	50.37	57.65	62.20	65.26
12.	58.13	64.09	64.78	67.98
13.	51.97	56.25	58.32	61.03
14.	78.26	79.03	78.81	81.83
15.	64.06	65.84	64.11	67.34
16.	68.09	77.23	82.83	85.83
17.	63.34	75.38	78.71	82.24
18.	67.29	72.44	71.40	74.77
19.	69.05	82.63	87.84	90.88
20.	91.86	92.76	94.67	97.72
21.	48.50	56.16	61.79	64.55
22.	51.27	55.55	57.92	60.88
23.	69.00	76.82	78.28	82.04
24.	85.10	88.75	90.63	93.61
25.	59.04	58.05	57.93	60.88
26.	91.20	90.57	92.25	95.22
27.	69.48	70.16	74.37	77.48
28.	82.27	82.21	83.55	86.59
29.	65.07	70.76	74.08	77.48
30.	50.99	61.53	66.37	68.96
31.	55.60	61.08	62.47	65.63
32.	47.91	56.49	60.77	63.94
33.	50.87	54.80	61.41	64.41
34.	74.76	76.74	77.42	80.43
35.	60.88	64.60	66.46	70.11
36.	54.36	59.09	63.07	66.51
37.	56.55	54.72	53.33	57.10
38.	62.08	66.38	69.34	72.46
39.	83.42	85.32	86.96	89.97
40.	52.68	57.62	56.40	59.65

Figure 2.1: Comparative results on Dataport

41.	86.15	87.18	88.50	91.49
42.	87.06	89.93	89.15	92.14
43.	77.27	78.07	77.64	80.86
44.	56.66	59.36	62.25	65.06
45.	57.49	61.46	67.18	70.55
46.	58.38	62.68	62.27	65.58
47.	47.76	52.99	54.11	56.80
48.	80.05	84.03	88.13	91.09
49.	78.36	86.52	87.32	90.29
50.	49.82	49.83	50.89	54.29
51.	48.71	51.85	51.50	54.24
52.	45.77	47.40	52.37	55.20
53.	78.53	77.88	79.42	82.41
54.	52.31	49.58	52.05	55.09
55.	40.52	50.97	52.50	55.89
56.	53.69	63.37	64.39	67.68
57.	66.31	69.59	72.58	76.09
58.	70.41	76.71	77.25	80.58
59.	75.50	76.18	79.13	81.96
60.	53.59	59.77	59.46	62.57
61.	47.13	50.19	52.29	54.98
62.	52.54	55.48	55.90	59.24
63.	53.27	57.96	59.17	72.32
64.	74.97	81.65	83.15	86.56
65.	50.99	54.26	55.02	57.70
66.	46.88	52.89	57.02	60.26
67.	58.14	63.68	64.41	67.33
68.	73.87	81.74	82.61	85.95
69.	57.67	57.83	59.13	62.25
70.	79.91	83.64	84.88	88.45
71.	47.82	54.06	56.12	59.32
72.	79.47	82.37	85.87	89.11
Total	**63.13**	**67.25**	**69.34**	**72.45**

Figure 2.2: Comparative results on Dataport

2.5.2.2 Dataport

We conduct this experiment on a subset of Dataport dataset [85] available in NILMTK (non-intrusive load monitoring toolkit) format. The details of this dataset are given in section 1.3.

To prepare training and testing data, aggregated and sub-metered data are averaged over a time period of 10 minutes. This is the usual protocol to carry out experiments on the Pecan street dataset. Each training sample contains power consumed by a particular device in one day while each testing sample contains total power consumed in one day in particular house.

The same metric as before (disaggregation accuracy) has been used to compare the different techniques. The results are shown in the following table. In the previous set of experiments on REDD, we have seen that FHMM and DiscSC always yields considerably worse results than the others. Therefore we do not show these results here.

The conclusions remain the same as before. The proposed method yields the best results.

For this dataset, the efficacy of the proposed method is checked in another fashion. We compute the normalized error for common appliances. Lower the error (between the measured and estimated), the better are the disaggregation results. These values are shown in Table 2.4. One conclusively finds that the proposed method yields the best results. In fact RSC also yields very good

Table 2.4: Normalized error for common devices

Appliance	FHMM	DiscSC	RSC	PED	IDL	DDL
AC	3.16	0.90	0.70	2.52	0.89	0.80
Dryer	51.47	16.57	2.04	35.69	1.11	1.02
Dishwasher	6.48	4.23	1.25	6.08	0.66	0.62
Microwave	4.96	4.55	0.84	4.3	0.76	0.70
Furnace	0.89	0.79	0.63	0.93	0.58	0.55
Fridge	2722.8	916.53	516.3	986.3	490.56	401.78
Washer	21.80	8.75	0.93	19.62	0.59	0.55

results. This is because, all the robust techniques can account for the anomalies which the others cannot.

2.5.3 Summary

Most prior studies in energy disaggregation assumes a linear mixing model; we have argued that this is an oversimplification and does not hold in general. However most prior studies in this area are based on computer science researchers who did not account for such subtle nuances of electrical engineering. The first work on this topic [1] modelled the non-linearities as sparse perturbations and proposed robust sparse coding / dictionary learning for disaggregation. It showed improvement over previous approaches.

This work proposes improvements over [1]. The first one learns incoherent dictionaries for different appliances; the second one learns a dictionary such that the disaggregation error is minimized. The proposed method yields considerably better results than benchmark disaggregation techniques.

In recent time, there has been significant progress in this front, researchers have used deeper versions of dictionary learning [53], and analysis versions

of dictionary learning [86]. It would be interesting to see, how the proposed modifications improve upon the aforesaid. Recent works also show a shift in the nature of the input signal. For example in [87, 88], instead of using power data, one uses high frequency electro-magnetic interference for disaggregation. We believe that this technique, with in-built capabilities for incoherence can succeed with such signals.

2.6 Proposed Transform Learning for NILM

The generality of dictionary learning methods in energy disaggregation motivates this work. Standard dictionary learning is a synthesis formulation. It learns a basis so as to synthesize the data along with the learnt coefficients. In the past decade, it has enjoyed significant amount of success in machine learning and signal processing. This work is based on the analysis version of dictionary learning - Transform learning [89, 90, 91].

Most NILM techniques are based on a learning based paradigm. The training stage is intrusive. It requires instrumenting the homes at plug level for collecting data that is later used for learning appliance specific models. These models are later used for disaggregating during the operational phase; at this stage the plug level sensors are removed. There are a few recent studies that do not need actual power consumption from every device and can act on aggregate data; they are based on the multi-label classification paradigm [34, 35, 36]. Studies such as [49, 38, 92] proposed completely unsupervised techniques for energy disaggregation. However, the accuracy of unsupervised techniques is generally less than supervised methods [36].

For supervised approaches, data acquisition at the training phase is expensive owing to the cost of buying/renting the plug level sensors - this determines the 'cost' of NILM. If one can reduce the training phase by reducing the period over which data needs to be collected (for training mode disaggregation) or reduce the number of homes from which the data needs to be collected (for testing

mode) the cost of NILM will reduce proportionately. This will have significant cost implications on the utilities and the consumers. More number of consumers will benefit without increase in any data acquisition cost.

Everything else remaining the same, a transform can generalize better than a dictionary, i.e. it has better representation capability. In other words, for a problem of fixed complexity the size of the transform can be much smaller than a given dictionary. Therefore, given limited volume of training data, the smaller sized transform will be less prone to overfitting than the dictionary. This is the major benefit of transform learning - it can learn from far less training data. This in turn means that practical advantage mentioned in the previous paragraph will become feasible. This motivates the proposed formulation.

2.6.0.1 Synthesis Sparse Coding

The idea of learning a basis for modelling each appliance was introduced by [32]. It follows the typical NILM scenario. Training data is collected over time, where the smart meter logs consumptions from every single device. This is achieved by plug level monitors (such as jPlug). The training data is expressed as X_i where i is the index for an appliance, the columns of X_i are the readings over a period of time (say every hour of the day) and the rows are the days. For each appliance they learnt a dictionary, i.e. they expressed,

$$X_i = D_i Z_i, \;\; i = 1 \ldots N \tag{2.72}$$

where D_i represents the basis/ dictionary, Z_i are the loading coefficients, assumed to be sparse and N is the total number of appliances. X_i will have a dimension of hourly sampling rate along the rows and dimension of 24 (hours) x the number of training days along the columns. The number of dictionary atoms (columns) for D_i has to be specified by the user.

In the training phase a dictionary is learnt to model each appliance. This is achieved by solving -

$$\min_{D_i, Z_i} \|X_i - D_i Z_i\|_1 + \lambda \|Z_i\|_1, \;\; i = 1 \ldots N \tag{2.73}$$

Since Z_i's are supposed to be loading coefficients, they are supposed to be positive. This is assured by projecting the solution of eq.(2.73) in every iteration to the positive space.

The basic interpretation of the dictionary is that its columns act as an abstract basis for representing an appliance. The power consumption is therefore a linear combination of these basis. To give a more concrete example, consider an electric fan; the columns/ basis in the dictionary can be thought of as its distinct states (say 1 to 5); the coefficients then are the proportions of how long each state had run during the time period. This is the reason for the positivity

constraint.

This concludes the basic dictionary learning approach to NILM. In a quest to improve the results, [32] introduced other complicated penalties; however in practice the pay off from these penalties was nominal since the results did not improve much over the basic formulation. This model in eq.(2.73) does not account for the time varying nature of the appliances. This was accounted for in [33] where they introduced an auto-regressive model on the dictionary atoms.

So far we discussed about the training stage. The test/ operational stage remains the same for all methods. Let X denote the total power from N appliances (the columns indicate smart meter readings over the same period of time as in training). This is expressed as:

$$X = \sum_{i=1}^{N} X_i = \sum_{i=1}^{N} D_i Z_i \tag{2.74}$$

Given the additive model, one can estimate the loading coefficients for each appliance by solving the following sparse recovery problem,

$$\min_{Z_i} \left\| X - \begin{bmatrix} D_1 \dots D_N \end{bmatrix} \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_1 + \lambda \left\| \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_1, \tag{2.75}$$

The interpretation here is that, given the basis for each device, one needs to estimate the corresponding loading coefficients.

As before, positivity constraints are enforced on the loading coefficients es-

timated from eq.(2.75). The formulation for disaggregation (2.75) is convex. From the estimated loading coefficients the device level power consumption can be computed.

$$\hat{X}_i = D_i Z_i, \, i = 1 \ldots N \tag{2.76}$$

In a very recent work [53], a deep version of sparse coding has been proposed. In there, multiple layers of dictionaries are learnt for each appliance; the rest of the mechanism remains the same. This is by far the state-of-the-art for standard supervised evaluation protocols.

In a recent work [86], a co-sparse analysis formulation has been proposed. Co-sparsity means that the signal is sparse under analysis. The main difference from the sparse coding/ dictionary learning formulation is that, [86] learns a co-sparsity promoting dictionary. This is given by -

$$\min_{D_i, \hat{X}_i} \|X_i - \hat{X}_i\|_F^2 + \lambda \|D_i \hat{X}_i\|_1, \tag{2.77}$$

Here co-sparsity is accounted for by the $\|D_i \hat{X}_i\|_1$ term. A device specific analysis basis is learnt such that the clean version of the data ($\hat{X}_i$) is co-sparse. The major motivation for moving from the synthesis to the analysis paradigm is to reduce the problem of over-fitting in limited data regimes. This will be explained in detail later.

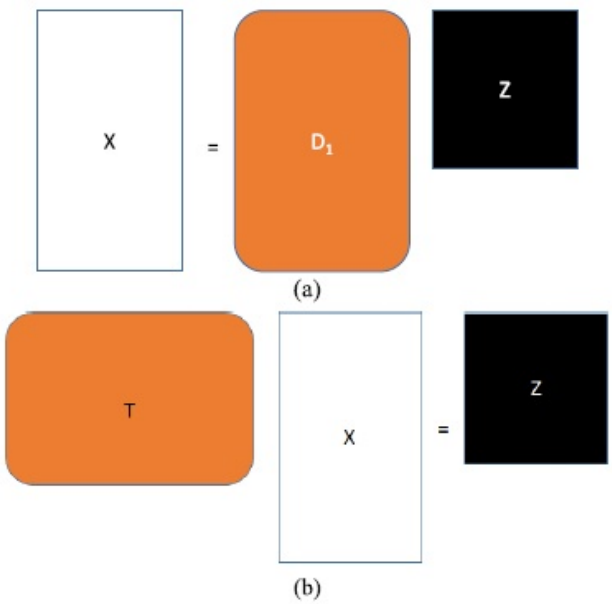


Figure 2.3: (a) Dictionary Learning; (b) Transform Learning

2.6.0.2 Transform Learning

Dictionary learning is a synthesis formulation (Fig. 2.3a), it synthesizes/generates the data (X) from the linear combination of atoms (D) and learnt coefficients (Z). It has been used profusely for analysis [86, 93] and synthesis problems [94]. Transform learning is the analysis equivalent (Fig. 2.3b). It learns a transform (T) so that it operates/analyses the data (X) to generate the coefficients (Z).

Unlike dictionary learning/ sparse coding, transform learning is relatively new. For the interested reader, we request to peruse [89, 90]. We will discuss the formulation briefly for the sake of completeness. Transform learning analyses the data by learning a transform/ basis to produce coefficients. Mathematically

this is expressed as,

$$TX = Z \tag{2.78}$$

Here T is the transform, X is the data and Z is the corresponding coefficients. The data matrix X consists of the features along the rows and samples along the columns. The number of transform atoms are determined by the user; this also defines the coefficient matrix Z.

The following transform learning formulation was proposed in [89] and [90]

-

$$\min_{T,Z} \|TX - Z\|_F^2 + \lambda(\|T\|_F^2 - \log\det T) + \mu\|Z\|_1, \tag{2.79}$$

The factor - $\log\det T$ imposes a full rank on the learned transform; this prevents the degenerate solution $(T = 0, Z = 0)$. The additional penalty is to balance scale; without this $\log\det T$ can keep on increasing producing degenerate results in the other extreme.

An alternating direction approach was proposed to solve (2.79). This is given by -

$$Z \leftarrow \min_{Z} \|TX - Z\|_F^2 + \mu\|Z\|_1, \tag{2.80}$$

$$T \leftarrow \min_{Z} \|TX - Z\|_F^2 + \lambda(\epsilon\|T\|_F^2 - \log\det T), \tag{2.81}$$

Updating the coefficients in (2.80) is straightforward. It can be updated via

one step of soft thresholding. This is expressed as,

$$Z \leftarrow signum(TX) \bigodot \max(0, abs(TX) - \mu), \tag{2.82}$$

Here $\bigodot$ indicates element-wise product.

In the initial paper on transform learning [32], a non-linear conjugate gradient based technique was proposed to solve the transform update. In the more refined version [33] with some linear algebraic tricks they were able to show that a closed form update exists for the transform.

$$XX^T + \lambda\epsilon I = LL^T, \tag{2.83}$$

$$L^{-1}XZ^T = USV^T(SVD), \tag{2.84}$$

$$T = 0.5V(S + (S^2 + 2\lambda I)^{1/2})U^T L^{-1} \tag{2.85}$$

The first step is to compute the Cholesky decomposition; the decomposition exists since $XX^T + \lambda\epsilon I$ is symmetric positive definite. The next step is to compute the full SVD. The final step is the update step. The proof for convergence of such an update algorithm can be found in [91].

There are only a handful of papers on this topic. Theoretical aspects of transform learning are discussed in the aforesaid papers on transform learning. So far it has limited visibility outside the signal processing community. The only application of transform learning in machine learning has been by [95] and

[96] where the same formulation has been dubbed as 'analysis sparse coding'. There it was used for simple unsupervised feature extraction. A later study [94] proposed a discriminative version for supervised feature extraction. In signal processing, it is mainly used for solving inverse problems [97, 98, 99].

2.6.1 Proposed Approach

Today dictionary learning is a popular representation learning tool. Standard dictionary learning is a synthesis approach. However, recent studies in analysis dictionary learning (such as [86]) empirically show improvement over its traditional synthesis counterpart; especially in limited data regimes. Analysis dictionary learning/transform learning is less prone to overfitting [86], [96], and [100], i.e. can learn from fewer samples.

In disaggregation, datasets such as REDD (testing mode) define 4:1 splits for training and testing [19]. These are impractical training scenarios. In most practical cases, it is not possible to instrument so many houses with sensors - the situation is exactly the opposite. One needs to disaggregate/ test on a large number of houses by learning from far fewer labeled (instrumented) households. For the training mode (where the data from the same house if used for training), one needs to ensure that the number of training days is minimized. This would directly benefit the utilities and consumers. It would enable the utilities to instrument more houses (for collecting training data) and thus bring the benefits of disaggregation to more consumers. In such data scarce scenarios, it is likely that, analysis transform learning, with its capacity to learn from fewer samples

will yield better generalizability on unseen (test) cases than the corresponding synthesis dictionary learning technique.

2.6.1.1 Training

The methodology/ protocol is exactly the same as in dictionary learning [32]. As before, given the training data for each device, we learn a device specific transform. Assuming X_i is the training data for the i^{th} device, this is expressed as,

$$T_i X_i = Z_i \tag{2.86}$$

The straightforward way to solve this problem is to learn one transform for each device in a naive fashion. Here X_i has the same meaning as (2.72).

$$\min_{T_i, Z_i} \sum_i \|T_i X_i - Z_i\|_F^2 + \lambda(\|T_i\|_F^2 - \log\det T_i) + \mu\|Z_i\|_1, \tag{2.87}$$

For disaggregation we would expect that the transforms are discriminative, i.e. the transform for the i^{th} appliance should only produce sparse codes for the i^{th} appliance but not represent any other appliances; i.e. should not generate sparse codes for other appliance. This means that $T_i X_{i^c}$ (the superscript 'c' on 'i' indicates complimentary set of appliances) should not be sparse - they should be dense and small.

The basic formulation (2.87) does not enable this. Therefore we need to modify (2.87). We achieve this by adding the additional term (for each i) $\|T_i X_{i^c}\|_F^2$; the Frobenius norm would ensure that the coefficients obtained from $T_i X_{i^c}$ should be dense and very small (approximately Normal distribution).

Incorporating these terms into (2.87) leads to,

$$\min_{T_i, Z_i} \sum_i \|T_i X_i - Z_i\|_F^2 + \lambda(\|T_i\|_F^2 - \log\det T_i) \\ +\mu\|Z_i\|_1 + \gamma\|T_i X_{i^c}\|_F^2, \tag{2.88}$$

We follow the same alternating minimization approach as proposed by [89] and [90]. The formulation can be decoupled for each device. Alternating minimization leads to,

$$\min_{T_i} \sum_i \|T_i X_i - Z_i\|_F^2 + \lambda(\|T_i\|_F^2 - \log\det T_i) + \gamma\|T_i X_{i^c}\|_F^2, \tag{2.89}$$

$$\min_{Z_i} \sum_i \|T_i X_i - Z_i\|_F^2 + \mu\|Z_i\|_1, \tag{2.90}$$

The sparse coding step (2.90) remains exactly the same as before (2.80). Hence can be solved using (2.82).

For the transform update, we can express (2.89) as,

$$\min_{T_i} \|T_i[X_i|\sqrt{\gamma}X_{i^c}] - [Z_i|0]\|_F^2 + \lambda(\|T_i\|_F^2 - \log\det T_i), \tag{2.91}$$

The $[\cdot|\cdot]$ means that the matrices are stacked horizontally. This brings the transform update to the same form as (2.90). Hence we can use the same technique as (2.83).

In a succinct fashion, the entire training algorithm can be expressed as in Algorithm 1.

Algorithm 1: Transform Learning for Load Disaggregation

Require: X_i

Initialize: $X_i = USV^T; Z_i = SV^T$

Until convergence run

$$[X_i|\sqrt{\gamma}X_{i^c}]\begin{bmatrix} X_i^T \\ \sqrt{\gamma}(X_{i^c}))^T \end{bmatrix} + \lambda\epsilon I = LL^T$$

$$L^{-1}[X_i|\sqrt{\gamma}X_{i^c}]\begin{bmatrix} Z_i^T \\ 0 \end{bmatrix} = USV^T$$

$$T_i = 0.5V(S + (S^2 + 2\lambda I)^{1/2})U^T L^{-1}$$

$$Z_i \leftarrow \text{signum}\,(T_i X_i) \cdot \max(0, abs(T_i X_i) - \mu)$$

Note that even though both the proposed formulation and [86] are based on the analysis paradigm, they are completely different. In [86] an analysis basis is learnt so as to 'clean' the data; it does not learn any representation. In the proposed formulation, we learn an analysis transform to generate the coefficients.

2.6.1.2 Testing

During testing, the transform based disaggregation is not as straightforward as the dictionary learning based formulation; it needs further analysis. We start with the standard model that the total power is the sum of the power consumed

by the individual devices. This is expressed as,

$$X_i = \sum_i X_i \tag{2.92}$$

Applying the learnt transform leads to -

$$\begin{bmatrix} T_1 \\ \dots \\ T_N \end{bmatrix} (X_1 + X_2 + ... + X_N) \tag{2.93}$$

$$= \begin{bmatrix} T_1(X_1 + X_2 + ... + X_N) \\ \dots \\ T_N(X_1 + X_2 + ... + X_N) \end{bmatrix} \tag{2.94}$$

$$= \begin{bmatrix} T_1X_1 \\ \dots \\ T_NX_N \end{bmatrix} + \begin{bmatrix} T_1(X_2 + .. + X_N) \\ \dots \\ T_N(X_2 + .. + X_N) \end{bmatrix} \tag{2.95}$$

The terms $T_iX_i^c$'s will be close to zero or negligibly small - this follows from the training formulation. We have learnt the transforms in such a manner that the transforms for one device when applied on the data for another, will produce almost zero valued coefficients. This allows representing 2.95,

$$\begin{bmatrix} T_1X_1 \\ \dots \\ T_NX_N \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \dots \\ \epsilon_2 \end{bmatrix} \tag{2.96}$$

Here $T_1(X_2 + .. + X_N) = \epsilon_1$, $T_2(X_2 + .. + X_N) = \epsilon_2$ and so on. The error terms

ϵ_i's are small (approximately Normal distribution). In a concise fashion, from (2.92), (2.95) and (2.96) we have the following expression

$$\begin{bmatrix} T_1 \\ \dots \\ T_N \end{bmatrix} X = \begin{bmatrix} T_1 X_1 \\ \dots \\ T_N X_N \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \dots \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \dots \\ \epsilon_2 \end{bmatrix} \tag{2.97}$$

where $T_i X_i = Z_i$ as per the definition of the transform.

The transforms for each device have already been learnt during the training phase. Therefore one can solve for the coefficients from (2.97) by sparse coding.

$$\min_{Z_i} \left\| \begin{bmatrix} T_1 \\ \dots \\ T_N \end{bmatrix} X - \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_F^2 + \mu \left\| \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_1 \tag{2.98}$$

The l_2-norm on the data-fidelity arises from the fact that the errors ϵ_i's are Normally distributed (by definition from the training phase). Solving (2.98) is straightforward. It requires one step of soft-thresholding, similar to (2.82). Once the sparse codes are obtained, one needs to solve the following set of inverse problems to generate the corresponding power consumptions for each device.

$$T_i X_i = Z_i \tag{2.99}$$

This has an analytic solution - Moore Penrose pseudo-inverse.

$$X_i = (T_i^T T_i)^{-1} T_i^T Z_i \tag{2.100}$$

Notice that even though the analysis of the Transform based testing algorithm is slightly more involved than the dictionary counterpart; operationally/ computationally it is much simpler and efficient. Both (2.98) and (2.100) have closed form solutions. For (2.98) one only requires a simple thresholding step; for (2.100) one needs a matrix vector product (the pseudoinverses for the T_i's can be pre-computed). Thus, operationally it is very fast - capable of real-time disaggregation.

2.6.2 Results

The proposed algorithm is tested on benchmark datasets. For the sake of reproducibility we experimented on two publicly available ones - REDD and Pecan Street.

In principle, we could have applied this technique on any kind of electrical signal. However, in practice, smart meter readings are more practical. Hence, we stick to the traditional datasets.

2.6.2.1 REDD Dataset

The disaggregation accuracy is defined as follows [19] -

$$Acc = 1 - \frac{\sum_t \sum_i |\hat{y_t}^i - y_t^i|}{2 \sum_t \bar{y}_t} \tag{2.101}$$

where t denotes time instant and n denotes a device; the 2 factor in the denominator is to discount the fact that the absolute value will "double count" errors. Here y_t denotes the actual (measured) power, $\hat{y}_t$ the estimated power and $\bar{y_t}$ the mean of the actual.

We benchmark the proposed technique against - the Factorial HMM (FHMM) [18], Powerlet based Energy Disaggregation (PED) [33], discriminating sparse coding (discSC) [32] and deep sparse coding (DSC) [53]; on the standard training protocols (with enough training data) DSC is the most accurate disaggregating technique known.

Note that in this work, we do not compare with multilabel classification techniques. The reason has been discussed at the onset. These methods can only estimate the state of the appliance and cannot estimate its consumption directly. Power consumption can be indirectly estimated but that gives a very crude value, which is worse than simple disaggregation techniques like FHMM.

There are two protocols for evaluation [19]. In the first one (called 'training), a portion of the data from every household is used as training samples and rest (from those households) is used for prediction - this is the easier protocol. In

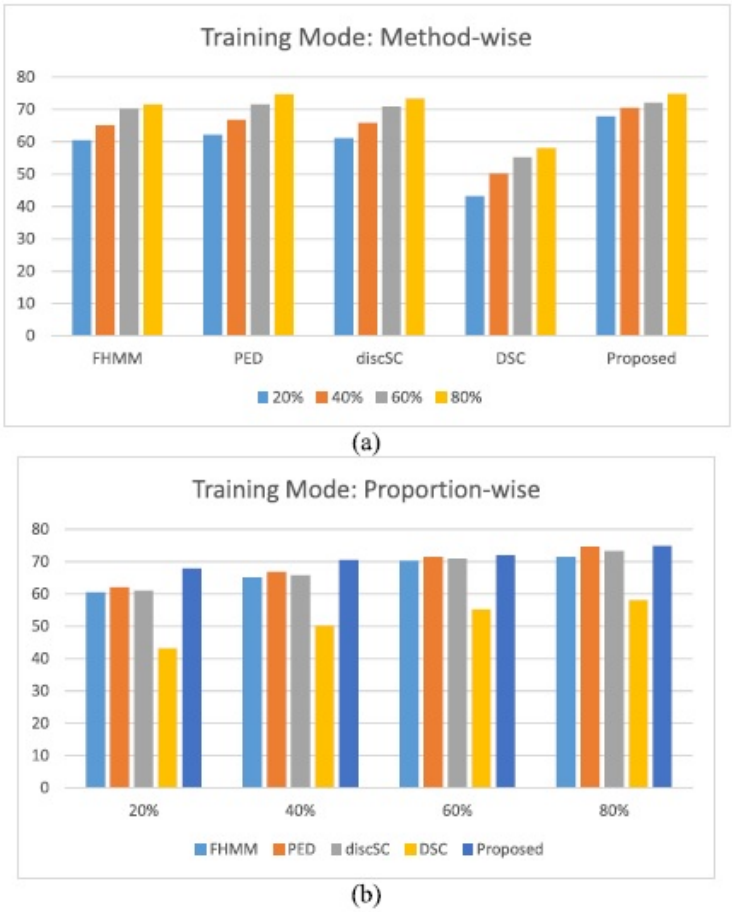
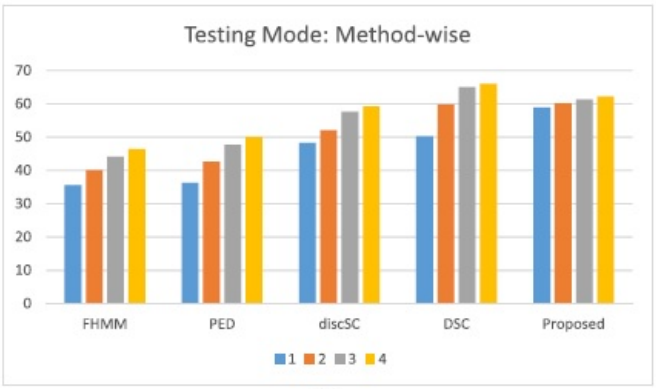


Figure 2.4: REDD training mode disaggregation results. Y-axis shows the disaggregation accuracy.

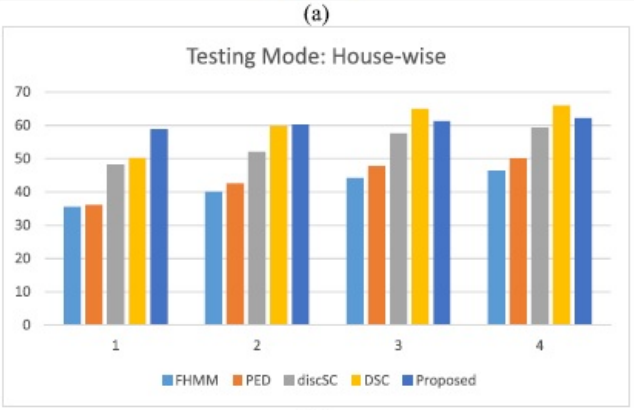
the second mode, the data from some households are used for training and the remaining ones are used for prediction (called 'testing').

Usually the split between training and testing is 4:1. This is an overtly optimistic scenario. In real life situations, the training data will always be small compared to the testing data. In this work, we will show that for smaller volumes of training data (practical situation), the proposed method yields better results than all other techniques.

We first show results for the training mode. The results are shown in Fig. 2.4. We are showing the mean disaggregation accuracy over all houses. In the graph,



(a)



(b)

Figure 2.5: REDD testing mode disaggregation results. Y-axis shows the disaggregation accuracy.

the X-axis shows the percentage of data (from each house) used for training; the Y-axis the corresponding aggregate (over all houses) disaggregation accuracy on the remaining portion (of each house). Since this is a small dataset, in order to make the results reproducible we have always taken the portion of training data starting from the beginning.

The parameter setting for the benchmarking techniques have been taken from the respective studies. Even though, the training and testing ratios differ between the aforesaid work and ours, the configurations will remain the same. This is because the configuration, for example the number of states of FHMM or the number of basis used for discSC, PED or DSC are representatives of the complexity of the device and not the volume of training data available. Since

the devices remain the same, the configurations remain the same as well.

For FHMM, we use the function from NILMTK1; it is available there with pre-tuned parameters for these datasets. For discSC, we need to specify the sparsity promoting parameter (which is 2), the discriminative parameter (which is 0.001) and the number of atoms (twice redundant). For DSC, we use the sparsity promoting parametric value of 0.1 and the number of dictionary atoms are reduced by half in each subsequent layer. For PED, sparsity promoting prior and the co-occurrence parameters were both set to unity and the temporal smoothness parameter is 30. As mentioned before, all these values are obtained from the corresponding papers.

For the proposed method, we have used 6 transform basis for each device; this is not a true representative of the complexity, e.g. a laptop or washing machine is more complex than a CFL or stove, and hence would require more basis - but such device specific fine tuning is time consuming. The parametric values used for transform learning have been tuned using the greedy L-curve method [76]. Here we first put μ to zero and tune λ; we obtain $\lambda = 0.1$ by the L-curve method. Then we fix the value of λ and tune μ to get a value $\mu = 0.5$.

In Fig. 2.4a we show how different methods perform with change in training volume; the results are grouped by the disaggregation technique. In Fig. 2.4b we show the same results in a different fashion; we see how different methods perform for a fixed training volume.

Especially from Fig. 2.4a, we find that the proposed method is the most ro-

bust. There is only a small drop in disaggregation accuracy across the various proportions of training data; the drop is around 7%. All other dictionary learning/sparse coding based methods drop more than 12%.

In [53] it has been claimed that deep sparse coding yields the best results; but we can see here that for limited training volume volume it performs bad. This is true for deep learning in general; they only yield good results when the volume of training data is large.

The next set of experiments are in testing mode.We choose k houses; for each k, there are 5_k^C possible combinations of training houses. We carry out experiments on all of them and test on the remaining houses for each training set. The results are shown in Fig. 2.5. The average disaggregation accuracy is reported. As before we have shown the same result in two different ways. In Fig. 2.5a the variation within a technique for changing training volume (number of houses) is shown, and in Fig. 2.5b, the variation among the methods for a fixed training volume is shown.

One finds that the testing mode results show a similar trend; especially from Fig. 2.5a. As the number of houses (training data) decreases, the disaggregation accuracy suffers as well. But the proposed method suffers the least drop in accuracy (less than 3% drop); all other methods show significant drops. FHMM and discSC drops by 12%, PED and DSC drops more than 15%. When the volume of training data is high, DSC yields better results than ours. But as the training volume decreases (practical scenario), we perform better.

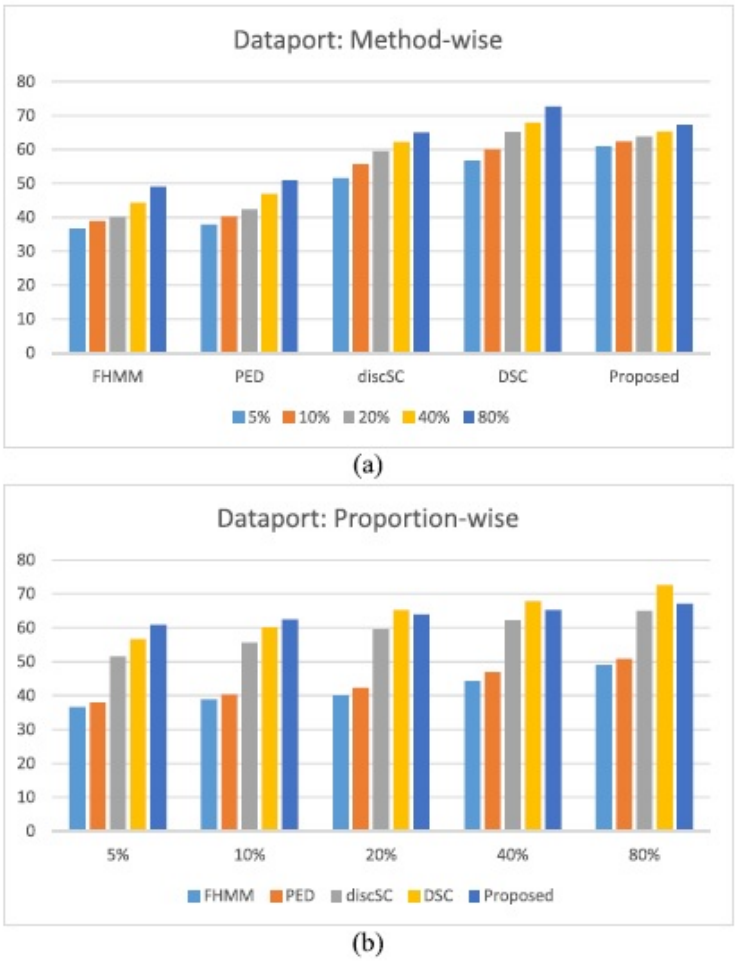


Figure 2.6: Dataport testing mode disaggregation results. Y-axis shows the disaggregation accuracy.

2.6.2.2 Dataport Pecan Street Dataset

The Dataport dataset is available in NILMTK (non-intrusive load monitoring toolkit) [20] format.

On this dataset, the usual protocol is to test on the 'testing' mode. Usually about 66% of the homes form the training dataset and the remaining 34% of the homes form the test set [20]. In this work, we train on fewer homes and test on the remaining; this is a more practical scenario. For each proportion of the training set, the homes are chosen randomly. This is done 100 times for each

configuration. For each such configuration, the remaining homes are used for testing. The mean disaggregation accuracy from various techniques is shown in Fig. 2.5.

As we did before for REDD, the training and testing data are prepared by aggregating the data for a period of 10 minutes. This reflects real life scenario and is the usual protocol on this dataset.

As in the previous sub-section, the configuration for the benchmarking techniques have been taken from the respective papers; the same parametric values mentioned before have been used. The configuration for the proposed algorithm and parametric values also remain the same.

In Fig. 2.6a we show the variation of disaggregation accuracy within a technique for changing proportions of training data. In Fig. 2.6b, the same results are shown in a different fashion; it shows the variation in accuracy among the methods given the training volume.

The results show a similar to that of REDD. As can be seen from Fig. 2.6a. For high volume of training data, deep sparse coding (DSC) yields very good results. But when the training volume decreases, the accuracy falls significantly; the drop in disaggregation accuracy is more than 15%. For FHMM and PED the drop in accuracy is around 13% and for discriminative sparse coding it is around 15%. For the proposed method, the drop is only 7%.

For this dataset, we show the normalized absolute error (defined below) on some common devices (indexed as i) in Table 2.5; the results are shown for two

Table 2.5: Normalized error for common devices at 40% and 5% training volumes

Appliance	FHMM		discSC		PED		DSC		Proposed	
	40%	5%	40%	5%	40%	5%	40%	5%	40%	5%
AC	3.16	10.21	0.70	2.41	2.52	4.17	0.89	3.11	0.92	1.11
Dryer	51.47	87.03	2.04	9.12	35.69	41.12	1.11	6.20	1.24	1.57
Dishwasher	6.48	12.11	1.25	9.70	6.08	14.89	0.66	3.97	0.76	2.24
Microwave	4.96	10.72	0.84	2.17	4.34	9.48	0.76	1.26	0.75	2.15
Furnace	0.89	2.31	0.63	1.29	0.93	2.05	0.58	1.04	0.61	1.16
Fridge	2722.82	2900.51	516.31	847.56	986.30	1182.41	460.56	774.61	467.78	514.08
Washer	21.80	27.63	0.93	1.97	19.62	28.17	0.59	2.19	0.62	1.76

training volumes - 40% and 5%.

$$Err(i) = \frac{\sum_t \sum_{(i)} |\hat{x_t}^{(i)} - x_t^{(i)}|}{\sum_t x_t^{(i)}} \tag{2.102}$$

The error metric normalizes the sum of the absolute differences between the predicted and actual power consumptions of the i^{th} device, summed over all instances of time.

One can see that as the training volume is reduced, the error increases sharply for all the methods compared against; apart from the refrigerator and the washer, the errors for the other devices increase by two fold. The proposed method is the least perturbed by the extreme change in training volume. In the low training sample regime, the proposed method yields the best results.

This has implications in cost; the training phase of energy disaggregation is intrusive; the appliances need to be sensed separately. Such sensors (such as

jPlug) are expensive. By the proposed method it will be possible to get the same disaggregation results with far lesser cost (fewer houses need to be instrumented for training mode and in testing mode fewer number of training days).

2.6.2.3 Comparison of run-time

Consider the computational complexity of testing during sparse coding (and its variants). The expression is given in (2.98), repeated here for the sake of convenience.

$$\min_{Z_i} \left\| X - \begin{bmatrix} D_1 \dots D_N \end{bmatrix} \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_1 + \lambda \left\| \begin{bmatrix} Z_1 \\ \dots \\ Z_N \end{bmatrix} \right\|_1, \tag{2.103}$$

This is an l_1-norm minimization problem. This needs to be solved iteratively and the usual complexity being a perturbed linear programming problem is O(n^3). The same time complexity applies for other discSC, PED and DSC.

The proposed method on the other hand requires solving (2.98) and (2.100), repeated here for the sake of convenience.

$$T_i X_i = Z_i \tag{2.104}$$

$$X_i = (T_i^T T_i)^{-1} T_i^T Z_i \tag{2.105}$$

Both of them are simple matrix products (the pseudo-inverse can be pre-

computed since it only depends on the transform learnt during the training stage). The complexity of this is O(n^2) - in optimization it is usually not possible to solve a problem any more efficiently. Thus, in theory the proposed method is significantly faster than the sparse coding based techniques.

In terms of computational complexity during training, we need to solve two problems iteratively - (2.89) and (2.90). The cost of solving the sparse coding problem is O(n^2). The cost of solving the transform update is O(n^3) since it is dominated by the singular value decomposition. This cost is at par with the cost of dictionary learning. In there the complexity of updating the dictionary is O(n^2) since it has a closed form update via the pseudoinverse. The cost of updating the sparse codes is O(n^3) since it needs to be solved iteratively via l_1-minimization. The variants discSC and PED have the same complexity. The cost for solving the DSC problem (during training) increases linearly in the number of layers.

The experiments have been carried out on a desktop PC running 64 bit Windows 10. It has an i7 CPU clocked at 3.1 GHz. The RAM size is 16 GB. The testing and training run-times are shown in Tables 2.6 and 2.7, respectively. These results are shown for the usual protocols, i.e. 4:1 testing mode for REDD and 3:1 for Pecan Street.

The results corroborate the theory. Note that discSC is slightly slower than DSC because the overall dictionary size (after training) for deep sparse coding is smaller than that of discriminative sparse coding. The proposed method is

Table 2.6: Testing times in seconds

Method	REDD	Pecan Street
FHMM	3.1	50.3
DiscSC	3.5	125.6
PED	4.2	198.4
DSC	2.8	100.9
Proposed	0.1	4.7

Table 2.7: Training times in seconds

Method	REDD	Pecan Street
FHMM	39.7	30.2
DiscSC	25.1	14.8
PED	29.6	17.7
DSC	43.9	25.3
Proposed	20.4	12.6

about an order of magnitude faster than sparse coding based methods; this is expected from theory. In terms of training times, we see that the proposed method is comparable to other techniques. Is it slightly faster than the synthesis sparse coding techniques (although they have the same computational complexity) because it converges faster.

2.6.3 Summary

This paper proposes a new energy disaggregation/NILM technique. The formulation is based on the transform (analysis equivalent of dictionary) learning formulation. The main advantage of the analysis formulation (as opposed to the synthesis sparse coding/dictionary learning) is that it has been seen to be less prone to overfitting and hence can learn from fewer samples. We tested this

capability experimentally in this paper; it has been seen that when the training volume is limited (practical scenarios), the proposed method outperforms the state-of-the-art. With only 20% training data, it can supersede results from standard approaches like FHMM (trained on 60% training data) and state-of-the-art approaches like deep sparse coding (trained on 80% training data).

This has implications on cost. The proposed technique can potentially reduce cost of instrumentation during the training phase. This allows the utilities to bring more customers under the umbrella of NILM. The other additional advantage of analysis transform learning is operational speed. Since we are working on 10 min interval meter readings, this may not be of much importance right now - all techniques can disaggregate in this large time interval but for faster sampling, the operational speed would become important. In such a scenario, this method would excel over others.

The proposed method excels over others at low training volumes. The advantages have been clearly pointed out. However, in cases where the training data is large, it loses out to deep techniques. This is seen across all domains of applied machine learning. We believe the only way to improve results for larger volumes is to propose a deeper architecture based on transform learning formulation. Some initial work on this topic has been done [101].

Chapter 3

Load Forecasting

3.1 Introduction

In today's scenario, energy demand forecasting has become one of the leading sector of research. There is a great need to accurately forecast the load and energy requirements for the better management of the utility companies. Short-Term Load Forecasting (STLF) in particular has become more important since the rise of the competitive energy market.

A poor prediction can lead to under or over estimation of load thereby resulting in higher operating costs. Thus an accurate load forecast method can reduce the operating costs, keep power markets efficient, boost customer satisfaction, achieve energy savings and increase revenue. Load forecasting is also required for budget planning, maintenance scheduling and fuel management. Having said that, an accurate forecast is difficult due to the following reasons; 1) the electric load time series is highly complex and nonlinear, 2) several external

factors that can have significant impact on the daily load curve [102].

In the deregulated economy, market participants use load forecasting to manage their cost and strategies. Load forecasting can be divided into three categories: short-term load forecasts (STLF) which is usually from one hour to one week, medium-term forecasts which is from one week to one year and long-term forecasts which is more than a year. The forecasts for different time horizons are significant for different operations in the utility company. Long-term forecasts are needed for capacity planning and maintenance scheduling, medium-term demand forecasts are required for power system operation and planning and the short-term predictions are used for control and scheduling of power. Short-term forecasts are also required by transmission companies when a self-dispatching market is in operation.

Several utility companies have adopted methods for forecasting the power load. In these methods, models are formulated based on the relationship between load power and factors influencing load power [103]. The factors that affect the system load behavior can be categorized as:

- Weather: This is the most crucial factor in the changing load behavior of the system. It includes temperature, humidity, precipitation, wind speed, etc. The change in these factors directly leads to the change in the usage patterns of the appliances such as air conditioners, heaters, coolers, etc. The intraday temperatures have significant impact on the load patterns, hence they are selected as the independent variables in the forecasting. An-

other factor that majorly affects the load trend is humidity. For instance, people in the environment of 35 °C with relative humidity 70% will feel much hotter than being in the environment of 37 °C with relative humidity 50%.

- Time: Time factor influences load at different periods of the day, holidays, weekdays/ weekends and seasons of the year. The load variation with time can reflect the lifestyle of the people, that is their work schedule, sleeping pattern, leisure time, etc.
- Economy: In the deregulated market, economy factors such as price of electricity, load management policy have a significant impact on the system load growth/ decline trend. especially in a deregulated market
- Random disturbances: The start-up and shutdown of the large loads such as steel mill, wind tunnels will lead to an impulse in the load curve. The other anomalous days/ events events which are known in prior but whose effect on load is uncertain, also fall in to the category of random disturbance.

The methods for solving load forecasting problems use either statistical techniques or artificial intelligence approach [104]. A review on variants of artificial neural network for the purpose of short-term load forecasting is done here [105]. A hybrid approach of using SVM with ANN was used by [106] to predict the 24-hour ahead load. They had used an adaptive combiner that would combine the weighted results from both the techniques.

Very few works have been done on using nonlinear methods for performing

short-term load forecasts. Here, the problem of applying Kalman filter to non-linear systems is considered. It is extremely important to perform estimation in nonlinear systems as almost all real world systems involve nonlinearities. Such nonlinear practical systems range from target tracking to vehicle navigation, from chemical process plant control to dialysis machines [107]. Authors of [108] found that Holt-Winters exponential smoothing method outperforms the rest (ARMA, periodic AR modelling, method based on PCA) when applied on 10 days of intraday electricity demand data from 10 European countries.

3.1.1 Literature Review

The importance of electrical load forecasting is well known. The issue has gained even more significance with the advent of smartgrids, microgrids and smart buildings. An excellent review on this topic can be found in [109]. While the aforesaid review is of technical nature, there are other review articles on the topic of demand response delving into the financial consequences [110, 111]. This work addresses the technical problem of forecasting demand. The financial aspects of the problem will not be discussed; the interested reader may peruse the aforesaid review articles.

There are different ways to classify load forecasting techniques. One way is on the basis of the forecasting horizon. One can have three different groupings:

1. Very short-term load forecasting - from seconds or minutes to hours
2. Short term load forecasting - from hours to weeks

3. Medium and long-term load forecasting - from months to years

Yet another way to classify forecasting techniques is based on the aim of the forecast. One can have two classes depending on the problem:

1. Single Value (point estimate) - next hour load, next day load, next month load etc.
2. Multiple values - load profiles such as loads for all the hours for the next day, loads for all the week days a month from now etc.

However, the best way to classify load forecasting techniques for this work will be based on the nature of the techniques.

1. Linear regression
2. Dynamical model
3. Non-linear regression

In the following sub-sections, each of these broad categories will be discussed. In linear (static) regression, the output (demand) is modeled as a linear combination of inputs. However, since regression is inherently static in nature and cannot naturally handle time variation, a windowing approach needs to be employed. To overcome this issue, recursive techniques have been developed and employed in demand forecasting. However, the basic assumption still pegged on linearity. This was a simplifying assumption that led to elegant algorithms in theory but poor results in practice. To overcome the pitfalls of linearity

assumptions, non-linear models (mostly based on support vector machines and neural networks) were developed.

3.1.1.1 Linear Models

In linear regression models the output is the prediction and the input exploratory variables are usually the past load profiles over a period of time/window. Sometimes, instead of using the raw load values, their transforms are used. In general, such models are generically expressed as,

$$L(t) = \sum_i \alpha_i f_i(t) + \eta(t) \tag{3.1}$$

Here $L(t)$ is the output, f_t(t) are the basis/regression variables and η is the noise assumed to be Normally distributed.

In studies like [112, 113, 114], the basis for regression was arbitrarily defined from known functions (sine/cosine/exponential). However, such arbitrary mathematical basis was not able to capture the complexity of the process and hence yielded very poor results.

Somewhat better results are obtained when the regression basis is the raw data [115]. However, the raw data is noisy. These noisy inputs/exploratory variables were not very good at predicting the load either. This problem was addressed in [116]. Instead of using the raw values, it derived the basis using Principal Component Analysis (PCA) - the top principal components were noise free and optimally (in the least squared sense) captured the variability of the

data.

One of the most recent techniques [117] is also based on the same approach but instead of using PCA, it uses dictionary learning to derive the regression basis. The corresponding coefficients are used in the regression framework for prediction.

3.1.1.2 Dynamical Models

The problem with linear regression based techniques is that it cannot naturally handle dynamical systems. This is addressed in linear dynamical models. The general linear autoregressive moving average (ARMA) model is expressed as,

$$L(t) = \sum_{j} \alpha(j) L(t-j) + u(t) + \eta(t) \tag{3.2}$$

The load $L(t)$ at the t^{th} instant is modeled as a linear combination of previous loads and the input at the t^{th} instant $u(t)$. The input is usually the weather data. As before η(t) denotes the model error.

Early studies on load forecasting were all based on the ARMA model [118, 119]. However, these techniques lost its applications in load forecasting at the turn of 1990's with the advent of neural networks. The major disadvantage of it is that it is linear and that it is strongly based on the Gaussian process assumption. Even though, the second shortcoming has been addressed in later studies like [120], owing to the linearity assumption, these techniques produced average results.

The linear ARMA model can be expressed in the equivalent state-space form via the Hamiltonian. The state-space representation is given by:

$$\text{State Model: } x(t) = A(x(t-1)) + u(t) \tag{3.3}$$

$$\text{Observation Model: } L(t) = B(x(t)) + \eta(t) \tag{3.4}$$

Here A and B are two fixed matrices. The state-space model assumes that there is a hidden variable - the state $x(t)$ that is dynamically varying. We cannot observe the state directly, but can observe the output, which is a linear function of the state.

The linear state-space model can be solved efficiently using the celebrated Kalman filter. In the late 90's and the turn of the century several studies used Kalman filter with a combination of other techniques for load forecasting [121, 122].

The linear state-space model and the ARMA model are equivalent. But the state-space model can be easily extended to handle non-linearities. The state and the observation models in such cases are expressed as,

$$\text{State Model: } x(t) = h(x(t-1)) + u(t) \tag{3.5}$$

$$\text{Observation Model: } L(t) = g(x(t)) + \eta(t) \tag{3.6}$$

Here h(·) and g(·) are non-linear functions. Such non-linear state-space models can be efficiently solved using extended Kalman filter, Unscented Kalman filter or other non-linear variants of Kalman filter. Several studies make use of

such techniques in forecasting problems [123, 124, 125]. In section 3.2, the proposed work uses nonlinear Kalman filtering to solve the problem of short-term load forecasting.

3.1.1.3 Non-Linear Regression

Non-linear regression based techniques are based mostly on neural networks and support vector machines. A typical neural network consists of an input layer, a hidden / representation layer and an output layer. For regression problem, since the output is a single variable, there is a single node. In neural network based load prediction all the variables like past loads, weather conditions and occupancy information form the input. At the output is the actual load from a future time point. The neural network learns the non-linear relationship between the input and the output. Such neural network based techniques were in vogue in the 1990's [125, 126, 127]; a review of all such techniques can be found in a paper from 2001 [128].

In 2000's the paradigm of support vector machine gained popularity in almost all machine learning tasks. Basically, it is a variant of the kernel trick used in regression. But instead of using the kernel from the raw data as the basis, the support vectors are used instead. Many papers since 2000 have been published on support vector regression based load forecasting [129, 130, 131]. In recent times, recurrent neural networks (RNN) have been used for load forecasting. It models dynamical system by feeding back the output from the hidden / representation layer as the input. The inputs to the RNN are usually the same as

that of a neural network. A variant of RNN called echo state network (ESN) has been used for this task [132] but more recent works use another variant - the long short term memory network (LSTM) [133, 134]. ESNs take care of the vanishing gradient problem associated with the solution of RNN via back-propagation through time. For overview of RNN based load prediction, one can peruse [135].

Today, the state-of-the-art techniques are based on deep neural networks. Models like stacked autoencoders and variants of recursive neural networks are being used for demand forecasting. In sec 3.3, an alternate model for forecasting based on the deep dictionary learning approach [136] is proposed. It has been used in the past for addressing problems like energy disaggregation [53, 137]. This would be the first time, it will be used for solving demand forecasting via an in-built regression framework. The proposed work is a deeper extension and generalization of the sparse coding based forecasting approach proposed in [117]. In there a single layer of dictionary is learnt, the coefficients of which acts as basis for a linear regression based forecasting framework. It improves upon the prior work in two ways. First, deep dictionary learning is invoked, which is a deeper non-linear extension of shallow dictionary learning used in [117]. Second, the in-built regression is used into the deep dictionary learning process. Such jointly learnt formulation is known to yield better results than piecemeal approaches such as [117].

The work proposed in section 3.2 & 3.3 focuses on load forecasting at the building level. It is an emerging application area and many recent studies are

being published on this. Most prior studies were based on grid level forecasting; this was a much easier problem. The fluctuations at the building level gets smoothed out at the grid level rendering highly accurate forecasting a relatively simple task. Each building being different, forecasting at the building level is a challenging task.

3.1.1.4 Miscellaneous Techniques

The techniques discussed so far are generic in nature. They can be applied from the grid level to the building level. However, for buildings, several recent papers [138, 139] proposed a more intelligent way to improve prediction accuracy. Based on the past readings homes are clustered together. And the trends of other similar homes are used to predict the demand.

Another technique called semi-parametric additive models are being used for predicting loads at the national grid scale [140, 141]. These statistical techniques have been successfully applied for analyzing data from France, USA and Australia. Even though most recent studies focus on improvement on point estimates using learning based models, Bayesian techniques have been also used in the past. A review on the same can be obtained from [142].Various other techniques can be found in [109]. Even though it is a bit dated but it covers all the conventional techniques. This review has covered all broad areas but it is far from being encyclopedic.

3.2 Proposed STLF using nonlinear Kalman filtering algorithms

Kalman filters give optimal estimates of parameters of interest from indirect, inaccurate and uncertain observations. They are recursive in nature, that is they compute the best estimate of state and covariance by updating the previous estimates with new measurements. The dynamics of Kalman filters is governed by Markov process. They are widely used in forecasting applications like stock price prediction, navigation, signal processing, etc.

In this work, nonlinear versions of standard kalman filter are used for 24-hour-ahead load prediction of the residential houses. The nonlinear extensions of Kalman filters are Extended Kalman filter (EKF) and Unscented Kalman filter (UKF).

3.2.1 Mathematical Formulation of EKF & UKF

EKFs were developed for nonlinear discrete-time processes. It gives an approximation of the optimal estimate. The nonlinearities of the system's dynamics are approximated by the linearized version of the nonlinear system model around the last state estimate. For this approximation to be valid, this linearization (using first order Taylor series) should be a good approximation of the nonlinear model in all the uncertainty domain associated with the state estimate. On the other hand, UKFs are the extension of EKFs. They address the approximation issues of the EKFs. EKFs face difficulties from its use of linearization such as implementation issues, difficulty with tuning, reliability issues, etc. To over-

come these issues, unscented transformation (UT) was introduced to propagate mean and covariance information through nonlinear transformations.

- In the extended Kalman filter, the state transition and observation models don't need to be linear functions of the state but may instead be differentiable functions.
- System equations are given as,

$$State\ equation : x_{k+1} = f(x_k, u_k) + w_k,$$
$$Output\ equation : y_{k+1} = h(x_{k+1}) + v_k.$$

  where w_k and v_k are the process and observation noises which are both assumed to be zero mean multivariate Gaussian noises with covariance Q_k and R_k respectively. u_k is the control vector.

- The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed.
- At each time step, the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

- The Predict equations are,

$$\textbf{Predicted state estimate} : \hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k),$$
$$\textbf{Predicted covariance estimate} : \hat{\boldsymbol{P}}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^{\top} + \boldsymbol{Q}_k.$$

- Update equations are,

$$\textbf{Innovation or measurement residual} : \tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1}),$$
$$\textbf{Innovation (or residual) covariance} : \boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top} + \boldsymbol{R}_k,$$
$$\textbf{Kalman gain} : \boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top}\boldsymbol{S}_k^{-1},$$
$$\textbf{Updated state estimate} : \boldsymbol{x}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k,$$
$$\textbf{Updated covariance estimate} : \boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\hat{\boldsymbol{P}}_{k|k-1},$$

where the state transition and observation matrices are defined to be the following Jacobians,

$$\boldsymbol{F}_{k-1} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k},$$
$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}.$$

3.2.2 Experimental Setup

1. **Dataset**: The following datasets were used in our work. Further details of these datasets can be found in Section 4.2.3.1.

$Houses$	$Missing(\%)$
House 1	37.8
House 2	52.7
House 3	43.4
House 4	46.9
House 5	80.85
House 6	12

Table 3.1: Missing data in REDD

- Energy dataset: The dataset used in our work is the Reference Energy Disaggregation Dataset (REDD) [19]. This dataset is anonymously collected from greater Boston area in US.
- Weather dataset: The weather dataset used here is retrieved from the weather underground website[1].

2. **Data Pre-processing**: For the purpose of the research, the data was preprocessed using the following steps:

   - Missing data: This dataset contains a lot of missing data. The missing values, (NaNs) were replaced by the previous non-NaN values. The percentage of missing data for different houses in REDD is given in table 3.1.

     Since house 5 has lot of missing values, it is not included in our experiments.

   - Inconsistent data : In case of weather data, the format of the data was inconsistent with the type of the attribute used.
   - Data aggregation: Aggregation of data from both the datasets, that is previous load values and weather data (Temperature and Wind Speed)

[1] https://www.wunderground.com/

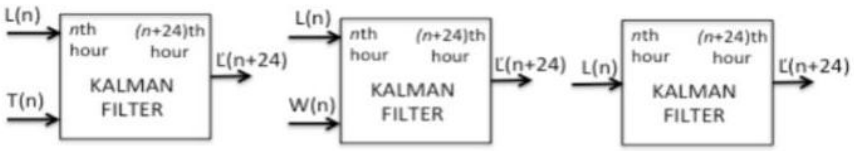


Figure 3.1: Input/Output cases of nonlinear Kalman filter

was performed at the hour-level interval using median filters.

3. **Input/Output of the model**: The input to the model is previous days' load (L), temperature (T) and wind speed (W). The output of the model is the one-day-ahead load forecast.

The first experiment is to predict the 24-hours ahead load forecasting of the residential houses using the standard linear Kalman filter. The baseline method used is echo state network (ESN). Using both the techniques, (n+24)th hour ahead estimate is recorded at every nth hour. The results recorded in this setup used all three inputs; past load, temperature and wind speed data. The mean absolute percentage error (MAPE) calculated for all the houses using linear KF and ESN are shown in table 3.4. The second experiment is to perform 24-hour ahead load prediction using nonlinear Kalman filtering algorithms, EKF and UKF. In this experiment, the estimates for three different input-output cases are recorded as shown in figure 3.1. For each hour, the results from the three cases are noted using both the techniques (EKF & UKF). The number of iterations in this experiment was fixed to 50.

3.2.3 Results

The results in table 3.2 show the root mean square error (RMSE) on the 5 houses of the REDD using Extended Kalman filter and Unscented kalman filter approach subscripted as (E) and (U) respectively. The experiment was performed for 3 periods of the day, **M**orning (9-10 am), **E**vening (5-6 pm) and **N**ight (10-11 pm). Each of these session is further divided into three categories. The first one is where historical load and temperature data is used which is represented by the subscript letter (L) and (T) respectively. In second type, the past load data and the wind speed (W) values are used and finally in the third category only past load data is used. No weather input is taken in the third category.

- In Table 3.4, linear Kalman filter clearly performs much better than echo state network. The performance of the Kalman filter can be attributed to the fact that each iteration tries to minimize the mean square error and move towards the estimate with higher certainty. In case of ESN, the recurrent neural network is less generalized as it easily over fits the training data.
- The relative variation in the error values of KF is much lesser as compared to ESN. This pattern can be resulted due to the sudden spikes/ surges in the signal, making the prediction using ESN change abruptly. Figure 3.2a) and 3.2b) show the comparison of MAPE for house 2 using ESN vs. KF and EKF vs. UKF.
- The model performs better when exogenous inputs are taken into consideration during prediction. It can also be observed that between the two non-

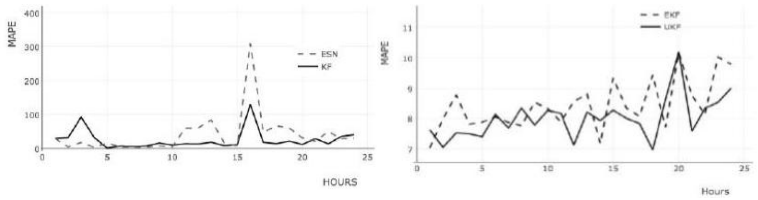


Figure 3.2: MAPE of house 2 using a) ESN & KF and b) EKF & UKF

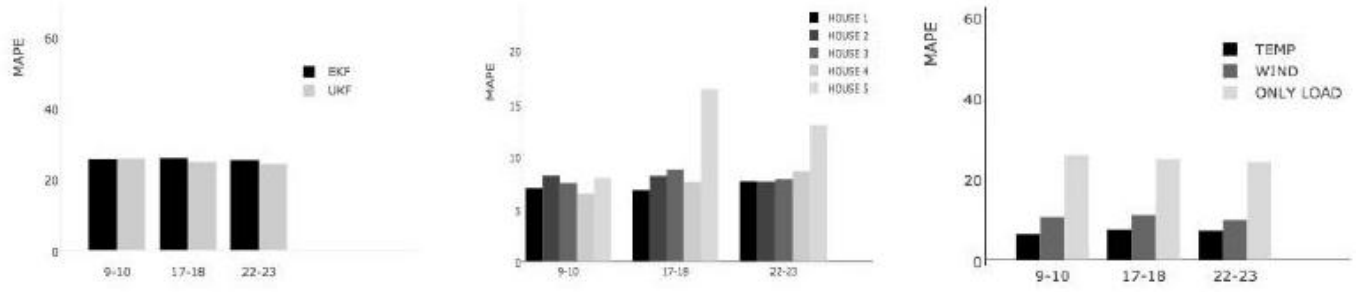


Figure 3.3: a) MAPE for load only input case b) MAPE using EKF on different houses c) Different input cases on house 3

linear techniques (EKF and UKF) used, UKF performs slightly better in terms of accuracy. In figure 3.3a), the bar chart represents the MAPE using EKF and UKF (only load input case) for different periods of the day. The comparison between different houses of REDD is shown in figure 3.3b). The resulting graph depicts the MAPE using EKF on 5 different houses. Figure 3.3c) gives a better comparison between different input cases of the nonlinear Kalman filter. It can be seen that using past load with temperature gives better performance. The bar chart shown for the morning and evening period gives the performance of the UKF and EKF respectively.

3.2.4 Summary

In the dataset [19], readings were collected from the greater Boston area, therefore temperature and wind speed were chosen as the exogenous inputs. It can

Table 3.2: Load forecasting results (in RMSE)

Houses	Morning			Evening			Night		
	M_T	M_W	M_L	E_T	E_W	E_L	N_T	N_W	N_L
$H1_E$	6.1	6.95	7.29	6.11	7.73	7.14	5.91	6.99	7.28
$H1_U$	6.3	6.58	7.53	6.18	7.6	7.29	6.01	6.85	7.3
$H2_E$	5.7	7.24	7.45	6.07	7.78	7.29	5.94	6.9	6.78
$H2_U$	5.54	6.49	7.04	6.05	7.52	6.74	5.97	6.89	6.79
$H3_E$	5.94	6.79	6.9	6.02	6.9	7.19	5.89	6.78	7.13
$H3_U$	5.99	6.68	7.11	6.16	6.81	7.08	6.05	6.6	6.84
$H4_E$	6.04	7.1	7.24	6.23	7.35	7.06	6.29	6.55	7.13
$H4_U$	5.93	6.76	7.16	6.41	7.34	7.12	6.71	6.73	7.03
$H6_E$	6.18	5.99	7.62	7.12	6.73	7.37	6.81	6.29	7.51
$H6_U$	6.36	6.14	7.74	7.09	6.36	7.24	6.57	6.32	7.22

Table 3.3: House-wise 24-hours ahead load forecasting (in MAPE)

Houses	ESN	KF
H1	178.6	49.17
H2	39.7261	24.91
H3	144.29	26.97
H4	215.84	68.34
H6	607.96	215.79

be seen that both the inputs in conjunction with the past load make the model perform better than otherwise. Therefore, careful selection of parameters is of prime importance in case of short-term load forecasting. The forecasting accuracy is a combination of good data, good process and a good model. Also, when linear and the nonlinear techniques were compared, then nonlinear techniques performed better in terms of accuracy though the variance was higher.

Table 3.4: 24-hours ahead load forecasting using ESN & KF (in MAPE)

Hours	$H1_{ESN}$	$H1_{KF}$	$H2_{ESN}$	$H2_{KF}$	$H3_{ESN}$	$H3_{KF}$	$H4_{ESN}$	$H4_{KF}$	$H6_{ESN}$	$H6_{KF}$
1	75.6	73.9	28.8	28.6	632.5	26.3	297.16	38.9	181.0	3583.1
2	106.1	74.7	3.9	31.1	518.6	2.7	216.2	3.5	262.8	820.4
3	46.7	11.5	17.2	91.8	200.5	0.18	256.6	7.7	368.7	16.5
4	162.0	48.1	2.7	31.5	268.8	13.5	197.0	64.9	2016.9	24.1
5	191.2	47.7	15.7	0.66	61.1	24.8	82.1	85.7	993.0	33.7
6	50.0	46.5	4.04	6.7	52.5	26.7	64.2	90.2	860.2	23.5
7	243.7	46.4	2.9	5.3	45.5	26.6	63.8	89.3	860.1	31.8
8	228.6	46.2	3.6	6.7	45.5	27.1	71.6	90.8	5006.7	32.9
9	70.2	43.9	6.6	15.3	43.9	26.6	127.7	90.7	752.4	106.4
10	69.3	46.1	4.6	9.8	56.2	26.2	42.8	90.7	139.2	103.8
11	113.7	44.1	58.9	13.3	47.3	27.9	94.8	90.7	18.9	32.2
12	152.8	83.7	59.9	12.6	35.9	46.2	96.9	88.7	375.1	26.5
13	86.2	63.8	83.1	17.8	58.7	41.8	50.5	83.9	95.6	34.0
14	365.1	18.7	15.7	7.7	68.9	40.7	80.8	63.6	261.6	31.3
15	297.9	11.9	6.9	10.8	56.0	40.3	245.5	66.0	260.9	29.5
16	149.6	16.9	308.7	128.6	62.6	40.3	339.6	69.5	109.1	32.9
17	325.9	18.2	47.1	17.6	51.0	38.8	148.0	86.6	194.1	30.7
18	128.4	17.8	65.8	13.6	93.5	37.0	434.4	62.0	165.1	29.7
19	884.9	18.9	59.7	20.9	84.1	41.3	469.8	63.0	104.2	25.9
20	57.7	80.7	30.5	11.4	42.3	39.3	496.6	53.9	185.0	21.1
21	121.1	79.9	20.6	28.0	119.1	3.8	289.9	85.1	63.6	23.2
22	92.7	80.1	49.1	13.2	39.7	9.5	296.4	7.1	214.8	24.5
23	193.6	79.9	27.8	34.6	354.5	4.5	467.1	81.4	70.6	29.3
24	72.5	79.6	28.8	39.9	423.0	34.6	249.9	85.3	1031.4	32.2

3.3 Proposed deep dictionary learning for building level short-term forecasting

3.3.1 Proposed Approach

3.3.1.1 Synthesis deep dictionary learning

Typically one can use the raw sample values from the recent past and other inputs like weather conditions into a regression framework to predict demand at a future instant. However, such an approach is unlikely to be very accurate. This is mainly because the data will be noisy, which means that the basis for regression will be noisy; therefore the predicted output will not be accurate. In the past, PCA has been used to clean the basis for regression [116].

A recent work [117] proposed to learn a clean basis by the dictionary learning approach. Instead of using the raw inputs (X), they learn a representation (Z) and a basis (D). This is formally expressed as follows,

$$X = DZ \tag{3.7}$$

This is a synthesis formulation, since the dictionary basis D is learnt such that, it can generate/synthesize the data X from the learnt representation Z.

The representation is learnt by the K-SVD algorithm [66].

$$\min_{D,Z} \|X - DZ\|_F^2 \text{ subject to } \|Z\|_0 \leqslant \tau \tag{3.8}$$

Here the first term is the data fidelity constraint (to remove noise); the constraint

promotes sparsity in the learnt representation Z by enforcing the number of non-zero entries to be smaller than some τ.

In [117], the representation Z is used as an input for regression. Several regression approaches were evaluated, but it was found that the simple ridge regression produces good results almost always.

There is a problem with the piecemeal approach of first learning the dictionary and representation, and then using the representation into a regression framework. The representation is being learnt for optimal noise removal, it (3.8) does not know that the coefficients will be further used for regression. Therefore, such a piecemeal approach is sub-optimal. A better approach would be to learn both regression and dictionary in a joint fashion.

$$\min_{D,Z,w} \|X - DZ\|_F^2 + \lambda\|q - w^T Z\| + \gamma\|w\|_2^2 \text{ subject to } \|Z\|_0 \leqslant \tau \qquad (3.9)$$

Here q represents the target and w are the regression weights. This (3.9) would have been a more optimal formulation; where the ridge regression (2^{nd} term) is in-built into the dictionary learning framework. Here w represents the regression weights. Unfortunately the optimal formulation was not proposed in [117]. In this work, it is shown that such a joint formulation indeed improves upon the piecemeal sub-optimal solution published in [117].

This simple extension (3.9) is not the contribution of this work. Our major contribution is to learn deeper representations. Instead of learning the representation from one layer of dictionary, we learn the representation from multiple

layers of dictionaries. This follows from the developing field of deep dictionary learning (DDL) [136, 53, 137].

DDL is a deep generalization of the shallow version (3.7), where multiple layers of dictionaries are learnt from the data X. We show it for three layers.

$$X = D_1\phi(D_2\phi(D_3Z)) \tag{3.10}$$

Here D_1, D_2 and D_3 are the three layers of dictionaries and ϕ the activation function between the layers. Compared to shallow dictionary learning, DDL not only learns more abstract deeper representations but can also handle non-linearities owing to the activation function. In the deep dictionary learning framework, we can incorporate regression in a simple fashion. All the layers of dictionaries along with the final level of coefficients and the regression weights can be jointly solved by -

$$\min_{D_1,D_2,D_3,Z,w} \|X - D_1\phi(D_2\phi(D_3Z))\|_F^2 + \lambda\|q - w^TZ\|_2^2 + \gamma\|w\|_2^2 \tag{3.11}$$

Note that we have dropped the sparsity enforcing term from the formulation. Sparsity is essential when we want to choose a few from a redundant basis. In DDL, we will be reducing the number of basis in each layer. Therefore the dimensionality of the coefficients in the final layer would be already low. Thus, there is no requirement for enforcing sparsity. Forcing sparsity might hamper the results; trying to reduce the basis from an already reduced set of basis in the final layer might restrict the expressibility.

The formulation (3.11) is specific to point forecasts, i.e. forecasting the total load for the next day or for next week. One might be interested in knowing the demand for each hour of next day, or each day of next week. In that case, one needs to predict the profile. Therefore the output q is not a vector anymore, but a matrix. Accordingly (3.11) is generalized to the following -

$$\min_{D_1,D_2,D_3,Z,W} \|X - D_1\phi(D_2\phi(D_3Z))\|_F^2 + \lambda\|Q - W^TZ\|_2^2 + \gamma\|W\|_2^2 \quad (3.12)$$

This is the most general form of regressing deep dictionary learning.

The solution approach is similar to the one taken in [137]. We follow the split Bregman technique [143]. With the proxies $Z_1 = \phi(D_2\phi(D_3Z))$ and $Z_2 = \phi(D_3Z)$, the split Bregman formulation for (3.12) becomes.

$$\begin{aligned}\min_{D_1,D_2,D_3,Z,Z_1,Z_2,W} &\|X - D_1Z_1\|_F^2 + \lambda\|Q - W^TZ\|_F^2 + \gamma\|W\|_F^2 \\ &+\mu(\|Z_1 - \phi(D_2Z_2) - B_1\|_F^2 + \|Z_2 - \phi(D_3Z) - B_2\|_F^2) \quad (3.13)\end{aligned}$$

where $B1$ and $B2$ are the Bregman relaxation variables.

The formulation (3.13) can be solved using the alternating direction method of multipliers (ADMM) approach [82]. Each of the variables are updated by solving the following subproblems.

$$P1 : \min_{D_1} \|X - D_1 Z\|_F^2 \quad (3.14)$$

$$P2 : \min_{D_2} \|Z_1 - \phi(D_2 Z_2) - B_1\|_F^2 \equiv \min_{D_2} \|\phi^{-1}(Z_1) - D_2 Z_2 - B_1\|_F^2 \quad (3.15)$$

$$P3 : \min_{D_3} \|Z_2 - \phi(D_3 Z) - B_2\|_F^2 \equiv \min_{D_3} \|\phi^{-1}(Z_2) - D_3 Z - B_2\|_F^2 \quad (3.16)$$

$$P4 : \min_{Z_1} \|X - D_1 Z_1\|_F^2 + \mu \|Z_1 - \phi(D_2 Z_2) - B_1\|_F^2 \quad (3.17)$$

$$\equiv \left\| \begin{bmatrix} X \\ \sqrt{\mu}(\phi(D_2 Z_2) + B_1) \end{bmatrix} - \begin{bmatrix} D_1 \\ \sqrt{\mu} I \end{bmatrix} Z_1 \right\|_F^2$$

$$P5 : \min_{Z_2} \|Z_1 - \phi(D_2 Z_2) - B_1\|_F^2 + \|Z_2 - \phi(D_3 Z) - B_2\|_F^2 \quad (3.18)$$

$$\equiv \left\| \begin{bmatrix} \phi^{-1}(Z_1 - B_1) \\ (\phi(D_3 Z) + B_2) \end{bmatrix} - \begin{bmatrix} D_2 \\ I \end{bmatrix} Z_2 \right\|_F^2$$

$$P6 : \min_{Z} \|Z_2 - \phi(D_3 Z) - B_2\|_F^2 + \|Q - W^T Z\|_F^2 \quad (3.19)$$

$$P7 : \min_{W} \|Q - W^T Z\|_F^2 + \|W\|_F^2 \quad (3.20)$$

All the problems are simple least square problems. Note that in P4 and P5, we are able to invert the activation function assuming they are unitary such as tanh or sigmoid. Owing to the least square form, all of them have a closed form solution. The benefit of such a solution is that, one can guarantee convergence to a local minimum following the results of [144].

The final step of the algorithm is to update the Bregman relaxation variables

via a gradient step.

$$B1 \leftarrow Z_1 - \phi(D_2 Z_2) - B_1 \tag{3.21}$$

$$B2 \leftarrow Z_2 - \phi(D_3 Z) - B_2 \tag{3.22}$$

This concludes the training. During testing we need to generate the features from the deep dictionaries and use these features as explanatory variables, i.e. multiply them with the regression weights learnt in the first layer. Given a test sample x, for generating feature z, we need solving

$$\min_z \|x - D_1\phi(D_2\phi(D_3 Z))\|_F^2 \tag{3.23}$$

The substitutions remain similar to the training phase: $z_1 = \phi(D_2\phi(D_3 z))$ and $z_2 = \phi(D_3 z)$. The split Bregman formulation takes the form,

$$\min_{z,z_1,z_2} \|x - D_1 z_1\|_2^2 + \mu(\|z_1 - \phi(D_2 z_2) - b1\|_2^2 + \|z_2 - \phi(D_3 z) - b2\|_2^2) \tag{3.24}$$

As before, ADMM is used to update each of the variables separately, leading to

the following subproblems -

$$P1 : \min_{z} \|z_2 - \phi(D_3 z) - b_2\|_2^2 \tag{3.25}$$

$$P2 : \min_{z_1} \|x - D_1 z_1\|_2^2 + \mu \|z_1 - \phi(D_2 z_2) - b_1\|_2^2 \tag{3.26}$$

$$\equiv \left\| \begin{bmatrix} x \\ \sqrt{\mu}(\phi(D_2 z_2) + b_1) \end{bmatrix} - \begin{bmatrix} D_1 \\ \sqrt{\mu} I \end{bmatrix} z_1 \right\|_2^2$$

$$P3 : \min_{z_2} \|z_1 - \phi(D_2 z_2) - b_1\|_2^2 + \|z_2 - \phi(D_3 z) - b_2\|_2^2 \tag{3.27}$$

$$\equiv \left\| \begin{bmatrix} \phi^{-1}(z_1 - b1) \\ (\phi(D_3 z) + b_2) \end{bmatrix} - \begin{bmatrix} D_2 \\ I \end{bmatrix} z_2 \right\|_2^2$$

As in the training algorithm, we have three linear least squares problems. The relaxation variables are updated via gradient descent.

$$b_1 \leftarrow z_1 - \phi(D_2 z_2) - b_1 \tag{3.28}$$

$$b_2 \leftarrow z_2 - \phi(D_3 z) - b_2 \tag{3.29}$$

Once the coefficient z is generated. The regression output is generated by multiplying it with the learnt regression weights $\hat{q} = Wz$ to give the predicted demand/load.

3.3.1.2 Analysis deep dictionary learning

Dictionary learning is a synthesis model. This has been explained before. There is an alternate analysis model that operates on the data to produce the represen-

tation [95]. This is given by -

$$TX = Z \tag{3.30}$$

Here T is the analysis basis and Z are the generated coefficients. This is also known as transform learning [89]. In recent times, the framework of deep transform learning is also being developed [145]. However, it is a less mature area compared to synthesis deep dictionary learning. For the analysis formulation, one can go deeper by analyzing the data X by multiple layers of analysis basis to produce the representation. This is given as,

$$T_3\phi(T_2\phi(T_1X)) = Z \tag{3.31}$$

As we did for the synthesis counterpart, we can incorporate ridge regression into this framework. In the most general form, it is expressed as,

$$\min_{T_1,T_2,T_3,Z,W} \|T_3\phi(T_2\phi(T_1X)) - Z\|_F^2 + \epsilon \sum_{i=1}^{3}(\|T_i\|_F^2 - \log\det\, T_i) \tag{3.32}$$
$$\lambda\|Q - WZ\|_2^2 + \gamma\|W\|_F^2$$

As before, we solve it using the split Bregman technique with the proxies $Z_2 = \phi(T_2\phi(T_1X))$ and $Z_1 = \phi(T_1X)$. After relaxing the augmented Lagrangian terms with the Bregman relaxation variables B_1 and B_2, we come

with the final formulation.

$$\min_{T_1,T_2,T_3,Z,Z_1,Z_2,W} \|T_3 Z_2 - Z\|_F^2 + \epsilon \sum_{i=1}^{3} (\|T_i\|_F^2 - \log\det\ T_i) \quad (3.33)$$

$$+\lambda \|Q - WZ\|_F^2 + \mu(\|Z_2 - \phi T_2 Z_1 - B_2\|_F^2 + \|Z_1 - \phi(T_1 X) - B_1\|)$$

The terms $(\|T_i\|_F^2 - \log\det\ T_i)$ are required for preventing the trivial solution $T_1 = T_2 = T_3 = 0$ and $Z = 0$. It also balances the scale of the transforms [95].

The approach to solve (3.33) is similar to that of (3.13). We use ADMM to segregate (3.33) into a series of subproblems, where in each of the sub-problems we update one of the variables. Each of the sub-problems have a closed form solution, including the updates for T_i's [89]. It is assumed that the activation functions are invertible so that the subproblems can be expressed as linear least square problems. This allows closed form solutions of all subproblems; this in turn results in convergence to a local minimum.

For testing, we need to solve the following in order to get the representation of the test sample x,

$$T_3 \phi(T_2 \phi(T_1 x)) = z \quad (3.34)$$

One does not need to solve any optimization problem. One simply needs to apply one transform after the other with the appropriate activations in between.

3.3.1.3 Computational Complexity

During training, both the analysis and the synthesis algorithms are iterative in nature, so we can only give estimates per iteration. The cost of the synthesis version is governed by the least square problems; the Moore-Penrose pseudo-inverse is its closed form solution. The complexity of which is $O(n^w)$ where $w < 2.37$ and is conjectured to be 2.

For the analysis version we will have two types of updates. One will be the simple pseudo-inverse for the updates of the representations Z, Z1, Z2 and W; and then there are updates for T_i's. The update for the T_i's require computing Cholesky decomposition and singular value decomposition. The cost for both are $O(n^3)$. Therefore, in terms of order of complexity the analysis version is slightly higher that of the synthesis version during training.

For testing, the synthesis version requires solving an iterative optimization problem. Therefore it is slow; having the same complexity as that of a pseudo-inverse. But the analysis version requires a few matrix vector products and can be updated in closed form. This makes the testing run-time much shorter for the analysis version compared to the synthesis one.

3.3.2 Experimental Evaluation

The first dataset (HUE) used was collected from five different residential houses located in Burnaby in British Columbia, Canada[2]. The second dataset used was

[2]HUE. http://summit.sfu.ca/item/18163

Pecan Street dataset. It was obtained via the NILMTK[3]. We did not require the appliance-level information here; we only used the aggregate power data. Of the entire dataset only few houses (used here) had continuous data for more than 3 years; only those were considered here (house ids: $1589, 3310, 3369$). The third dataset is the I-BLEND [22] dataset which was collected for 52 months at the Indraprastha Institute of Information Technology, New Delhi, India.

For all the datasets, we collected the corresponding hourly weather (temperature) information at the city level. These values (arranged as a vector) were appended with the power consumption values and served as inputs to the algorithm.

We carried two types of experiments - point and profile forecasting. For point forecasting the total demand for next day is predicted. For profile forecasting we predict the hour-wise load for the next day. For all the datasets, the first half of the data is used for training and the remaining (second) half is used for testing.

Our methods are compared against two state-of-the-art techniques - sparse coding (SC) [117] and long short-term memory (LSTM) network [134]; and one time tested benchmark - ARIMAX. For all the techniques we tuned the parameters for best results. The inputs to all the methods (benchmarks as well as ours) are exactly the same (power consumption and weather data) for fair comparison.

[3]https://github.com/nilmtk/nilmtk

Our proposed work comes in two varieties - regressing synthesis deep dictionary learning (Syn) and regressing analysis deep dictionary learning (Ana). Our parameters were tuned on the training set by cross validation via grid search.

In this work, we follow an experimental protocol similar to [121]. One half of the data (for each building) is used for training, the remaining half is used for testing. For tuning the parameters, all the algorithms used 5 fold cross validation using the training set. Evaluation is carried out in terms of three metrics - Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

In the first set of experiments, Tables 3.5, 3.6 and 3.7, we show results for point estimation, i.e. the total load for the next day. In the second set of experiments, Tables 3.8, 3.9 and 3.10, we show results for profile estimation, i.e. the hourly load for the next day.

One can see that in terms of every possible metric our methods yields the best results in general. The next best results are from SC. This is expected since SC is the shallow version of our proposed method. The time tested ARIMAX is worse than SC but better than LSTM. The state-of-the-art LSTM yields the poorest results.

One would expect that as the window size increases from 3 to 5 to 7, the results in general should improve. This is the case for the HUE dataset. This is expected since a larger window size captures greater variability in the data. However in Pecan Street data, one can see that this trend is not always followed.

Table 3.5: Point estimation comparative results on HUE

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
3	3	4.423	.039	.131	.008	.008	5.721	.031	.161	.010	.010	25.59	26.03	25.58	15.3	15.84
	5	4.500	.038	.124	.007	.008	5.890	.030	.159	.010	.010	25.93	25.26	24.63	14.5	15.23
	7	4.121	.038	.128	.007	.008	5.366	.030	.160	.010	.011	23.14	24.39	25.38	14.07	15.53
4	3	5.387	.038	.172	.007	.007	7.174	.030	.229	.010	.010	19.88	25.43	24.07	14.68	15.3
	5	5.528	.039	.173	.007	.007	7.409	.031	.231	.010	.010	20.62	27.39	24.07	14.40	15.87
	7	5.376	.038	.191	.007	.008	7.311	.031	.245	.010	.011	18.89	27.69	26.38	14.91	16.62
5	3	5.073	.044	.175	.013	.013	6.587	.038	.218	.017	.017	43.38	39.19	38.92	28.10	27.92
	5	4.967	.043	.188	.012	.012	6.441	.037	.237	.016	.016	43.77	35.30	40.98	26.27	27.03
	7	5.026	.043	.174	.013	.013	6.517	.037	.228	.016	.017	42.80	36.70	37.27	26.63	26.21
6	3	2.232	.040	.062	.009	.097	2.831	.033	.081	.012	.013	63.78	33.79	31.73	20.85	22.23
	5	2.292	.040	.063	.009	.098	2.932	.033	.080	.012	.013	68.92	33.72	32.85	21.73	23.45
	7	2.236	.040	.064	.009	.010	2.835	.033	.083	.012	.013	69.33	34.53	35.10	21.94	22.43
7	3	3.921	.041	.117	.011	.011	4.873	.034	.150	.014	.014	41.03	29.85	30.51	20.12	20.14
	5	3.869	.041	.116	.010	.010	4.761	.034	.150	.013	.013	43.88	28.84	30.33	19.19	19.36
	7	3.795	.041	.122	.010	.010	4.720	.034	.150	.013	.014	40.99	29.19	33.53	19.25	19.34

Table 3.6: Point estimation comparative results on Pecan Street

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
1589	3	9.612	.042	.364	.011	.011	12.328	.116	.495	.014	.014	20.60	27.80	26.60	15.0	15.60
	5	7.943	.043	.357	.011	.011	10.293	.118	.485	.015	.015	17.80	30.50	26.7.5	14.90	17.0
	7	6.561	.045	.398	.014	.012	8.845	.120	.537	.015	.016	15.40	33.40	29.7	15.80	20.4
3310	3	5.946	.043	.241	.010	.010	7.448	.116	.317	.013	.013	15.50	28.90	21.50	14.10	15.0
	5	6.202	.043	.269	.010	.011	7.742	.117	.352	.014	.015	16.20	30.70	24.0	14.70	17.80
	7	6.442	.044	.291	.010	.011	8.060	.118	.381	.014	.015	16.60	31.90	24.90	13.20	14.90
3367	3	7.891	.044	.342	.013	.013	9.455	.117	.422	.017	.017	25.2	27.4	29.8	17.80	18.30
	5	7.443	.044	.338	.013	.013	8.908	.117	.422	.017	.016	23.3	27.70	30.10	17.60	17.40
	7	7.384	.044	.342	.012	.013	8.802	.117	.426	.016	.016	23.0	26.60	30.60	16.30	15.70

The reason may be the non-stationarity of the data; Houses 4, 5 and 6 may be more non-stationary than the rest; in that case shorter the window better would be the results. Information from the distant past would reduce the performance.

For visual inspection, the forecasting performance of a section for House 6 of the HUE dataset is shown in Fig. 3.4. The visual plots corroborates the numerical results.

From the numerical results, it may appear that the MAPE values are relatively high. But note that it is consistent with existing literature. For example, see SC [117] and LSTM [134]; the results obtained in our paper from the corresponding techniques are better than those obtained in the said studies (albeit on different datasets).

Table 3.7: Point estimation comparative results on I-BLEND

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
Lecture	3	.095	.020	.133	.018	.020	.137	.029	.161	.028	.028	106.30	47.70	93.8	47.5	47.2
	5	.095	.020	.135	.018	.019	.144	.029	.163	.029	.028	116.80	49.30	95.40	51.6	47.0
	7	.119	.018	.102	.019	.018	.174	.029	.152	.027	.028	128.50	49.9	68.70	46.70	45.90
Acad	3	.099	.007	.154	.007	.007	.128	.010	.196	.009	.009	27.6	18.60	27.30	16.80	16.60
	5	.101	.008	.150	.007	.007	.127	.010	.180	.009	.009	27.7	19.50	26.2	17.80	18.10
	7	.091	.006	.080	.005	.007	.116	.009	.107	.007	.008	27.6	16.70	11.8	12.90	16.90
Facilities	3	.017	.003	.002	.003	.003	.020	.005	.002	.004	.004	29.4	6.60	17.70	6.40	6.60
	5	.020	.003	.002	.003	.003	.027	.005	.002	.004	.004	26.90	6.50	17.5	6.40	6.50
	7	.017	.003	.002	.003	.003	.022	.004	.002	.004	.004	28.40	6.30	18.40	6.10	6.20
Girls Hostel	3	.084	.005	.101	.004	.004	.107	.008	.132	.006	.006	62.0	11.60	24.60	9.50	9.50
	5	.089	.006	.125	.004	.005	.112	.008	.151	.006	.007	59.20	12.50	33.50	9.60	10.60
	7	.082	.006	.119	.006	.005	.106	.009	.153	.008	.007	61.8	13.50	35.10	13.30	11.90
Boys Hostel	3	.065	.003	.135	.003	.003	.083	.004	.171	.004	.004	42.50	7.10	33.10	5.90	6.0
	5	.067	.004	.205	.003	.003	.086	.005	.246	.004	.004	42.50	8.0	50.90	6.30	6.60
	7	.074	.004	.229	.004	.004	.095	.006	.259	.005	.005	41.40	8.40	50.20	8.60	7.70

Table 3.8: Profile estimation comparative results on HUE

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
3	3	12.039	.015	.245	.004	.006	19.527	.017	.177	.006	.007	74.8	63.2	76.1	56.8	56.9
	5	12.604	.015	.237	.004	.005	20.164	.017	.176	.006	.006	79.3	61.3	78.4	56.8	53.9
	7	12.363	.015	.237	.004	.005	19.867	.017	.176	.006	.006	78.0	63.7	77.2	56.5	53.4
4	3	12.513	.015	.223	.003	.005	18.419	.017	.189	.004	.006	40.8	58.7	62.3	36.3	43.5
	5	12.417	.014	.221	.003	.004	18.167	.015	.186	.004	.005	40.0	44.1	60.8	36.8	36.8
	7	12.236	.014	.220	.003	.004	18.125	.015	.185	.004	.005	39.2	44.9	60.3	37.1	36.5
5	3	11.648	.015	.257	.004	.004	22.593	.018	.172	.007	.007	275.5	50.5	87.5	47.8	43.3
	5	11.698	.016	.254	.004	.004	22.653	.018	.171	.007	.007	281.4	50.1	85.7	48.7	45.6
	7	11.687	.014	.251	.005	.004	22.626	.017	.171	.008	.007	283.2	50.1	84.2	45.4	44.6
6	3	4.278	.015	.289	.003	.003	7.908	.017	.173	.006	.005	164.2	68.5	88.8	46.9	45.2
	5	4.249	.015	.285	.003	.003	7.867	.017	.172	.005	.005	164.1	67.1	87.9	40.8	40.4
	7	4.351	.013	.284	.004	.003	7.960	.016	.171	.007	.005	177.2	68.3	87.1	40.5	39.2
7	3	7.734	.014	.237	.003	.003	14.481	.017	.192	.006	.006	198.8	37.2	62.4	42.3	39.8
	5	7.676	.014	.235	.004	.003	14.478	.016	.189	.007	.006	194.9	36.8	60.0	39.6	37.7
	7	7.550	.014	.235	.004	.003	14.350	.016	.188	.007	.006	186.3	35.6	60.7	37.9	37.0

Table 3.9: Profile estimation comparative results on Pecan Street

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
1589	3	14.51	.017	.331	.005	.005	24.05	.028	.219	.007	.006	26.7	163.9	31.2	17.0	13.5
	5	14.38	.017	.330	.005	.004	23.76	.028	.215	.006	.006	26.5	165.0	31.2	19.8	19.8
	7	14.26	.017	.330	.004	.004	23.66	.028	.214	.006	.006	26.3	164.6	29.9	19.3	19.1
3310	3	7.66	.015	.278	.003	.003	12.37	.27	.208	.005	.004	16.9	28.61	26.3	13.5	14.8
	5	7.61	.015	.276	.003	.002	12.42	.26	.208	.005	.004	16.9	28.09	26.1	15.5	19.6
	7	7.50	.015	.273	.003	.002	12.11	.26	.205	.004	.004	16.7	27.79	26.0	19.8	19.5
3367	3	9.74	.016	.306	.005	.004	16.34	.028	.231	.007	.006	25.0	35.2	28.4	11.7	12.8
	5	9.63	.016	.301	.004	.004	16.25	.028	.229	.007	.006	24.5	36.0	28.3	16.8	15.7
	7	9.62	.016	.300	.004	.004	16.22	.028	.229	.006	.005	24.5	35.8	28.2	12.1	16.9

Table 3.10: Profile estimation comparative results on I-BLEND

House ID	Window	MAE(kWh)					RMSE(kWh)					MAPE(%)				
		LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana	LSTM	SC	ARIMAX	Syn	Ana
Lecture	3	.016	.005	.046	.002	.002	.048	.007	.005	.005	.004	104.4	64.3	94.1	28.8	28.2
	5	.016	.005	.050	.002	.002	.047	.007	.005	.005	.005	104.0	64.7	77.8	29.9	29.2
	7	.014	.005	.046	.002	.002	.046	.007	.004	.005	.005	104.5	64.8	71.5	30.5	30.0
Acad	3	.040	.003	.152	.001	.001	.065	.003	.034	.001	.002	49.9	31.9	30.6	15.3	14.7
	5	.037	.003	.229	.001	.001	.065	.003	.062	.002	.002	50.0	32.9	62.2	16.3	15.5
	7	.039	.003	.213	.001	.001	.065	.003	.050	.002	.002	50.5	33.3	52.4	16.9	15.9
Facilities	3	.015	.001	.006	.001	.000	.019	.001	.001	.001	.001	25.0	15.8	8.1	12.2	10.1
	5	.012	.001	.007	.001	.000	.016	.001	.001	.001	.001	24.4	16.1	8.9	12.6	10.4
	7	.012	.001	.006	.001	.000	.016	.001	.001	.001	.001	23.6	17.1	8.9	13.1	10.6
Girls Hostel	3	.050	.003	.059	.001	.001	.072	.003	.008	.002	.002	43.5	24.1	20.5	16.2	16.2
	5	.051	.003	.52	.001	.001	.072	.003	.006	.002	.002	42.5	25.2	16.5	16.4	16.5
	7	.051	.003	.51	.001	.001	.073	.003	.006	.002	.002	42.9	24.4	17.5	16.5	16.7
Boys Hostel	3	.062	.002	.085	.001	.001	.082	.002	.012	.001	.001	45.2	14.9	21.7	11.3	11.4
	5	.063	.002	.157	.001	.001	.082	.002	.035	.001	.001	44.6	16.8	49.8	11.5	11.5
	7	.062	.002	.107	.001	.001	.081	.002	.019	.001	.001	45.0	14.8	25.7	11.5	11.5

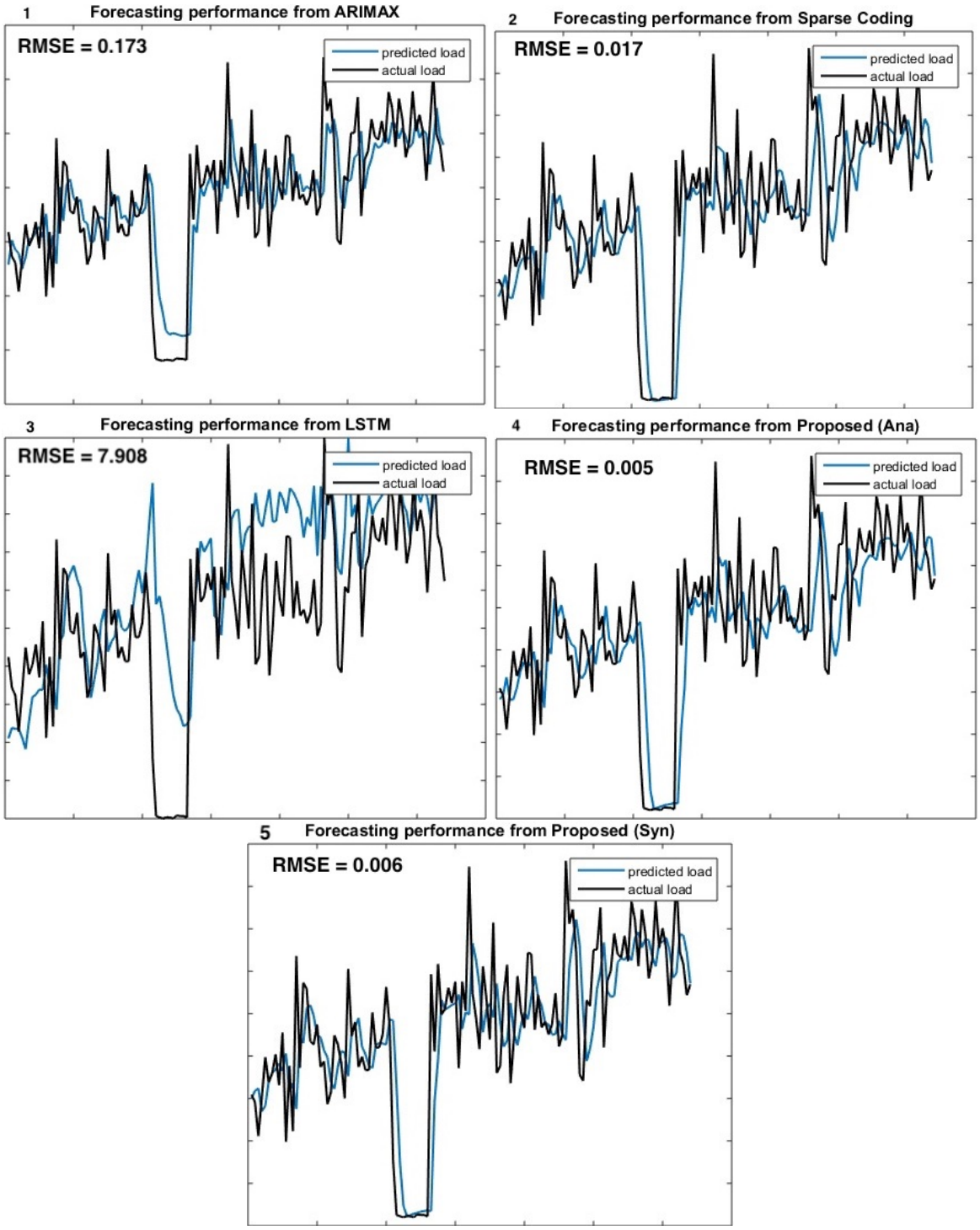


Figure 3.4: Forecasting performance (RMSE) of 1. ARIMAX, 2. SC, 3. LSTM, 4. ANA(Prop), 5. SYN(Prop) on House 6 in HUE dataset

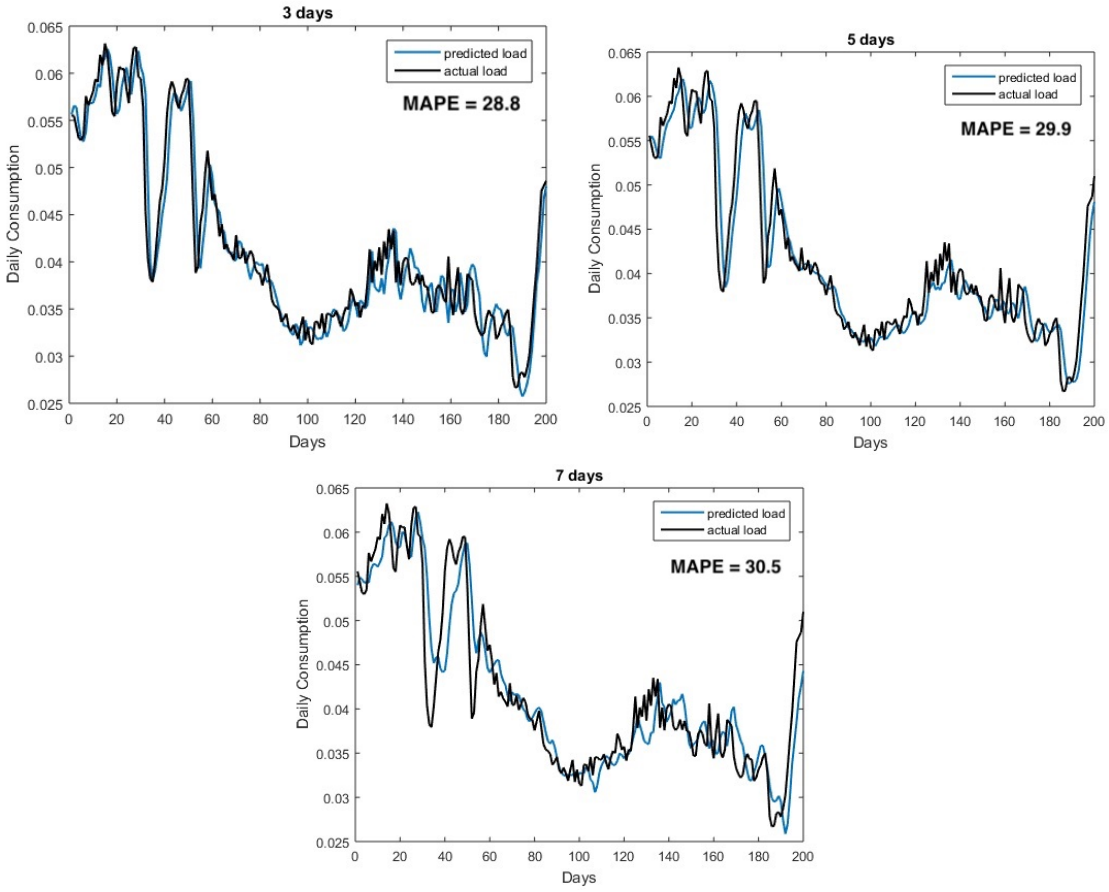


Figure 3.5: Effect on forecasting performance by varying the window size to 3,5 and 7 on I-BLEND dataset

One may have a concern regarding the poor performance of LSTM. Note that LSTMs incorporates information from samples that are in the distant past with those that are of recent past. In non-stationary data such as load forecasting, the information from the distant past should not be considered, as it hinders the forecasting results. This may be the reason behind the poor results from LSTM. It should be noted that, LSTMs were popular between 2014-2016 for time-series analysis. Owing to the said issue, in recent years companies like Facebook and Google, are increasingly using one dimensional CNNs (especially ResNet) for time series analysis. All the other techniques used here (apart from LSTM), only consider data from the recent past, ignoring 'old' data, that maybe the reason they are better at forecasting on such non-stationary data.

In order to see the effect of window size visually, we have shown the results in Fig. 3.5 for HUE dataset. These are from our proposed technique. One can see that as the window size increases, the prediction becomes smoother and is shifted from the actual values. Meaning that longer windows might not always translate to better results.

Next we show the run-times from different techniques. All the experiments have been carried out on a Macbook Pro running on 2.9 GHz dual-core Intel Core i7 processor with 8 GB DDR3 memory. The SC and the proposed techniques were implemented on MATLAB; ARIMAX and LSTM were implemented on Python.

One can see that our methods are considerably faster than all. Even though

Table 3.11: Run-times in seconds

Method	Training	Testing
LSTM	340.4	2.4
SC	14.8	0.003
ARIMAX	10.4	9.1
Proposed (Syn)	0.479	0.028
Proposed (Ana)	0.543	0.001

SC is only one layer, it uses an inefficient K singular value decomposition algorithm for its solution and hence has a higher training time compared to ours. In terms of testing, the synthesis formulations (ours and SC) are slower than the analysis formulation because they need to solve an iterative optimization problem (for the former).

3.3.2.1 Empirical Analysis of Proposed Technique

In this work we have shown the results for three layers. This is because, when we use fewer or more layers, the results deteriorate. In deep learning, the general idea is that 'deeper you go better are the results'. But the caveat is that, going deeper means more parameters to learn, and with limited training data this leads to overfitting. This is the reason, 3 layers yields the best results. In the following table (Table 3.12) we show the results on Pecan Street dataset for 1, 2, 3 and 4 layers.

In this work, we have proposed a joint formulation for learning deep dictionaries with the regression weights. In the same table, we will show what happens when they are learnt in a piecemeal fashion, i.e. when the dictionaries are learnt separately and the regression carried on the coefficients. The results

Table 3.12: Comparative MAE from multiple layers

House ID	Window	1-layer		2-layer		3-layer		4-layer	
		Joint	Greedy	Joint	Greedy	Joint	Greedy	Joint	Greedy
1589	3	.037	.042	.021	.028	.011	.024	.017	.025
	5	.038	.043	.021	.029	.011	.025	.018	.025
	7	.039	.045	.022	.029	.012	.025	.018	.025
3310	3	.039	.043	.022	.029	.010	.024	.019	.025
	5	.039	.043	.023	.029	.011	.024	.019	.026
	7	.039	.044	.023	.030	.011	.025	.019	.026
3367	3	.039	.044	.025	.031	.013	.026	.020	.026
	5	.040	.044	.025	.032	.013	.027	.021	.026
	7	.040	.044	.026	.032	.013	.027	.021	.026

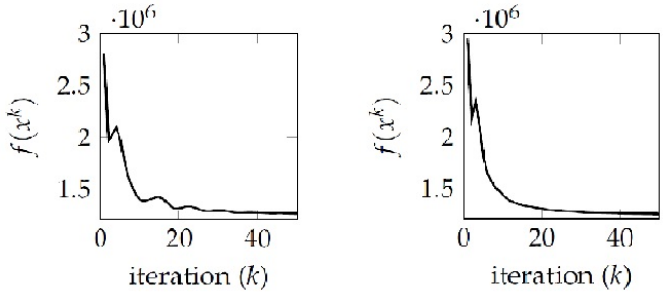


Figure 3.6: Convergence plot. Left - Synthesis; Right - Analysis

are shown on the Pecan Street dataset for point estimation. Since both the analysis and synthesis formulation yields almost the same results, we are showing results from the analysis technique only. Note that the greedy single layer formulation is the same as sparse coding (SC).

From Table 3.12 we can draw some interesting conclusions. First, our joint formulation indeed yields better results than the piecemeal one. Second, as we go deeper, the results first get better from one to three layers, but deteriorate after that. This is because of over-fitting. Third, we see that for the joint formulation, there is improvement in going deeper, but there is hardly any difference between two, three and four layers. This is largely owing to the lack of feedback between deeper to shallower layers.

Finally we show the convergence of our proposed technique. The plot of cost versus iteration number is shown in Fig. 3.6. One can see that both the synthesis and analysis algorithms converge. However, they do not converge monotonically. This is because of the variable splitting technique used here; such non-monotonic convergence phenomenon for such a class of techniques is well known.

3.3.3 Summary

The goal of this work is building level demand forecasting. Since the output (load/demand) is continuous valued it is natural to recast it as a regression problem. In this work, we have considered simple ridge regression for forecasting (used in [117]. This is the first work that incorporates regression into the deep dictionary learning framework. Even though, our simple ridge regression based approach yields better results than the state-of-the-art techniques compared against, given past literature on forecasting [146, 147] a better idea for the future might be to use quantile regression instead.

Although the technique has been developed for building level load forecasting, it can as well be applied for grid level forecasting. Another direction, we would like to explore is the possibility of domain adaptation, i.e. sharing the learnt dictionaries between datasets. This would pave way for forecasting on previously unseen datasets. Another interesting problem could be to use forecasting for anomaly detection, i.e. if the difference between the predicted and the actual is more than a certain threshold one can instantaneously flag it as an

anomaly. This is the topic of our next chapter.

Chapter 4

Anomaly detection in building energy consumption

4.1 Introduction

Commercial and residential buildings together consume a significant fraction of the total energy use. In the USA, this fraction was as high as 41% [148] while in India it was 37% [149] in 2016. Energy powers our heating and cooling systems, our ovens and stoves, lighting, and our refrigerators and freezers within these buildings. Any appliance or equipment in disrepair, while operating, can lead to high energy costs.

Studies done by the U.S. Environmental Protection Agency (EPA) suggest that buildings waste an average of 30% of the energy they consume [150]. A 2012 analysis done by Lawrence Livermore National Laboratory (LLNL) suggested the USA is only 39% energy efficient [151]. Strategies to help increase the energy efficiency in buildings are needed, especially in the case of older

buildings where appliances have a higher likelihood of failing.

One strategy, anomaly detection, is to identify appliances in a state of disrepair or used improperly. Identifying these types of anomalies can create alerts to either repair an appliance or to suggest a more optimal use. Anomaly detection, also referred to as outlier detection, deals with finding patterns in the signal that are abnormal, unexpected, or interesting.

4.1.0.1 Defining Anomalies

An anomaly can be defined in several different ways and there are many different types of anomalies. For example, an anomaly can be vacation days [152] because these are days with low total consumption as compared to typical, non-vacation days. Power utilities can define an anomaly as unexpected power consumption that results in a customer contacting customer service to complain.

The textbook definition of an outlier as defined by Harkins [153] is as follows, "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." Anomaly detection is widely used in different applications domains like credit card fraud detection in banking and finance, insurance or health care, telecommunications [154], intrusion detection in cyber security [155, 156], sensor networks [157], military surveillance, discovering criminal behavior, to name a few. Chandola et al. [158] presents a comprehensive review of anomaly detection techniques in general, whereas Sodemann et al.[159] reviews techniques

used for outlier detection in automated surveillance. In this work, our focus is on detection of anomalous power consumption in residential buildings.

The energy consumption can be labelled as anomalous or non-anomalous only when it is compared with historical data. Anomalies are broadly classified into three types: point anomalies, collective anomalies, and contextual anomalies.

Point anomaly: When an individual observation is considered anomalous with respect to the rest of the data.

Sequential or collective anomaly: When a sequence of observations are anomalous with respect to the rest of the data.

Contextual anomaly: When a observation is considered normal with respect to one context but not in another context. For example, consumption behaviour on weekdays versus weekends.

Power utility companies can define anomalies as calls into customer service where customers report their bills with unexpectedly high consumption charges. Some examples include an appliance left on by mistake, a compressor failure in a fridge, a basement renter using different appliances (e.g., plug-in heater), the purchase and usage of a new appliance, guests visiting for a long period of time (holiday season), and having the thermostat set-points too low/high as seasons change. Abnormal energy consumption pattern could also imply a malicious activity like energy theft [160].

4.1.0.2 Challenges with anomaly detection

Anomaly detection poses several different challenges that can be domain specific. For the problem of detecting anomalies in energy usage there are several such challenges including: no clear definition of normal vs abnormal, imprecise boundaries between normal and abnormal behaviour, lack of ground truth, lack of a unified metric used for performance evaluation, and evolving normal behaviour of the data [158].

One of the most significant barriers to design and test anomaly detection algorithms is the lack of labelled ground truth data. Metadata that labels the occurrences of anomalies (and their type) in datasets simply does not exist and creating such datasets is onerous and expensive. Therefore, we will present an alternative way to test the accuracy of anomaly detection algorithms using a basic statistical approach that gives us the anomaly scores at hour-level and day-level.

Additionally, from a review of anomaly detection algorithms [161, 3, 2], we have found there is no consistent way to measure the accuracy of these algorithms. In order to compare one algorithm against another there must be a standard set of metrics used to measure and report the accuracy results. In this work, we review and compare the metrics used to measure the accuracy of various anomaly detection algorithms using an automatically generated baseline.

4.2 Annotating ground truth and measuring performance accuracy

4.2.1 Literature Review

4.2.1.1 Anomaly detection in building energy consumption

Anomalies are often considered as noise or error but they may contain some important information which, on rectification, could lead to better energy utilization [162]. The research community has addressed the detection of abnormal energy consumption in several ways. An extensive review of techniques using machine learning and statistical methods for general outlier detection has been provided by [163, 164, 165]. We give a brief review of methods used specifically for identifying abnormal energy consumption in buildings.

4.2.1.2 Statistical methods

Statistical anomaly detection techniques use statistical properties of the normal activities to build norm profile and employ statistical tests to determine the deviation of the observed data from the norm profile [166]. These methods are based on the assumption of known underlying distribution of observations [153, 167]. Any observation that deviates from the model assumption is flagged as an anomaly.

Proximity based methods: These methods compute the neighbourhood for each data point using a distance metric. An analysis of the neighbourhood is done to determine whether a point is an anomaly or not. These techniques

are simple and do not make any prior assumption about the underlying data distribution.

The k-Nearest Neighbor (k-NN) method requires euclidean distances between all data instances, leading to exponential computation growth. Therefore, several different variations of k-NN were developed to improve run-time [168, 169, 3, 170]. Ramaswamy et al. [168] introduced an optimized k-NN by using techniques such as partitioning the data into cells. This helped in speeding up the processing as the distance for only the cells with data points lesser than a pre-defined threshold was computed. Wettschereck [171] used a supervised k-NN method to classify a new exemplar based on the majority classification of the nearest neighbours. The weighted voting power decreased as the distance increased.

Belalla et al.[3] proposed unsupervised clustering based anomaly detection, which flagged data points lying outside tight clusters as anomalous. They first created a low dimensional representation of each day's energy consumption and used k-NN density estimation based approach to compute anomaly scores by comparing lower dimensional representation of various days. These scores ranked days based on how anomalous they were.

Arjunan et al.[161] proposed a multiuser energy consumption monitoring and anomaly detection technique that uses an unsupervised k-medoid clustering algorithm based on Partitioning Around Mediods (PAM) and also uses neighbourhood information to adjust the anomaly scores.

Parametric methods: Statistical parametric methods assume the known underlying distribution of observations [153, 167]. They annotate as outliers those observations that deviate from model assumption. These methods allow the model to be evaluated quickly for new instances and are suitable for large datasets. Seem [2] uses a statistical approach (mean and standard deviation) to identify anomalous days. He first groups days based on energy consumption profile (weekends/weekdays) and then computes anomaly score for each day using generalized extreme studentized deviate (ESD) many-outlier procedure that was proposed by Rosner [172]. Wang [173] uses a strategy based on principal component analysis (PCA) to detect and diagnose the faults in air handling units (AHU). Fault detection using PCA is based on the intuition that anomalous readings are far away from the centre (mean/median) of the principal components of sensor data. Principal components with lower variance are preferred because, on such dimensions the normal objects are likely to be close to each other and outliers deviate from the majority. Narayanswamy et al.[174] compares the correlation, PCA and rules based methods [175] with a data mining technique proposed by them called model, cluster and compare (MCC) to detect faults in variable air volume boxes in large commercial buildings. Zhang et al.[152] proposed a regression, entropy and clustering based method to detect anomalous days for accurate demand response (DR) prediction. They define anomalous days as vacation days, when energy consumption mainly consists of automatic cycling of appliances. The regression method obtained the best test results.

Non-Parametric methods: The model of normal (non-anomalous) data is

learned from the input data rather than assuming it *apriori*. Since fewer assumptions about the data are made, these models are more flexible and autonomous. Histogram based anomaly detection [176] is a non-parametric statistical technique that involves building a histogram using the feature values in the training data. The size of the bins plays a key role in determining the accuracy of the technique. If the test instance falls in any of the bins of the histogram, it is considered normal, else anomalous. Desforges et al.[177] proposed a semi-supervised statistical technique that used kernel functions to estimate the probability density function of the normal instances. Any observation lying in the low probability area of this function is anomalous. Neural networks have also been employed by researchers to model and predict the energy consumption in a solar building [178]. Karatasou et al. [179] show how the performance of neural networks used for building's energy prediction can be improved by using some statistical procedures. Brown et al.[180] used kernel regression method to predict the power output by using the weighted average of nearby neighbourhoods. They outperformed neural networks significantly when the training data used was for 6 months or less.

4.2.1.3 Machine learning-based methods

Most commonly used machine learning methods for outlier detection employ ensemble learning. Ensemble learning methods [181] are based on the intuition that a single algorithm can not detect variety of anomalies present in the data. Initially, ensemble learning builds several homogeneous or heterogeneous

base learners and then uses combination techniques to combine their outputs. Ensemble methods for anomaly detection can be categorized as sequential or independent [182]. In the former approach, different algorithms are applied sequentially whereas in the latter approach, the results are combined from execution of different algorithms in parallel. Araya et al. [183] proposed ensemble anomaly detection (EAD) framework combining several different learners, which in turn relied on pattern and/or prediction based approaches. They evaluated a combined threshold value (ensemble threshold) depending on the optimal sensitivity and specificity. References [184, 178, 179] investigate an unsupervised autoencoder-based ensemble method in detecting anomalies in building energy data. Some hybrid approaches [185, 186, 187] have also been developed, which combine statistical, neural and machine learning approaches. Chou et al. [188] proposed a real-time prediction model, neural network auto regression (NNAR) combining time series autoregressive integrated moving average (ARIMA) and artificial neural network (ANN). They used the 2-sigma rule for anomaly detection and compare their proposed method with standard ARIMA.

4.2.1.4 Baseline

This work addresses the problem of unavailability of proper benchmarking data as well as a unified set of metrics to evaluate the performance of various outlier detection schemes. The accuracy measures to evaluate different anomaly detection approaches developed so far are not well defined. Public ground truth is not readily available; therefore, existing work uses the following ways to create

a baseline:

- manual inspection of hundreds of traces of the dataset by a domain expert,
- artificial injection of anomalies in the dataset, and
- discussion with building managers or owners to verify the anomalies.

All the above mentioned strategies either rely on third-party account or create privacy concerns and are intrusive. The manual inspection of hundreds of different traces of data seems impractical and inefficient, and is subject to human error (e.g., memory recall). Many studies also employ synthetic datasets or artificial anomalies with the objective to successfully uncover them using their proposed methods. This technique may not be able to model a realistic distribution of anomalies. Another way to have the true information about the anomalies is by asking the users or home owners to review their activities on a fixed time basis.

The authors of [189] used outlier detection schemes to uncover the injected anomalies in their work. Whereas, [161, 174, 157] asked the building managers to manually verify anomalies in their dataset. The work done by [190] discusses the anomalies uncovered (high true positive rate and low false positive) by their method through visual inspection. The authors however, claim to not be able to analyze the missed anomalies (false negatives) due to the lack of labelled data. In [3], the authors use help from building administrators to select a threshold k, such that top-k days are labelled as anomalous.

4.2.2 Proposed Methods

4.2.2.1 Nomenclature

The following symbols are used in the remainder of this work.

d day of the month

g number of groups with similar score

h hour of the day

m number of monthly data points per house

n number of houses in the dataset

s a segment in segmented or piecewise linear regression

$\mathbf{X}$ weekday or weekend data matrix with hours of the day as rows and days of the month as columns with entries $x_{h,d} \in \mathbf{X}$

$\mathbf{Z}$ z-score matrix with entries $z_{h,d} \in \mathbf{Z}$

δ user or building administrator-defined threshold

μ_h hourly mean

σ_h hourly standard deviation

$label1$ positive side of the anomaly

$label2$ negative side of the anomaly

$score$ a row vector of anomaly scores

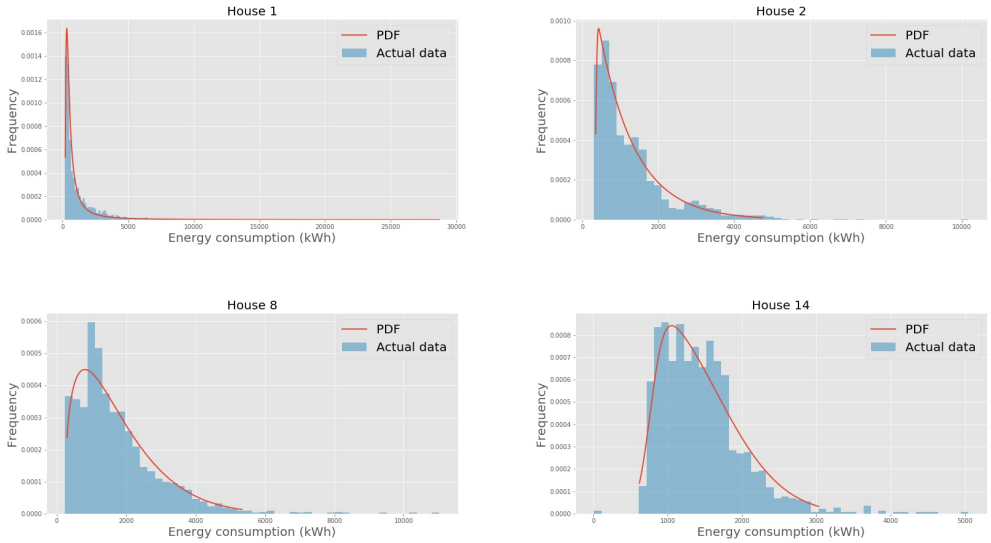


Figure 4.1: Probability density functions that best-fit four different houses in short-range dataset. The best-fit distribution for house ids starting from top-left quadrant, going in clockwise direction (1,2,14,8) are alpha, exponnorm, skewnorm and beta respectively.

pos a list of indices of anomaly scores sorted in descending order

4.2.2.2 Methodology

One concern that power utilities have is to reduce the number of customer complaint calls when they receive a high energy bill. Informing a customer in advance, proactively, can be a positive experience for customers.

For cases like these, we have devised an approach that would detect anomalies from weekly or monthly data. This approach gives power utilities a flexibility to input the desired threshold, δ such that if the z-score is above or below δ standard deviations (δ-SDs), the observation would be considered anomalous and marked as '1'. This flexibility can help the utilities to easily segregate customers based on the degree of their anomalous activities. Customers with high anomaly scores are likely to get high energy bills in the future, hence they have

a higher chance of making a complaint call.

The energy consumption (kWh) histograms of different houses in Dataport dataset were fitted with best (least sum of squared error) probability density functions. The set of probability density functions used to fit the histograms were alpha, beta, gamma, chi squared, boxcox, rayleigh, skewnorm, lognorm, loggamma, weibull, exponorm and logistic. The details of these continuous distributions can be found in the statistics package of SciPy[1]. Fig. 4.1 shows histograms of four different houses with ids 1,2,8 and 14 fitted with four different probability density functions that are alpha, exponnorm, beta and skewnorm. As can be seen in fig. 4.1, there is no particular distribution that best fits all the houses in the dataset. Therefore, a distribution function can not be generalized for all the houses.

We know that Chebyshev's inequality [191] guarantees that at least 75% of data lies within 2-SDs of the mean or in other words for a threshold δ, we can say that at most $(100/\delta^2)\%$ of values that are outside (δ-SDs) are considered as anomalous. This theoretical bound is much weaker than the actual but that is expected. We propose two approaches to generate ground truth anomaly labels based on the size of the available data: short-range and long-range.

4.2.2.3 Short-Range Data

This method is based on the z-scores. A z-score is a measure of how many standard deviations a data point is from the sample mean. The intuition behind

[1]https://docs.scipy.org/doc/scipy/reference/stats.html

Algorithm 2: Statistical method to generate ground-truth anomalies for short-range data

Require: n, m, δ
  for i = 1 : n **do**
    for j = 1 : m **do**
      $calculate\ \mu_h, \sigma_h$
      $calculate\ z_{h,d} = \frac{x^m_{h,d} - \mu_h}{\sigma_h}$ (1)
      $label1_{h,d} \leftarrow z_{h,d} > \delta$
      $label2_{h,d} \leftarrow z_{h,d} < -\delta$
      $label_{h,d} \leftarrow label1_{h,d} \mid label2_{h,d}$
      $label1_d \leftarrow \sum_h (label1_{h,d})$
      $label2_d \leftarrow \sum_h (label2_{h,d})$
      $score \leftarrow label1_d - label2_d$
      $[pos] \leftarrow Rank$ days based on $score$
      $Find$ g groups of days with same $score$,
      for k = 1 : g **do**
        $calculate\ z_d \leftarrow \sum_{h:z_{h,d}>\delta} z_{h,d}$ (2)
        $sort$ days in each group g based on z_d scores
        $update\ pos$
      end for
    end for
    $normalize\ score \leftarrow \frac{score - min}{max - min}$ (3)
  end for
  $Return\ score, pos$

a separate algorithm for a short-range data is the uncertainty in the consumption pattern and also unavailability of long-range datasets due to privacy concerns.

For this method, we separate the monthly data into groups of days with similar energy consumption profiles. For residential houses, we create two groups, weekdays and weekends. The energy consumption profile of days belonging to the same group would be similar. We then perform the following steps for each group.

1. For each data group matrix $\mathbf{X}$, whose $(d, h)^{th}$ entry represents the amount of energy consumed at h^{th} hour of the day and d^{th} day of the month, we compute the hourly mean μ_h and standard deviation σ_h across all days in

the group.

2. We then compute the z-score for each element in the matrix $\mathbf{X}$ using eq. (1) in Algorithm 2.

3. The threshold for anomaly is taken as an input by the user. The values in matrix $\mathbf{Z}$ are compared with the threshold. If the value of $|z_{h,d}| > \delta$, then the label is marked as '1' (abnormal) else '0' (normal). The obtained label values are stored in binary matrices, $label1_{h,d}$ and $label2_{h,d}$, respectively. The label for each hour of the day $label_{h,d}$ is obtained by performing a logical 'or' operation between $label1_{h,d}$ and $label2_{h,d}$.

4. For the day-level score, we sum the rows of ground truth matrix obtained at hour-level. We then subtract the scores in $label1_{h,d}$ and $label2_{h,d}$ matrices to get the net score. The value of net score determines the extent of abnormal energy consumption on a particular day. The positive net score indicates the positive side of anomaly, that is when the energy consumption is more than usual, whereas the negative score indicates the abnormally low energy consumption.

5. The annotated positive and negative label on a weekend group is shown in fig. 4.2. This figure shows energy consumption of House1 from Dataport on weekend days. The red circles indicate positive anomaly whereas the black star represents negative anomaly. As we are more interested in the positive side of the anomaly, we sort the days in the descending order of the day-level anomaly scores.

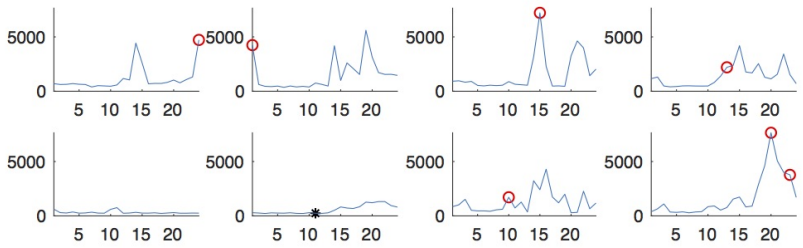


Figure 4.2: Annotated anomalies on a weekend data group in short-range data

6. We create groups of days with the same anomaly score to resolve the ranking conflict between days with the same score.

7. For each group, we compute the sum of z-score values that are greater than the desired threshold δ, as shown in eq. (2).

8. Finally, we normalize the scores using min-max normalization using eq. (3) in Algorithm 2, where the max and min values are taken from the data group matrix $\mathbf{X}$.

9. The algorithm outputs pos which represents days in decreasing order of their anomalous behaviour and $score$ which defines the extent to which these days are anomalous. A day is anomalous if it is assigned a $score$ greater than 0.

4.2.2.4 Long-Range Data

Long-range energy usage and temperature data gives a better understanding of the user consumption pattern through the annual seasons. With more data, it is easier to know the consumption trend of a user.

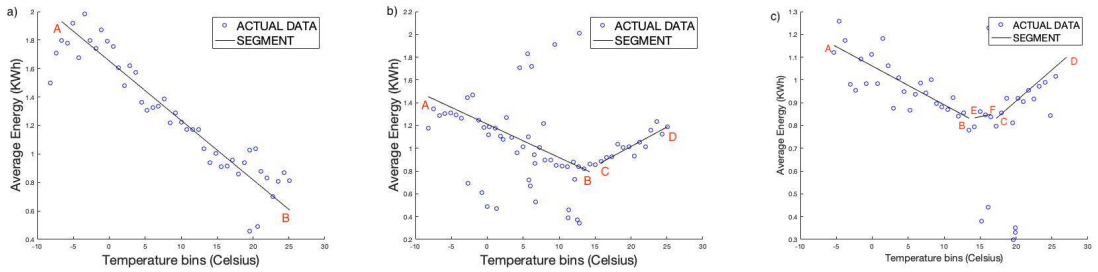


Figure 4.3: Prediction of energy usage by segmented linear regression. Three different case scenarios of energy consumption using segmented linear regression are: a) unsegmented linear regression b) segmented linear regression with one breakpoint, and c) segmented linear regression with two breakpoints. Actual energy consumption is shown in blue dots whereas the segments that best fits the data are shown in black.

For this case, we have used annual energy and temperature data from different houses. This data is sampled at hour-level. The correlation coefficient between outside temperature and energy consumption as shown in fig. 4.3a is 0.955. The high correlation between these two variables is the foundation of this approach. The graphs shown in fig. 4.3 represent three different cases of energy consumption with respect to the outside temperature. During winters, the energy consumption increases as the temperature decreases due to invariable heating needs. This is represented in fig. 4.3 by the negative slope segment 'AB'. Similarly, during summers, as the temperature increases the energy consumption also increases due to cooling loads as can be seen by the positive slope segment 'CD'. The energy used for cooling or heating of the building is referred to as temperature-sensitive usage. The energy used by computers, lights or appliances not sensitive to the outside temperature is referred to as temperature-insensitive usage. This kind of usage can be identified by a segment with a near-zero slope, 'EF'. A house could have a cooling or a heating appliance, or both, or none, therefore a single linear regression function is not adequate to cover all the cases. This is why the segmented linear regression is

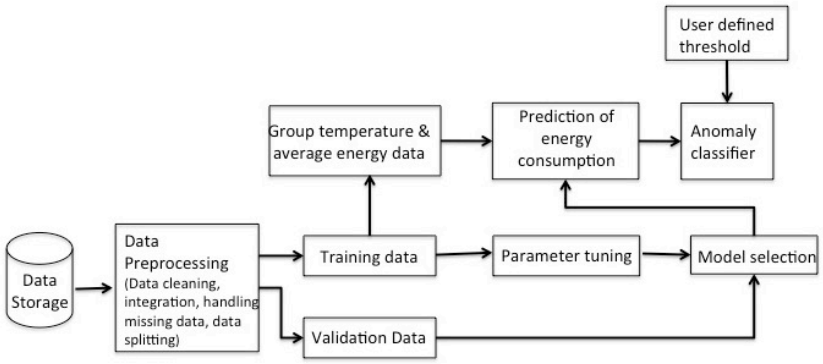


Figure 4.4: Block diagram for annotating ground truth anomalies using long-range data

necessary, as described below.

1. The first step is to prepare the dataset for training. Preprocessing involves cleaning the data, removing inconsistent and redundant timestamps, adding missing timestamps, adding missing energy consumption values and integrating hourly temperature data. The missing values are usually caused by a hardware or a software failure of the measurement device. The missing energy values were replaced by the values of previous or next year's data corresponding to the same timestamp. For cases where the previous and next year data was not available, the average of previous and next hour of the current year was used.

2. After the data is preprocessed, it is split into training and validation sets in the ratio of 9:1 respectively. A single instance or sample at h^{th} hour was selected from every 10 samples to create the validation set. The remaining samples were used for the training set.

3. Next, the training set is used for model selection or model training. Grid

search is used to train the model. The parameters in the case of pure, unsegmented linear regression are also tuned in this step. The parameters tuned include the set of breakpoints in the case of segmented regression, regression coefficients and constants. The optimal value of the breakpoint is found such that the coefficient of determination, R^2 shown in eq. (4) is maximum. In this equation, y_i refers to the observed data point, $\bar{y}$ is the mean of all the observed data points and f_i represents the predicted power consumption.

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{4}$$

4. The model is tested on the validation set. In a scenario where no heating or cooling is used, the unsegmented linear regression may perform better than the segmented one.

5. The sorted temperature values are grouped together such that each group has sufficient number of energy consumed data points. Since the frequency distribution of data points at the maximum and minimum temperature values will be minimum, we merge the groups such that each group has sufficient (in our case atleast 20) number of data points.

6. The grouped training data is partitioned based on the parameters obtained from the test on the validation set. The number of segments and the breakpoints are optimally chosen depending on the best R^2 value.

7. For each segment in the partitioned training dataset, energy consumption values are predicted using the best learned linear regression coefficients and constants values.

8. To determine the anomalous data point, we compute the z-score of the difference between the actual and predicted energy consumption as we did in step 2 of Algorithm 2. We compare the z-score with δ, as we did in step 3 of Algorithm 2 to obtain two binary column vectors, $label1_h$ and $label2_h$ representing the positive and negative side of the anomaly respectively. To obtain the final $label$, we performed a logical 'or' operation between $label1$ and $label2$ generated for each timestamp.

9. Using these binary anomaly labels for the grouped training dataset, we identify the corresponding anomalous data in the actual hourly readings and annotate the ground truth. On the grouped training dataset, figure 4.5 shows the normal or non-anomalous data (blue dots), regression model (shown by straight lines) and anomalies (red stars). We may observe annotated anomalies (red star) closer to the straight line than the normal data (blue dot) because data points are an average of energy consumption values lying in the same bin. So, it may be possible for one of the value in the averaged group has z-score of the difference between actual and predicted energy greater than a threshold whereas the rest of the values be closer to the straight line. The correlation coefficients between the average yearly energy consumption and the outside temperature corresponding to the three regression line segments are 0.6545, 0.8440 and 0.7937 respectively.

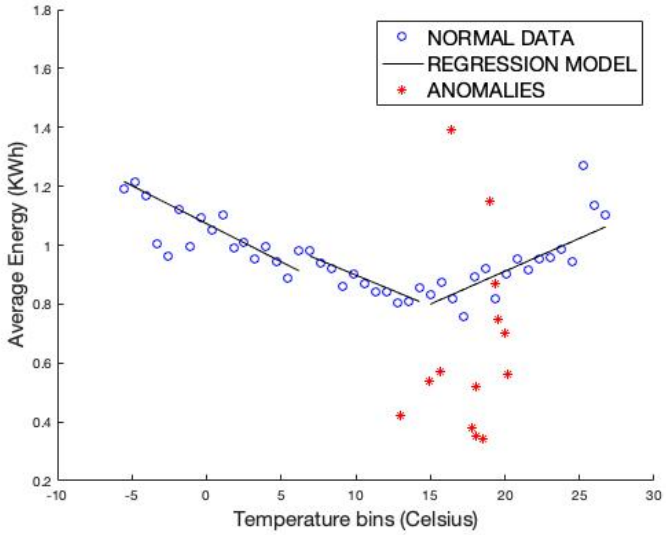


Figure 4.5: Annotation of anomalous observations in long-range data

4.2.3 Experimental Setup

4.2.3.1 Dataset

Dataport Dataset: The first dataset used is the publicly available Dataport (Pecan Street) dataset with NILMTK [85]. We are using two months (April and May) of meter-level data from nine houses. For convenience, we consider 30 days in both months. The average temperature in these two months was 19°C and 23°C, respectively. Short-range data analysis 4.2.2.3 was applied to this dataset. We have used houses with ids 1, 2, 3, 4, 5, 8, 11, 12 and 14. The house ids which were discarded due to missing data are 6, 7, 9, 10, 13. For each month, the data was grouped based on the day types, that is, days of the week with similar energy consumption profiles were grouped together. Therefore, for each month we have weekdays consisting of 22 days and weekends consisting of eight days. Aggregation of both groups of data takes place at an hour-level.

Hence, the size of the weekday dataset per month would be (24×22) and that of weekend would be (24×8).

HUE Dataset: The second dataset used is collected from different residential houses located in Burnaby in British Columbia, Canada [21]. This dataset has meter-level energy consumption values which are sampled at each hour. The data is collected over a period of three years, ranging from January 2015 to January 2018. We have used five houses from this dataset with house ids 3, 4, 5, 6, 7. However, there are more than five houses in this dataset. Hourly temperature data was included in the dataset, which we used to detect abnormal energy consumption from yearly data. The integration of weather and energy consumption data was done to find the correlation between them. Long-range data analysis 4.2.2.4 was applied to this data.

4.2.3.2 Performance metrics

Performance metrics allow us to measure how accurately a detection algorithm identifies an anomaly in the energy consumption pattern. It is important to measure how effective an approach is in the classification task of anomalous vs non-anomalous behaviour of energy signals. The notion of anomaly score is used to quantify the extent of anomalous behaviour in the energy signals. High anomaly score means high degree of anomalousness. For example, the work done by [2], uses robust statistical methods to determine if the current day's energy consumption is significantly different from the previous days' energy consumption. They use generalized extreme studentized deviate (ESD) as an

outlier identification method [172]. To quantify how far and in which direction an outlier is from the mean value of non-outlier observations, a modified z-score was used.

We have compiled a list of performance evaluation metrics that have been used to measure the performance accuracies in the majority of the anomaly detection methods. These are explained in the remainder of this section.

4.2.3.3 True Positive Rate (TPR)

This is also commonly known as sensitivity, or outlier detection rate, or recall. TPR is the proportion of correctly identified positive classes from the total possible positive conditions, that are true positives (TP) and false negatives (FN). In the context of anomaly detection, TPR measures the fraction of anomalous events identified by a given method.

$$TPR = \frac{TP}{TP + FN} \,, \tag{4.1}$$

The research done by [152, 183] have used TPR as their performance evaluation metric. In section 4.2.4, TPR is reported as the mean of all TPR values obtained from different houses.

4.2.3.4 True Negative Rate (TNR)

Also known as specificity, TNR is the proportion of correctly identified negative classes from the total possible negative conditions, that are true negative (TN)

and false positive (FP). In the context of anomaly detection, TNR measures the fraction of non anomalous events identified by a given method. References [183, 3] have used TNR to measure the accuracy of their testing. Both TPR and TNR aim to reveal how accurately a technique has identified the true nature of a given sample, that is whether it is anomalous or not.

$$TNR = \frac{TN}{TN + FP}\,, \tag{4.2}$$

4.2.3.5 False Positive Rate (FPR)

FPR refers to the rate of false alarms or fall-out, which means misclassifying some non-outliers as outliers. It has been applied as an accuracy metric in [192, 152, 183].

$$FPR = 1 - TNR\,, \tag{4.3}$$

4.2.3.6 F_1 score

F_1 score or F-measure is widely used in the field of information retrieval for measuring search, document classification and query classification performance. It indicates the retrieval effectiveness of the system and is defined as the harmonic mean of the precision defined in eq. (4.5) and recall (TPR). F_1 score is defined in eq. (4.4).

$$F_1 = \frac{2 * prec * recall}{prec + recall}, \tag{4.4}$$

Precision and Recall (TPR), on the other hand are the traditional performance metrics used to evaluate the quality of the information retrieval system [193, 194], and are also widely used to measure the performance of outlier detection schemes. Precision ($prec$) is the fraction of relevant instances among the retrieved instances as defined in eq. (4.5). High precision is when the algorithm returns more relevant results than irrelevant ones. On the other hand, recall (or TPR), as defined in eq. (4.1) is the fraction of relevant instances that have been retrieved over total relevant instances. High recall is when the algorithm returns most of the relevant results. Equation (4.6) expresses F_1 score in terms of TP, FP and FN.

$$prec = \frac{TP}{TP + FP} , \tag{4.5}$$

$$F_1 = \frac{2TP}{2TP + FN + FP} , \tag{4.6}$$

Reference [156] uses precision and recall as metrics to compare and evaluate the performance of outlier detection schemes on real-life and synthetic datasets.

4.2.3.7 Jaccard Index

Jaccard index, also referred to as intersection over union (IOU), is a metric used for comparing the similarity and diversity of sample sets. In the context of anomaly detection, this measure estimates the similarity between the two sets

of data, one obtained through the anomaly detection method and other from the ground truth anomalies.

$$Jaccard = \frac{TP}{TP + FP + FN}, \tag{4.7}$$

4.2.3.8 False Positive when detection rate is 100% (FP-100)

FP-100 is the number of false positives returned by the algorithm when the algorithm has detected all the anomalous days as given in the ground truth. It can be used to compare two algorithms, suppose if a dataset has 10 known anomalies and the rank of the 10th anomaly is 17 by algorithm 'A' and 20 by algorithm 'B', then algorithm 'A' is better than 'B' because for 100% detection rate, 'A' has only 7 false positives whereas 'B' has 10. The work by [192] have used FP-100 as a metric to evaluate the performance of their algorithm.

4.2.3.9 Area Under Curve (AUC)

The receiver operating curve (ROC) is commonly used to measure the performance of the classifier by plotting true positive rate against false positive rate. The area under this curve, AUC, defines the quality of the detector. AUC is often used to measure the performance of the algorithm [192, 3, 152, 183]. The value of $AUC = 1$ represents a perfect anomaly classifier whereas a value of AUC = 0.5 signifies the performance of the model to be no better than a random guess.

4.2.3.10 Partial Area Under Curve (pAUC)

The partial area under the curve is a performance metric defined as the area within the range of specific true positive and false positive rate. It is more suitable for comparing classifiers whose ROC curves cross [183]. For example, if amongst two anomaly classifiers A and B, let us say A has better true positive rate than B in a specific false positive rate range while classifier B performs better in a different false positive rate range, then we can identify a specific range relevant to the application to apply pAUC rather than AUC, which gives an overall combined metric.

4.2.3.11 Rank Power

Even though Precision and Recall are widely used to measure the accuracy of anomaly detection, they still lack in some respects, mainly because they do not give any preference to the ranks, that is, how anomalous is a particular sample. As proposed by [156], rank power shown in eq. (4.8) evaluates the ratio of known anomalies and anomalies returned by an algorithm along with their rankings [192].

$$RankPower(k) = \frac{l \cdot (l+1)}{2 \cdot \sum_{i=1}^{l} R_i} \tag{4.8}$$

where l is the number of outliers among top k objects. R_i is the position of the ith outlier in a rank-order list.

4.2.3.12 Implementation

Algorithms used to annotate the data observations using short and long-term approaches were implemented in MATLAB. The codes to implement these methods have been made available on GitHub[2]. The implementation of various performance metrics is also publicly available at the same site.

4.2.4 Results

We conducted performance experiments on real-world publicly available datasets. As mentioned in section 4.2.3.1, we have used a subset of Dataport dataset [85] and the HUE dataset [21] to generate labels for short-term data (weekly or monthly) and long-term energy consumption data (yearly) respectively. The ground truth labels are generated for three different thresholds, that are 1.65-SDs, 2-SDs and 2.5-SDs. As the value of threshold increases, the anomalies become sparser.

The anomaly scores obtained using anomaly detection methods [2, 3, 161] are compared with the scores generated through Algorithm 2. These methods which we refer to as 'multiuser' [161], 'hp' [3] and 'seem' [2] have been briefly discussed in section 4.3.1 of this work. Table 4.1 shows a comparison of different accuracy measures for weekdays and weekends separately, provided that the data lying outside ± 1.65-SDs is considered anomalous. Similarly, Tables 4.2 and 4.3 report the accuracies of algorithms for thresholds ± 2-SDs and ± 2.5-

[2]https://github.com/megha89/AnomalyDetection

SDs, respectively. The upward (↑) and downward (↓) arrows in tables 4.1, 4.2 and 4.3 indicate the direction of desirable performance according to that metric.

For both weekday and weekend groups, it can be observed from Tables 4.1, 4.2 and 4.3 that [161] gives the best TPR whereas [2] outputs the lowest TPR across all thresholds. One possible reason why [2] outputs the lowest TPR could be that they consider an upper bound on the number of potential outliers, o_u. The maximum number of potential outliers can be $o_u < 0.5(o - 1)$ where o is the total number of observations. We should also note that its anomaly detection rate increases as we increase the threshold for anomalous data.

Contrastingly, in case of TNR, [2] clearly outperforms the rest in correctly identifying the normal observations from abnormal ones during weekdays and weekends. Furthermore, in case of false alarms or FPR, [2] again has a very low misclassification rate in comparison to others methods. After comparing TPR, TNR and FPR values, it can be concluded that the techniques [161] and [3] classified majority of the normal data as abnormal therefore maximizing TPR and FPR but minimizing TNR. Thus, only presenting a very high TPR result can be misleading if it is not accompanied with high TNR and low FPR values. All three rates are important to make an informed decision about the classification accuracy.

The ability of an algorithm to return all known outliers with minimum number of false positives is captured by the metric FP-100. Here, [3] returns less false positives when detection rate is 100% than [161] in case of weekdays but

vice versa for weekends. We have not mentioned the results from [2] because this method does assign a score to all the observations, therefore leading to cases where known anomalies are more than the assumed potential anomalies.

The other metric, F_1 score, which is based on precision and recall, evaluates the ranking of results. From eq. (4.6), we observe that F_1 is directly proportional to TP and inversely proportional to the sum of FN and FP. Therefore, best F_1 can only be attained with high true positives and low false positives and false negatives. On comparing F_1 scores across all thresholds, we observe that when the threshold is highest, [2] attains the best F_1 score. Also, it should be noted that as the threshold value increases, the F_1 score using [2] also increases but on the other hand, this score decreases for [161].

Jaccard index as shown in eq. (4.7) is a metric similar to F-score. It is a statistic used to estimate the similarity of two sets of data. We use the Jaccard index values to compare the accuracy of different anomaly detection methods. Intuitively, it follows a similar trend as F_1 score but with lower scores.

The most commonly used metric is area under the ROC curve (AUC). After the ranked list of data is obtained from a algorithm, the user chooses a threshold, $\tau \in (0,1)$ declaring that the points above the threshold are anomalous and, the remaining normal. Each choice of value of τ gave out a certain value of true positive and false positive. On varying this threshold τ, different values of TPR (y-axis) and FPR (x-axis) were obtained, leading to a ROC curve. The values for AUC presented in Table 4.2 were calculated after considering thresholds

from 10% to 90% with a step size of 10%. It has been presented in the table 4.2 even though it has no relation with the specific threshold, $\tau = 2$ as it is the are under the ROC curve across all thresholds. The low values of AUC is due to the low range of false positives. Though the values of TPR and FPR were high but the range across all thresholds was very low leading to low AUC.

Rank power [156] is an effective metric that meets the users' satisfaction by factoring in the rank of the outliers. As shown in eq. (4.8), rank power for k objects is the ratio of known l anomalies in top k data to the ranking of those l anomalies as returned by the algorithm. In our study, we took the value of k as 3 because the number of anomalies returned by [2] in case of weekend data were at most 3. Tables 4.1, 4.2 and 4.3 show that [2] outperforms other approaches across all thresholds. From the results, we can say that the ranking of anomalies in case of [2] was more precise than the rest. We have also graphically presented the results given in the tables using overlapping bar graphs in figures 4.6 to 4.11.

From this experiment, we can conclude that [2] outperformed the rest of the techniques. If it had assigned scores to all the days, then it would have performed the best across all the reported metrics.

4.2.5 Summary

In this work, we discuss the two most common problems in detecting abnormal energy consumption in buildings. The first problem is the the lack of labelled ground truth to train supervised models, and the second is the lack of consistent

Table 4.1: Performance accuracies on weekdays and weekends on Dataport when the threshold is 1.65-SDs

	weekdays			weekends		
Metrics	seem [2]	hp [3]	multi-user [161]	seem [2]	hp [3]	multi-user [161]
TPR ↑	0.4788	0.9491	**0.9974**	0.3706	0.8679	**0.8721**
TNR ↑	**0.9889**	0.038	0.0056	**1.0000**	0.0944	0.2093
FPR ↓	**0.0111**	0.962	0.9944	**0.0000**	0.9056	0.7907
F-score ↑	0.6397	0.9	**0.9218**	0.5307	0.7795	**0.7908**
Jaccard Index ↑	0.4741	0.829	**0.8651**	0.3706	0.6558	**0.6696**
FP-100 ↓	NA	**2.7778**	**2.7778**	NA	1.5556	**1.4444**
Rank Power ↑	**0.4133**	0.1006	0.1060	**0.5660**	0.2870	0.3013

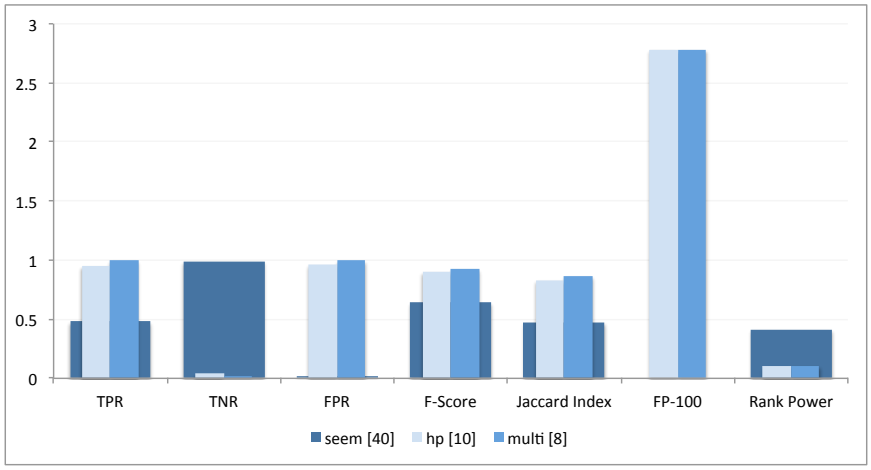


Figure 4.6: Comparison of performance accuracies on weekdays when the threshold is 1.65-SDs

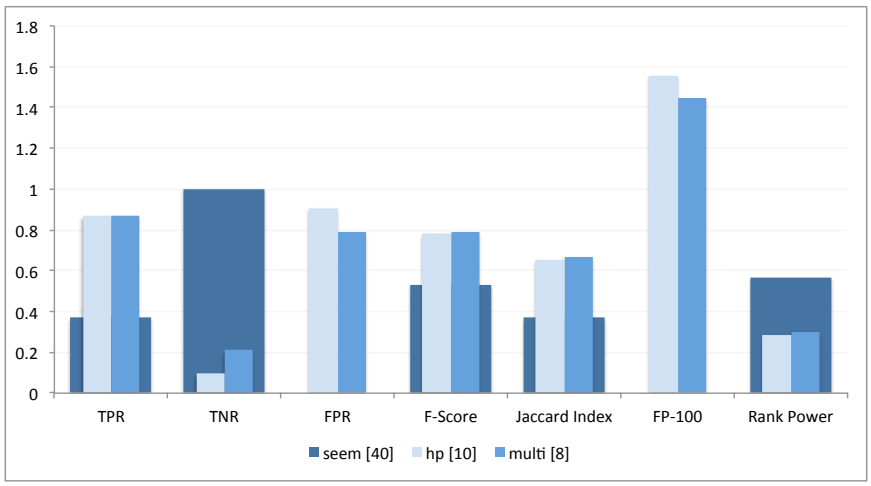


Figure 4.7: Comparison of performance accuracies on weekends when the threshold is 1.65-SDs

Table 4.2: Performance accuracies on weekdays and weekends on Dataport when the threshold is 2-SDs

	weekdays			weekends		
Metrics	seem [2]	hp [3]	multi-user [161]	seem [2]	hp [3]	multi-user [161]
TPR ↑	0.5478	0.9592	**0.9974**	0.4540	0.8721	**0.9156**
TNR ↑	**0.9239**	0.0389	0.0046	**0.8976**	0.1376	0.1857
FPR ↓	**0.0761**	0.9611	0.9954	**0.1024**	0.8624	0.8143
F-score ↑	0.6641	0.8085	**0.8208**	0.5412	0.6117	**0.6469**
Jaccard Index ↑	0.5032	0.7035	**0.7259**	0.3993	0.4702	**0.5030**
FP-100 ↓	NA	**4.8333**	5.5000	NA	2.6111	**2.2778**
AUC ↑	0.0930	**0.1233**	0.0554	0.0773	0.0870	**0.0949**
pAUC ↑	**0.0879**	0.0133	0.0013	**0.0815**	0.0365	0.0244
Rank Power ↑	**0.3724**	0.0830	0.0855	**0.4329**	0.1988	0.2456

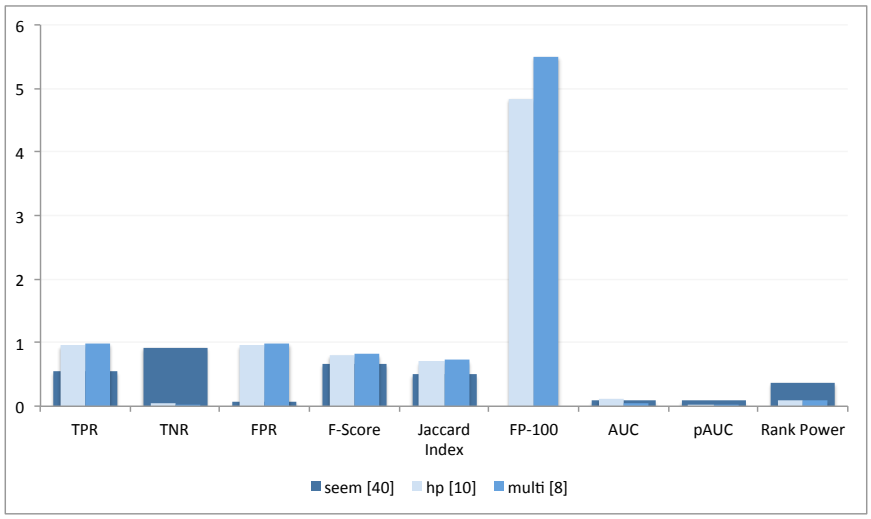


Figure 4.8: Comparison of performance accuracies on weekdays when the threshold is 2-SDs

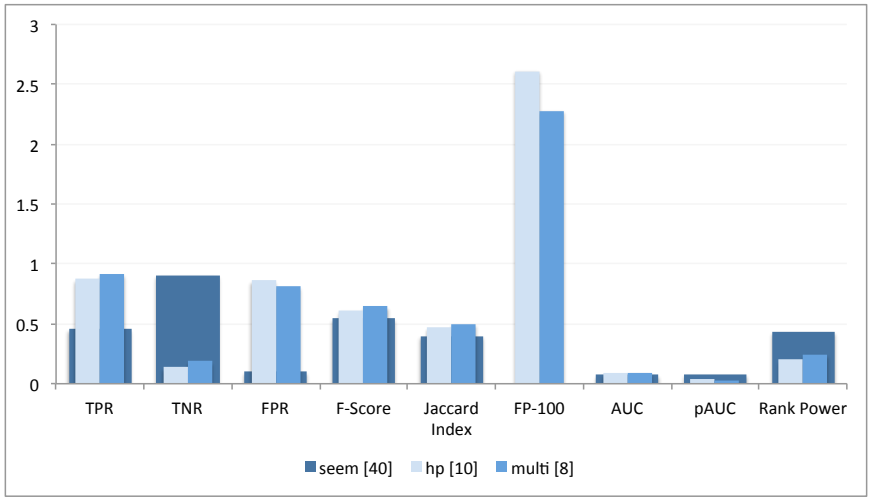


Figure 4.9: Comparison of performance accuracies on weekends when the threshold is 2-SDs

Table 4.3: Performance accuracies on weekdays and weekends on Dataport when the threshold is 2.5-SDs

	weekdays			weekends		
Metrics	seem [2]	hp [3]	multi-user [161]	seem [2]	hp [3]	multi-user [161]
TPR ↑	0.7174	0.9676	**0.9974**	0.4696	0.8721	**0.9782**
TNR ↑	**0.8377**	0.0517	0.0043	**0.8125**	0.1376	0.1821
FPR ↓	**0.1623**	0.9483	0.9957	**0.1875**	0.8624	0.8179
F-score ↑	**0.7012**	0.6131	0.6142	0.3914	**0.6117**	0.4325
Jaccard Index ↑	**0.5516**	0.4692	0.4733	0.2844	**0.4702**	0.3135
FP-100 ↓	NA	**7.6667**	8.3889	NA	2.7778	**1.333**
Rank Power ↑	**0.4133**	0.1500	0.1180	0.1446	0.1686	**0.2071**

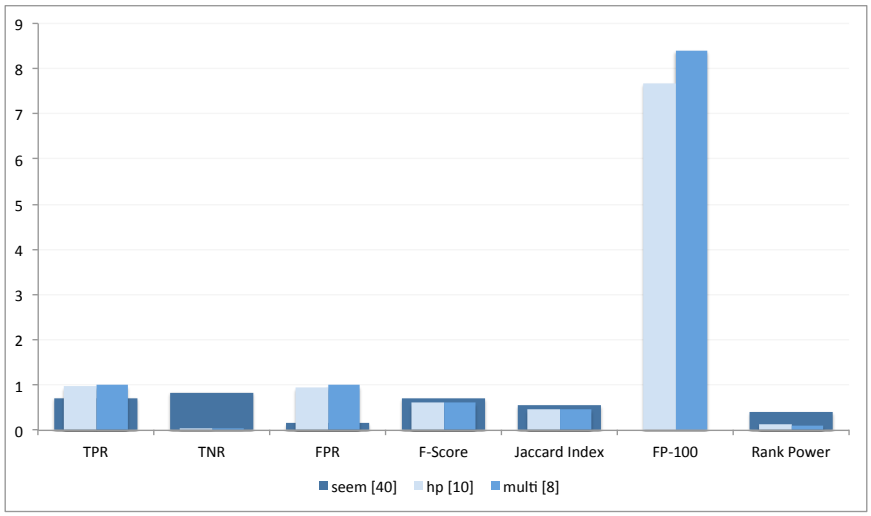


Figure 4.10: Comparison of performance accuracies on weekdays when the threshold is 2.5-SDs

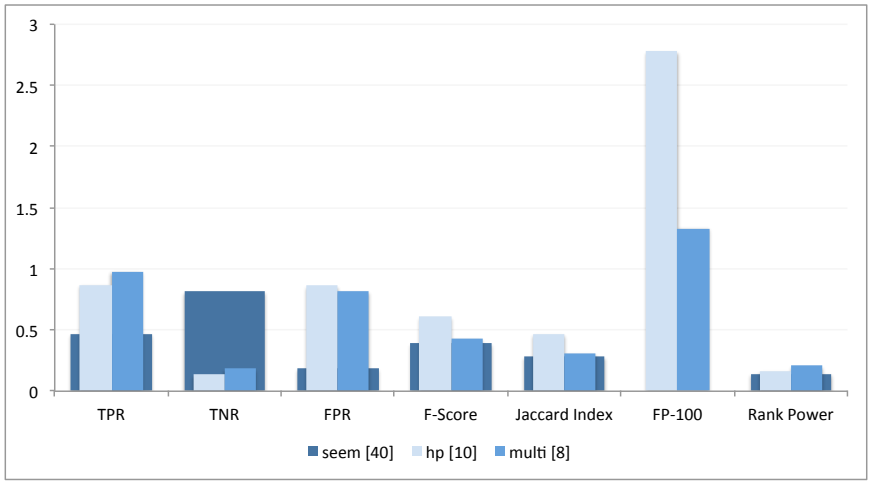


Figure 4.11: Comparison of performance accuracies on weekends when the threshold is 2.5-SDs

performance accuracy metrics.

To mitigate the first problem, we have proposed two methods to generate labelled data for abnormal energy consumption in buildings. These methods are based on the size of the available dataset. For a short-term dataset, we have proposed a statistical approach that uses user-defined input as a threshold for anomaly scores. It outputs hourly and day-level binary labels and scores denoting whether the given hour is anomalous or not, and to what extent, respectively. The other method is for long-range data, where an approach based on segmented linear regression is proposed. It uses the correlation between the average temperature values and average energy consumption values to find the anomalous timestamps.

For the second problem, we studied and conducted experiments to evaluate different performance metrics used in the field of anomaly detection. We can therefore conclude that there is no perfect metric available that can capture all kinds of anomalous behaviour. However, the combination of TPR, TNR, FPR,

Rank Power, AUC and FP-100 metrics gives a more robust and accurate view of an algorithm's performance.

The contributions made through this work are: (1) proposed two novel methods to generate labelled data, (2) a publicly available source code to generate labelled data, (3) a publicly available annotated dataset of anomalies, (4) a comprehensive review of different accuracy measures, and (5) a framework and discussion of what performance accuracy metrics to use.

4.3 Anomaly detection using online RPCA

4.3.1 Literature Review

Anomalies observed in energy consumption are either due to malfunctioned appliance or due to change in ambient environment. So contextual (weekday/weekend effect, social gathering) information, if available is used in addition to energy consumption data to detect anomalies. Accordingly, various anomaly detection works are classified either as contextual or non-contextual. In this section we mention all the notable works in anomaly detection.

John[2] uses a statistical approach (mean and standard deviation) to identify anomalous days. He first groups days according to various criteria (weekend/weekday) and then computes anomaly score for each day consumption. The division of days into various groups is done according to energy consumption pattern.

Belalla et al.[3] proposed an unsupervised anomaly detection algorithm. They first created a low dimensional representation of each day energy consumption and then used k-NN to compute anomaly scores by comparing lower dimensional representation of various days. A work proposed by [161] identified anomalous users by periodically computing each user's anomaly score, just by considering their respective energy consumption. This score was then adjusted by analyzing the consumption in the neighborhood for unknown context variables that influence the historic consumption pattern in the same way. Partial

context information which was directly available from the meter readings, i.e., time stamp and meta-data attached to the meter identity is used in their work.

Rashid et al. [195] proposed a generic anomaly detection method for commercial and residential buildings following different energy usage patterns. They use a CCS (Collect, Compare and Score) strategy where first the hour-level energy usage data for several days was collected, then it was compared with other days using euclidean distance. Finally, the score (0/1) was computed for each day using Local Outlier Factor.

Another work [196] proposed various predictive machine learning approaches (*k*-NN, ANN, SVM) to identify anomalies in the energy consumption of university buildings.

Instead of working on directly on energy consumption data, Chen et al. [129] created a symbolic representation of the data on which they used clustering technique to identify anomalies. A comprehensive survey on anomaly detection in several different domains is presented by Chandola et al.[158].

The objective of our approach is to detect the anomalies from the meter-level (house) data. It follows the same framework as followed in video surveillance where the moving objects in the foreground are separated from the background [197]. Robust Principal component analysis (RPCA) is used to decompose the data matrix into low-rank component (L) representing the background sequence and moving foreground object constituting the correlated sparse outliers (S). [198] proposed a convex program called Principal Component Pursuit (PCP)

which surprisingly guarantees the exact recovery of a low-rank matrix L from highly corrupted energy measurements $E = L + S$.

Our work monitors the energy consumption levels of different houses. All these houses are modeled separately. Each column vector in energy consumption matrix (E) represents the hourly energy consumption of a day. The concatenation of such column vectors represents the energy consumed by a house on different days of the month. We can assume that the energy consumed by a house on different days of the month would exhibit a repetitive pattern. If we stack these column vectors of different days together, we would produce a low rank matrix, (L). There would be some days that would not conform to this regular pattern, such days are called anomalous days. These days would be modelled by a sparse matrix. The problem is to decompose the energy consumed (E) into its low-rank (L) and sparse component (S) where the former denotes the fair energy consumed by the users and the latter represents the anomalous behavior of the signal.

Table 4.4 demonstrates how decomposition in RPCA takes place. We take noisy energy consumption, E as input. $E(19, 7)$ denotes the energy consumption of day 7 at 1900 hours. Here, we show the aggregated energy consumption values from 8 days of a single house. RPCA chooses the value of low-rank component based on the energy consumption across 8 days. In this example, the low-rank component is found to be 2243 and sparse component is the remaining of the consumed energy, $(S = E - L)$. Clearly, higher value of (S) represents higher abnormal behaviour. The anomaly scores are assigned values

Table 4.4: Decomposition of total energy (E) into low-rank (L) & sparse matrix (S)

Day	7	8	9	10	11	12	29	30
E(19)	2248	2211	4352	2334	3158	2238	3237	5043
L(19)	2243	2243	2243	2243	2243	2243	2243	2243
S(19)	5	-32	2109	91	915	-5	994	2800

in proportion to the value of the sparse component (S). In this example, we can say that day 30 is the most anomalous day followed by day 9.

The energy signal has outliers that are sparse but can be large in magnitude. The objective of our work is to recover both the signals; however, the sparse component is the object of interest in our case. We apply a convex program called Stable Principal Component Pursuit (SPCP) to recover the L & S matrices using the meter and appliance-level data.

The major contributions of this work include:

1. Modeling energy consumption within buildings as a mixture of low-rank and spare components.
2. Evaluation of our approach on publicly available dataset.
3. Comparison of our approach with state-of-the-art anomaly detection techniques.

4.3.1.1 Robust Principal Component Analysis

Principal Component Analysis (PCA) is a statistical procedure for identifying a smaller number of uncorrelated variables called principal components from a large set of data. The objective of PCA is to explain the maximum amount

of variance with the fewest possible number of principal components. It is the most widely used tool for data analysis and dimensionality reduction. However, when the observations are grossly corrupted, this technique does not perform well. Since the gross errors are ubiquitous in modern day applications, several modifications have been proposed to robustify this technique [199, 200, 201]. However, none of these proposed approaches result in a polynomial-time algorithm with strong performance guarantees.

The first work on robust PCA was developed by Candes et al. [198] who proposed a convex program called Principal Component Pursuit to address the robust PCA problem. This version of RPCA-PCP, guarantees the exact recovery of a low-rank matrix L from highly corrupted energy measurements $E = L + S$. The entries in S are sparse but can have large magnitude unlike the small noise in classical PCA. In our application, the sparse component is the object of interest.

RPCA has found several important applications in areas where the data can be naturally modelled as a low-rank plus a sparse contribution. It is applied to several applications in the field of image and video processing like image analysis, image denoising, motion saliency detection, video coding, foreground and background separation, etc. The past research in RPCA which is based on the decomposition into low-rank and sparse component differs in the loss function, the decomposition, optimization problem and the solvers used in solving the problem. Bouwmans et al. [202] gave a comparative review of RPCA-PCP based methods like RPCA via PCP [198, 203], RPCA via outlier pursuit [204], RPCA via Iteratively reweighted least squares [197, 205, 206], bayesian RPCA

[207], variational BRPCA [208] and approximated RPCA [209] that are used in the field of video surveillance for foreground detection. Wang et al. [210] proposed an efficient face recognition algorithm which is robust in uncontrolled environmental conditions like illumination, expression and occlusion. The authors claim that this method outperforms all other techniques [211, 212, 213] in all conditions. Other areas where RPCA is used are latent semantic indexing [214, 215] where a document-term matrix is decomposed as a sum of low-rank (L) and sparse components (S); (L) being the common words used in all documents and (S) being the keywords that best distinguishes each document. To the best of our knowledge, this is the first time RPCA-PCP has been applied in the energy domain to identify anomalies in residential buildings.

4.3.2 Proposed Approach

In this work, we consider the problem of detecting anomalous days/ appliances from the uniform trend of the noisy energy consumption data in a residential building using robust principal component analysis method.

In Fig. 4.12, we show the hourly power consumption of a randomly chosen house of five consecutive weekdays. The anomalies can be visually identified. In general all the days have three peaks between 5-10, 10-15 and 20-24. One of the peaks (10-15) is missing on day 4. Intuitively this is an anomalous event where the power consumption is less than what it should be (maybe a case of power outage). On the other hand, the third peak (20-24) is much higher for day 1 than the rest of the days. This too is likely to be an anomaly - possibly due to

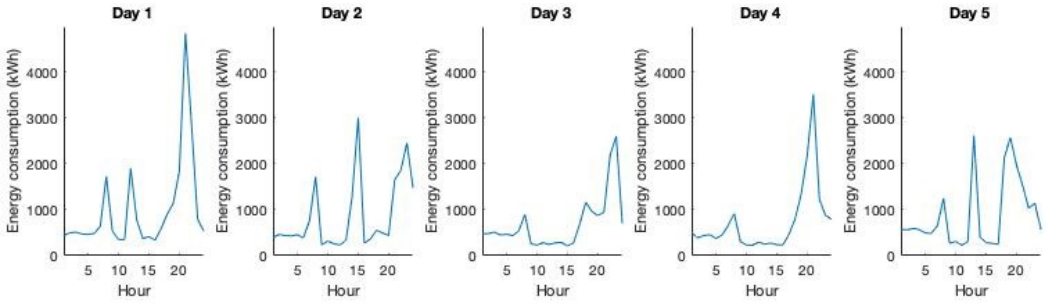


Figure 4.12: Power consumption on 5 consecutive weekdays

power theft.

For both the cases, one can see that the period of anomaly is brief - this follows from the nature of anomaly. Also, from the definition of anomaly, one can say that the power consumption changes sharply for such events. Therefore mathematically speaking, anomalies will always be sparse and sharp. If we assume that the hourly data for each day is represented by a vector x_k, k=1...n, then we can stack the daily vectors as columns of a matrix $X = [x_1|\ldots|x_n]$. This matrix can be segregated into two additive components. The first one models the non-anomalous / normal portion (say L). This portion will remain approximately the same for every day (as can be seen from Fig. 4.12); hence the columns will be linearly dependent. Therefore, the matrix (L) will be of approximately low-rank.

The second portion (say S) models the anomalies. By definition this will be sparse but with relatively high values (both negative - for outage and positive - for theft). Overall, we can express the model as follows,

$$X = L + S + N \tag{4.9}$$

where N is the modelling noise assumed to be Normally distributed. Our task is to estimate L and S given X. This problem is called robust principal component analysis (RPCA) [198]. The solution to (4.9) is called Principal Component Pursuit (PCP). This is expressed as a solution to the following problem,

$$\min_{L,S} \|X - (L + S)\|_F^2 + \lambda\|L\|_* + \mu\|S\|_1 \tag{4.10}$$

Here the first term is a Frobenius norm minimization owing to the Gaussian nature of noise. The nuclear norm models the low rank component; the l_1-norm models the sparse component. PCP is convex being a sum of convex functions. Therefore it is guaranteed to reach the global minimum.

This is the offline version of RPCA. One can only detect and estimate the anomalies once the full data is available. However, in practical situations, an online algorithm is required, i.e. given the estimate till the previous day, how to decompose the current day's power consumption into normal (low-rank) and anomalous (sparse) components? This is achieved via online RPCA [216].

Suppose that for the $k-1^{th}$ day, the X_{k-1} has already been segregated into low-rank (L_{k-1}) and sparse (S_{k-1}) components, i.e. $X_{k-1} = L_{k-1}+S_{k-1}$. When the data for the k^{th} day is available, online RPCA needs to solve -

$$\min_{L_k,S_k} \|X_k - (L_k + S_k)\|_F^2 + \lambda\|L_k\|_* + \mu\|S_k\|_1 \tag{4.11}$$

This is done iteratively by alternately updating the two variables.

$$L_k^{j+1} = \min_{L_k} \|X_k - (L_k + S_k^j)\|_F^2 + \lambda \|L_k\|_* \tag{4.12}$$

$$S_k^{j+1} = \min_{S_k} \|X_k - (L_k^{j+1} + S_k)\|_F^2 + \lambda \|S_k\|_1 \tag{4.13}$$

Here the superscript j denotes the iteration number; and $L_k = [L_{k-1}|l_k]$ and $S_k = [S_{k-1}|s_k]$ where l_k and s_k corresponds to the normal and anomalous (sparse) components for the k^{th} day. Since S_{k-1} and L_{k-1} are known one can efficiently estimate s_k and l_k via thinSVD [217] and projection onto l_1-ball (such as soft thresholding [83]). Both of them have linear computational complexities. This brings down the order of computational complexity from O(n^3) to O(n).

4.3.3 Experimental Setup

4.3.3.1 Simulation Study

We have carried out experiments on the Pecan Street dataset. The dataset has longitudinal power consumption information for houses at both the appliance and meter level. The appliance level information will not be used for this work; we will only use the meter level consumption data.

In prior studies on unsupervised anomaly detection [3, 161, 2] the detection was subjective. In the supervised approaches like [218, 219, 183] the anomaly was assumed to be labeled. In [219, 183] it is not clearly mentioned how the labelling was performed; the availability was tacitly assumed. A more princi-

pled approach was followed in [218, 220] - the anomaly was injected in the data; hence the label information was explicitly available. We follow the same approach.

Strictly speaking there is no other study that can detect anomalies. Other unsupervised techniques [3, 161, 2] can only say if a day was anomalous or not - they cannot temporally segment the anomalies. Therefore these studies are not directly comparable. However, for the sake of benchmarking we have compared with them. If any portion of the day was injected with anomalies, we say that the day was anomalous. Thus we are able to compare with [3, 2]. We have used the standard metric of area under the curve (AUC) (Receiver operating Characteristics) for comparison.

From the dataset different proportions of days (5%, 10% and 25%) were injected with anomalies. The duration of the anomaly (half of it being negative and the other half positive) was varied randomly from a uniform distribution of 1 to 6 hours per day. If anomaly was injected, the entire day was labeled as anomalous, otherwise the day was labeled normal. The AUC values are shown in Table 4.5. The experiments were repeated 1000 times and the mean and the standard deviations are reported. The results using different evaluation metrics were also reported in Table 4.6. The proportion of days which were injected with anomalies in this case was less than 5%. It can be seen here that the proposed techniques outperforms the rest of the methods in identifying anomalies.

Method	Proportion of anomalous days (mean $\pm$ std)		
	5%	10%	25%
Bellala et. al.[3]	0.85 ± 0.16	0.82 ± 0.19	0.61 ± 0.25
Seem [2]	0.78 ± 0.06	0.74 ± 0.11	0.57 ± 0.20
Online RPCA	0.92 ± 0.04	0.91 ± 0.06	0.87 ± 0.11

Table 4.5: Comparative performance (AUC)

Metrics	Methods		
	RPCA	Seem [2]	Bellala et. al.[3]
TPR	**0.783**	0.750	0.771
FPR	**0.066**	0.191	0.412
F1 score	**0.519**	0.482	0.374
Jaccard score	**0.376**	0.251	0.236
AUC	**0.921**	0.869	0.782

Table 4.6: Comparative performance using different evaluation metrics

The results clearly establish the superiority of our method. Not only is ours more accurate, but is more robust as well. The degradation in accuracy with increasing proportion of anomaly is obvious - more the anomaly more does it start looking like normal data and harder it is to separate. While our method shows marginal decline in detection rate between 5% and 25% anomaly, the existing techniques degrade much more rapidly.

4.3.3.2 Experiment on real data

For this study we use data collected on our campus that consists of lecture hall complex, office complex, facilities and utilities, and dormitories. The dataset is called I-BLEND [22]. It consists of meter-level data for all the buildings. On our campus, in the case of a power outage, the supply automatically switches to in-house generation. But heavy loads such as microwaves, washer, refrigerator, air conditioner etc. do not work. Therefore, from the power consumption data one can clearly see reduced consumption (compared to the normal) in the

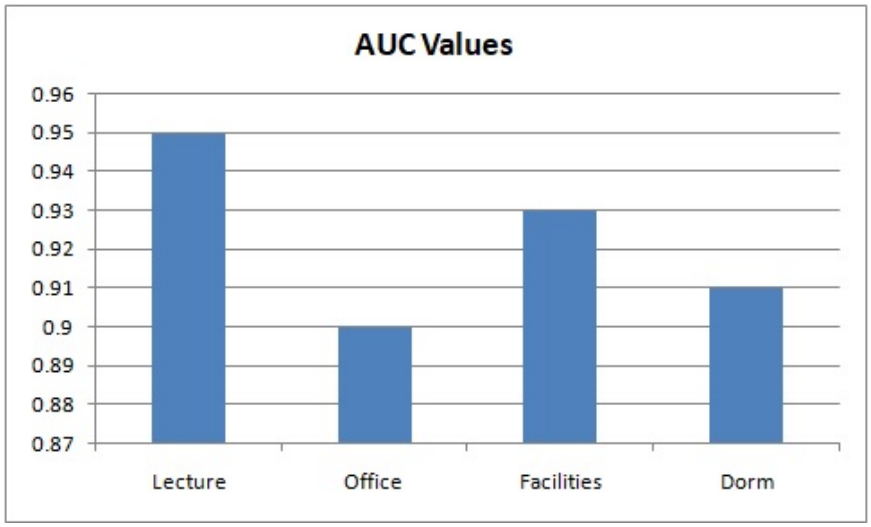


Figure 4.13: AUC values for different buildings

case of a power outage. Being a secure campus, there is no case of power theft. Therefore this dataset only consists of negative anomalies. Unfortunately the anomaly labels are not available in the published dataset; it was curated from afterwards from the facilities department and ratified by General Manager, Operations.

We run online RPCA on this dataset and report the AUC for different buildings. The results are shown in fig. 4.13.

The lecture halls are manually controlled, and hence on normal days show the most uniform pattern. Therefore any anomalous behavior is easy to spot. Offices are the least controlled since faculty members work odd hours, hence the normal pattern is not very regular. This is the reason, the anomaly detection results are the poorest here. Facilities building is also largely manually controlled (normal behavior is uniform) and hence anomalies are easy to detect. Dorms, although occupied by students who have varied lifestyles, are manually

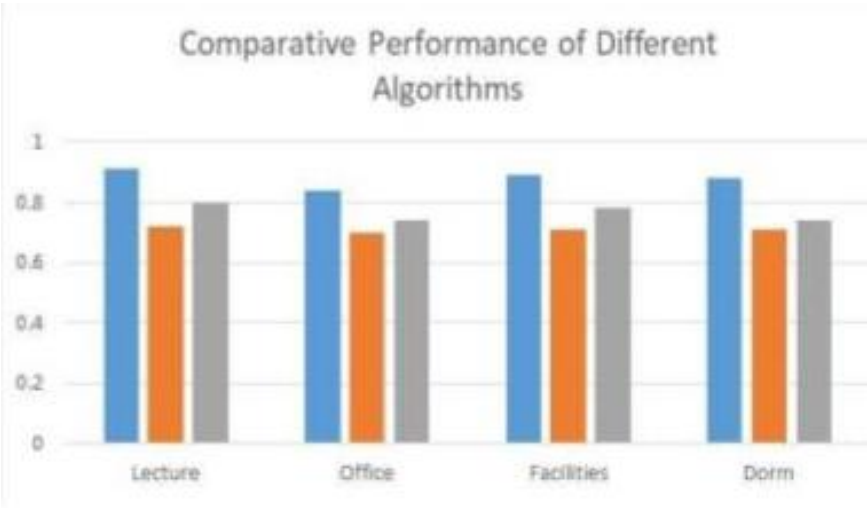


Figure 4.14: Comparative performance in terms of AUC for different buildings on campus. From left to right, we have shown lecture buildings, office, facilities and dorm. Here, blue represents RPCA (proposed), orange is for [2] and grey denotes [3]

.

controlled to a certain degree. For example, ACs are switched ON and OFF centrally. Hence the anomaly detection results are better than that of offices.

Next we compare the day-level anomaly detection algorithms of Bellala et al [3] and Seem [2] with the proposed online RPCA. The AUC plot is shown in Fig. 4.14. The results corroborate the findings of synthetic experiments.

4.3.4 Summary

The goal of this work had been to develop an approach to detect anomalies in meter level energy consumption. Anomalies were assumed to be of two types - power theft and power outage leading to either positive or negative anomalies. Robust Principal Component Analysis (RPCA) was employed for unsupervised anomaly detection. The results show that we perform better than existing techniques.

Chapter 5

Conclusion and Future Work

In this work, we focused on the techniques that empower power utilities to formulate efficient demand-side activities for sustainable energy. These include non-intrusive load monitoring, load forecasting and anomaly detection. The main contributions of this thesis are summarized below -

1. In the first part of this thesis, we proposed to modify the mathematical framework for disaggregation to address two different problems. The first problem is of the missing data and the other is the problem of data corruption by large yet sparse outliers. The resultant mathematical framework after incorporating both the modifications has been shown to yield improved accuracy compared to the benchmarks.

2. Next, the work on learning robust dictionaries is extended to deal with the problem of data corruption by large and sparse outliers. The problems with the assumptions made by prior studies about the linear mixing model have been discussed. As an extension, an approach that combines learning

robust dictionaries and learning rank deficient codes was proposed.

3. In the next contribution, two supervised models on top of the unsupervised robust dictionary learning are proposed to ensure that the learned dictionaries and sparse codes generated look different for each appliance. The proposed models were shown to have improved accuracy over the existing works.

4. The cost of NILM is directly proportional to the amount of data acquired at the training phase. To reduce this cost, the analysis equivalent of dictionary learning for NILM was proposed. It has better representation capability than a dictionary. In practical scenarios of low training data regime, this method always excelled over the state-of-the-art techniques.

5. The problem of short-term load forecasting was solved using Kalman filtering algorithms. Owing to the problem's non-linearity and non-stationarity, we resort to using the nonlinear variants of the Kalman filter which are extended Kalman filters (EKF) and unscented Kalman filters (UKF). The research concluded that both the inputs (temperature and wind speed) in conjunction with the past load improved the performance accuracy of the model.

6. A deeper extension and generalization of the sparse coding based forecasting approach was proposed. The improvement upon the prior work was shown two ways. First, using deep dictionary learning, which is a deeper non-linear extension of shallow dictionary learning. Second, in-built re-

gression into the deep dictionary learning process. The results showed improved prediction accuracy along with faster run-time.

7. Two problems commonly faced by the anomaly detection research community were mitigated. First, lack of availability of ground truth to test the algorithms and second, lack of a unified performance accuracy metric. Two approaches were proposed to generate the ground truth based on the range of the available dataset. A detailed analysis of a list of performance metrics used in the literature was presented. The existing works were evaluated against the ground truth generated by the proposed methods using all the different performance metrics to review what works best with the problem at hand.

8. An approach for unsupervised anomaly detection using robust principal component analysis (RPCA) was proposed. Anomalies injected in the dataset are assumed to be positive (power outage) and negative (power theft). The results from the proposed method showed that proposed method outperformed the existing techniques.

Some of the future research directions are highlighted below -

1. In this work, supervised NILM using dictionary learning and sparse coding methods has been explored. These methods required sub-metering active power measurements during training phase which did not fully qualify the non-intrusive aspect of NILM. Recent studies have explored semi-supervised [221, 222] and unsupervised [223, 28, 26, 49] approaches to

solve NILM. Semi-supervised methods require small number of labelled training samples in a larger corpus of unlabelled training samples. Though unsupervised techniques are a better fit for NILM problems but supervised and semi-supervised approaches give better accuracies at disaggregation. The next task would be delve more into semi-supervised and unsupervised techniques for NILM.

2. In this work, load is predicted for a short horizon, that is prediction ahead of days to weeks. An extension to this work could be to use appliance-level load forecasting to detect anomalies present in the home appliances. Another extension is to use NILM and anomaly detection in a single framework.

3. The sparse coding deep dictionary learning models performed better overall but the predictions from these methods looked like lagged version of actual values. This requires further investigation by tuning different parameters like the size of the dictionaries.

4. In anomaly detection, there is no unified accuracy metrics that could evaluate the performance of different approaches. In the proposed work, a list of metrics to evaluate different approaches is used. It is found that no single metric gives consistent results on different techniques. To solve this issue, a weighted combination of different metrics can be devised that further can be used by the research community.

References

[1] A. Majumdar and R. Ward, "Robust dictionary learning: Application to signal disaggregation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 2469–2473.

[2] J. E. Seem, "Using intelligent data analysis to detect abnormal energy consumption in buildings," *Energy and Buildings*, vol. 39, no. 1, pp. 52 – 58, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778806001514

[3] G. Bellala, M. Marwah, M. Arlitt, G. Lyon, and C. E. Bash, "Towards an understanding of campus-scale power consumption," in *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, ser. BuildSys '11. New York, NY, USA: ACM, 2011, pp. 73–78. [Online]. Available: http://doi.acm.org/10.1145/2434020.2434043

[4] S. Sharma K, "India: Greenhouse gas emissions 2007," *Ministry of Environment and Forests, Government of India*, pp. 1–84, May

2010. [Online]. Available: https://www.iitr.ac.in/wfw/web_ua_water_for_welfare/water/WRDM/MOEF_India_GHG_Emis_2010.pdf

[5] N. Girouard, E. Konialis, C. Tam, and P. Taylor, "Oecd green growth studies," May 2011. [Online]. Available: https://www.oecd.org/greengrowth/greening-energy/49157219.pdf

[6] K. Ehrhardt-Martinez, K.and Donnelly and J. Laitner, "Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities," June 2010.

[7] E. Almeshaiei and H. Soltan, "A methodology for electric power load forecasting," *Alexandria Engineering Journal*, vol. 50, no. 2, pp. 137 – 144, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1110016811000330

[8] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1484–1491, Nov 1989.

[9] V. S. K. Murthy Balijepalli, V. Pradhan, S. A. Khaparde, and R. M. Shereef, "Review of demand response under smart grid paradigm," in *ISGT2011-India*, Dec 2011, pp. 236–243.

[10] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug 2011.

[11] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213 – 234, 2013, special Section: Transition Pathways to a Low Carbon Economy. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421512007446

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

[13] J. R. B. Neenan, "Residential electricity use feedback: A research synthesis and economic framework," February 2009. [Online]. Available: http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000000001016844&Mode=download

[14] R. Mathur, "The role of energy efficiency in the indian context," *Mitigation Talks*, vol. 1, pp. 7–9, 08-12 2017.

[15] M. Lee and K. Colopinto, "World bank cities and climate change mitigation: Case study on tokyoâĂŹs emissions trading system," May 2010.

[16] A. Golub, R. Balassiano, A. Araújo, and E. Ferreira, "Regulation of the informal transport sector in rio de janeiro, brazil: Welfare impacts and policy analysis," *Transportation*, vol. 36, no. 5, pp. 601–616, 2009.

[17] H. T. Brandon Davito and R. Uhlaner, "The smart grid and the promise of demand side management," December 2009. [Online]. Available: https://www.smartgrid.gov/files/The_Smart_Grid_Promise_DemandSide_Management_201003.pdf

[18] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, N. D. Lawrence and M. A. Girolami, Eds., vol. 22, 2012, pp. 1472–1482. [Online]. Available: http://jmlr.csail.mit.edu/proceedings/papers/v22/zico12/zico12.pdf

[19] J.Z.Kolter and M.J.Johnson, "Redd: A public dataset for energy disaggregation research," in *Proceedings of the SustKDD workshop on DAta Mining Applications in Sustainability*, 2012.

[20] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmtk: An open source toolkit for non-intrusive load monitoring," in *In ACM E-Energy*, 2014.

[21] S. Makonin, "Hue: The hourly usage of energy dataset for buildings in british columbia," *Data in Brief*, vol. 23, p. 103744, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352340919300939

[22] H. Rashid, P. Singh, and A. Singh, "I-blend, a campus-scale commercial and residential buildings electrical energy dataset," *Scientific Data*, vol. 6, pp. 1–12, Feb 2019, data Descriptor. [Online]. Available: https://doi.org/10.1038/sdata.2019.15

[23] P. Chakravarty and A. Gupta, "Impact of energy disaggregation on consumer behavior," 2013. [Online]. Available: http://www.escholarship.

org/uc/item/62d3456p

[24] G. Crabtree, L. Glicksman, D. B. Goldstein, D. Goldston, D. Greene, D. M. Kammen, M. Levine, M. Lubell, B. Richter, M. Savitz, and D. Sperling, "Energy future - think efficiency: How america can look within to achieve energy security and reduce global warming," 2008. [Online]. Available: http://www.aps.org/energyefficiencyreport/

[25] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, pp. 356–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2900728.2900780

[26] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-markov models," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 673–701, Feb. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2502581.2502602

[27] H. GonÃğalves, A. Ocneanu, and M. Berges, "Unsupervised disaggregation of appliances using aggregated consumption data," in *International Conference on Cloud Computing and Security*, 2011.

[28] D. Egarter and W. Elmenreich, "Autonomous load disaggregation approach based on active power measurements," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, March 2015, pp. 293–298.

[29] A. Ruano, A. Hernandez, J. UreÃśa, M. Ruano, and J. Garcia, "Nilm techniques for intelligent home energy management and ambient assisted living: A review," *Energies*, vol. 12, no. 11, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/11/2203

[30] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas, "Non-intrusive load monitoring through home energy management systems: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1266 – 1274, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032117307359

[31] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, and M. Mustapha, "Application of load monitoring in appliancesâĂŹ energy management âĂŞ a review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 235 – 245, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136403211630555X

[32] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 1153–1161. [Online]. Available: http://books.nips.cc/papers/files/nips23/NIPS2010_1272.pdf

[33] E. Elhamifar and S. Sastry, "Energy disaggregation via learning 'powerlets' and sparse coding," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press,

2015, pp. 629–635. [Online]. Available: http://dl.acm.org/citation.cfm?id=2887007.2887095

[34] K. Basu, V. Debusschere, B. Seddik, U. Maulik, and S. Bandyopadhyay, "Non intrusive load monitoring: A temporal multi-label classification approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 262–270, 10 2014.

[35] D. Li and S. Dick, "Whole-house non-intrusive appliance load monitoring via multi-label classification," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 2749–2755.

[36] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, Jan 2017.

[37] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *IEEE Power and Energy Magazine*, vol. 1, no. 2, pp. 56–63, Mar 2003.

[38] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, *Unsupervised Disaggregation of Low Frequency Power Measurements*, ch. 64, pp. 747–758. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.64

[39] M. Wytock and J. Z. Kolter, "Contextually supervised source separation with application to energy disaggregation," *CoRR*, vol. abs/1312.5023, 2014.

[40] M. Baranski and V. J, "Nonintrusive appliance load monitoring based on an optical sensor," in *Power Tech Conference Proceedings, 2003 IEEE Bologna*, vol. 4, June 2003, pp. 8 pp. Vol.4–.

[41] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study;part i: Basic concept, structure, and methodology," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, April 2010.

[42] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in nialm systems," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, Oct 2004, pp. 3462–3468 vol.4.

[43] A. Marchiori, D. Hakkarinen, Q. Han, and L. Earle, "Circuit-level load monitoring for household energy management," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 40–48, Jan 2011.

[44] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, February 2011.

[45] G. yuan Lin, S. chiang Lee, J. Y. jen Hsu, and W. rong Jih, "Applying power meters for appliance recognition on the electric panel," in *2010 5th IEEE Conference on Industrial Electronics and Applications*, June 2010, pp. 2254–2259.

[46] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *2010 7th Annual IEEE Communications Society Conference*

on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), June 2010, pp. 1–9.

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

[48] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2, Dec 2012, pp. 515–520.

[49] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[50] V. Stankovic, J. Liao, and L. Stankovic, "A graph-based signal processing approach for low-rate energy disaggregation," in *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, Dec 2014, pp. 81–87.

[51] H. Shao, M. Marwah, and N. Ramakrishnan, "A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, ser. AAAI'13. AAAI Press, 2013, pp. 1327–1333. [Online]. Available: http://dl.acm.org/citation.cfm?id=2891460.2891645

[52] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, p. 16838, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/12/16838

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

[54] C. R. Paul, *Introduction to Electromagnetic Compatibility (Wiley Series in Microwave and Optical Engineering)*. Wiley-Interscience, 2006.

[55] L. Wang, M. D. Gordon, and J. Zhu, "Regularized least absolute deviations regression and an efficient algorithm for parameter tuning," in *Proceedings of the Sixth International Conference on Data Mining*, ser. ICDM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 690–700. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2006.134

[56] E. J. Schlossmacher, "An iterative technique for absolute deviations curve fitting," *Journal of the American Statistical Association*, vol. 68, no. 344, pp. 857–859, 1973. [Online]. Available: http://www.jstor.org/stable/2284512

[57] G. O. Wesolowsky, "A new descent algorithm for the least absolute value regression problem," *Communications in Statistics - Simulation and Computation*, vol. 10, no. 5, pp. 479–491, 1981. [Online]. Available: http://dx.doi.org/10.1080/03610918108812224

[58] Y. Li and G. R. Arce, "A maximum likelihood approach to least absolute deviation regression," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 12, pp. 1–8, 2004. [Online]. Available: http://dx.doi.org/10.1155/S1110865704401139

[59] S. Makonin, "Real-time embedded low-frequency load disaggregation," Ph.D. dissertation, SIMON FRASER UNIVERSITY, 2014. [Online]. Available: https://github.com/smakonin/DataWrangle_REDD

[60] J. Peppanen, Xiaochen Zhang, S. Grijalva, and M. J. Reno, "Handling bad or missing smart meter data through advanced data imputation," in *2016 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Sep. 2016, pp. 1–5.

[61] D. L. X. D. X. Zhou, C. Zhou, *Applied Missing Data Analysis in the Health Sciences*, 1st ed. Wiley Publishing, 2014.

[62] P. Allison, *Missing data*, ser. Quantitative Applications in the Social Sciences. SAGE Publications, Inc, August 2001, vol. 136.

[63] B. Zhao, K. He, L. Stankovic, and V. Stankovic, "Improving event-based non-intrusive load monitoring using graph signal processing," *IEEE Access*, pp. 1–15, September 2018. [Online]. Available: https://strathprints.strath.ac.uk/65550/

[64] M. Aiad and P. H. Lee, "Non-intrusive load disaggregation with adaptive estimations of devices main power effects and two-way interactions,"

Energy and Buildings, vol. 130, pp. 131 – 139, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778816307472

[65] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Comm. Pure Appl. Math*, vol. 59, pp. 797–829, 2004.

[66] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[67] S. Gleichman and Y. C. Eldar, "Blind compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6958–6975, Oct 2011.

[68] A. Majumdar, N. Ansari, H. Aggarwal, and P. Biyani, "Impulse denoising for hyper-spectral images: A blind compressed sensing approach," *Signal Processing*, vol. 119, pp. 136 – 141, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168415002546

[69] P. J. Huber, *Robust Estimation of a Location Parameter*. New York, NY: Springer New York, 1992, pp. 492–518. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-4380-9_35

[70] J. Branham, R. L., "Alternatives to least squares," *Astronomical Journal, vol. 87, June 1982, p. 928-937. (AJ Homepage)*, June 1982.

[71] M. Shi and M. A. Lukas, "An l1 estimation algorithm with degeneracy and linear constraints," *Computational Statistics & Data*

Analysis, vol. 39, no. 1, pp. 35 – 55, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947301000494

[72] Y. Zhang, "User's guide for YALL1: Your algorithms for l1 optimization," *Technical Report*, pp. 1–9, July 2009.

[73] A. Majumdar and R. Ward, "Some empirical advances in matrix completion," *Signal Processing*, vol. 91, no. 5, pp. 1334 – 1338, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168410004196

[74] A. Majumdar and R. K. Ward, "Increasing energy efficiency in sensor networks: blue noise sampling and non-convex matrix completion," *IJSNet*, vol. 9, no. 3/4, pp. 158–169, 2011. [Online]. Available: http://dx.doi.org/10.1504/IJSNET.2011.040237

[75] I. Selesnick, "Sparse signal restoration," September 2009. [Online]. Available: http://cnx.org/contents/yccwvhC3@3/Sparse-Signal-Restoration

[76] P. Hansen and D. O'Leary, "The use of the l-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1487–1503, 1993. [Online]. Available: http://dx.doi.org/10.1137/0914086

[77] C. FÃl'votte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4810–4819, Dec. 2015.

[Online]. Available: http://www.unice.fr/cfevotte/publications/journals/tip2015.pdf

[78] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3501–3508.

[79] H. Nien and J. A. Fessler, "A convergence proof of the split bregman method for regularized least-squares problems," *SIAM J. Imaging Sci.*, vol. 4371, 02 2014.

[80] P. L. Combettes and J.-C. Pesquet, *Proximal Splitting Methods in Signal Processing*. New York, NY: Springer New York, 2011, pp. 185–212. [Online]. Available: https://doi.org/10.1007/978-1-4419-9569-8_10

[81] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of admm," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 343–352. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045156

[82] Y. Wang, W. Yin, and J. Zeng, "Global Convergence of ADMM in Nonconvex Nonsmooth Optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, Jan 2015. [Online]. Available: https://doi.org/10.1007/s10915-018-0757-z

[83] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[84] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20042

[85] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, "Dataport and NILMTK: A building data set designed for non-intrusive load monitoring," in *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015, pp. 210–214.

[86] S. Singh and A. Majumdar, "Analysis co-sparse coding for energy disaggregation," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 462–470, Jan 2019.

[87] M. Gulati, S. S. Ram, A. Majumdar, and A. Singh, "Single point conducted emi sensor with intelligent inference for detecting it appliances," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3716–3726, July 2018.

[88] P. Bhattacharjee, S. Banerjee, M. Gulati, A. Majumdar, and S. S. Ram, "Supervised analysis dictionary learning: Application in consumer electronics appliance classification," in *Proceedings of the Fourth*

ACM IKDD Conferences on Data Sciences, ser. CODS '17. New York, NY, USA: ACM, 2017, pp. 2:1–2:10. [Online]. Available: http://doi.acm.org/10.1145/3041823.3041825

[89] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, March 2013.

[90] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning - part i: Algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 625–636, June 2015.

[91] S. Ravishankar and Y. Bresler, "Online sparsifying transform learning - part ii: Convergence analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 637–646, June 2015.

[92] R. Jia, Y. Gao, and C. J. Spanos, "A fully unsupervised non-intrusive load monitoring framework," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2015, pp. 872–878.

[93] Y. Xu, Z. Li, J. Yang, and D. Zhang, "A survey of dictionary learning algorithms for face recognition," *IEEE Access*, vol. 5, pp. 8502–8514, 2017.

[94] L. Jia, S. Song, L. Yao, H. Li, Q. Zhang, Y. Bai, and Z. Gui, "Image denoising via sparse representation over grouped dictionaries with adaptive atom size," *IEEE Access*, vol. 5, pp. 22 514–22 529, 2017.

[95] S. Shekhar, V. M. Patel, and R. Chellappa, "Analysis sparse coding models for image-based classification," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5207–5211.

[96] J. Maggu and A. Majumdar, "Robust transform learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1467–1471.

[97] L. Pfister and Y. Bresler, "Tomographic reconstruction with adaptive sparsifying transforms," 05 2014, pp. 6914–6918.

[98] B. Wen, S. Ravishankar, and Y. Bresler, "Video denoising by online 3d sparsifying transform learning," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 118–122.

[99] S. Ravishankar and Y. Bresler, "Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015. [Online]. Available: https://doi.org/10.1137/141002293

[100] J. Guo, Y. Guo, X. Kong, M. Zhang, and R. He, "Discriminative analysis dictionary learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 1617–1623. [Online]. Available: http://dl.acm.org/citation.cfm?id=3016100.3016125

[101] J. Maggu and A. Majumdar, "Greedy deep transform learning," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 1822–1826.

[102] M. Gastaldi, R. Lamedica, A. Nardecchia, and A. Prudenzi, "Short-term forecasting of municipal load through a kalman filtering based approach," in *Power Systems Conference and Exposition, 2004. IEEE PES*, Oct 2004, pp. 1453–1458 vol.3.

[103] S. Mishra, "Short term load forecasting using computational intelligence methods," Masters thesis, National Institute of Technology, Rourkela, Rourkela, India, 2008.

[104] G. Gross and F. D. Galiana, "Short-term load forecasting," *Proceedings of the IEEE*, vol. 75, no. 12, pp. 1558–1573, Dec 1987.

[105] S. Patnaik, A. Baliyan, K. Gaurav, and S. K. Mishra, "International conference on computer, communication and convergence (iccc 2015) a review of short term load forecasting using artificial neural network models," *Procedia Computer Science*, vol. 48, pp. 121 – 125, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915006699

[106] A. Jain and B. Satish, "Integrated approach for short term load forecasting using svm and ann," in *TENCON 2008 - 2008 IEEE Region 10 Conference*, Nov 2008, pp. 1–6.

[107] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[108] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2213–2219, 2007.

[109] L. HernÃąndez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, and J. Massana, "A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings," *Communications Surveys Tutorials, IEEE*, vol. 16, pp. 1460–1495, 04 2014.

[110] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 1–1, 06 2015.

[111] P. Siano, "Demand response and smart gridsâĂŤa survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461 – 478, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032113007211

[112] A. Quintana, J. Gomez, and N. D. Reppen, "Integration of an adaptive short-term load forecast procedure into a new energy control center," *7th Power Systems Computation Conference*, vol. 7, pp. 426–431, 1978.

[113] H. R. Fankhauser, "A novel approach to on-line short and intermediate term load forecasting," *8th Power Systems Computation Conference*, vol. 8, pp. 376–380, 1984.

[114] R. Anelli, U. D. Caprio, V. Marchese, and S. Pozzi, "Short term prediction of stationary load processes with a correlation function finite sum of exponentials," *7th Power Systems Computation Conference*, vol. 7, pp. 401–408, 1978.

[115] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1535–1547, Nov 1990.

[116] D. D. Belik, D. J. Nelson, and D. W. Olive, "Use of the karhunen-loeve expansion to analyze hourly load requirements for a power utility," *IEEE Power Engineering Society Winter Meeting*, vol. A78, pp. 225–230, January 1978.

[117] C. Yu, P. Mirowski, and T. K. Ho, "A sparse coding approach to household electricity demand forecasting in smart grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 738–748, March 2017.

[118] M. s. Abou-Hussien, M. S. Kandlil, M. A. Tantawy, and S. A. Farghal, "An accurate model for short-term load forecasting," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 9, pp. 4158–4165, Sep. 1981.

[119] G. Gross and F. Galiana, "Short-term load forecasting." *Proceedings of the IEEE*, vol. 75, no. 12, pp. 1558–1573, 12 1987.

[120] Shyh-Jier Huang and Kuang-Rong Shih, "Short-term load forecasting via arma model identification including non-gaussian process considerations," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 673–679, May 2003.

[121] J. H. Park, Y. M. Park, and K. Y. Lee, "Composite modeling for adaptive short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 450–457, May 1991.

[122] T. Zheng, A. A. Girgis, and E. B. Makram, "A hybrid wavelet-kalman filter method for load forecasting," *Electric Power Systems Research*, vol. 54, no. 1, pp. 11 – 17, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779699000632

[123] H. Al-Hamadi and S. Soliman, "Short-term electric load forecasting based on kalman filtering algorithm with moving window weather and load model," *Electric Power Systems Research*, vol. 68, no. 1, pp. 47 – 59, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779603001500

[124] N. Amjady, "Short-term bus load forecasting of power systems by a new hybrid method," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 333–341, Feb 2007.

[125] J. R. Azzam-ul-Asar, J. R. McDonald, and M. I. Khan, "A specification of neural network applications in the load forecasting problem," in *[Proceedings 1992] The First IEEE Conference on Control Applications*, Sep. 1992, pp. 577–582 vol.l.

[126] K.-H. Kim, J.-K. Park, K.-J. Hwang, and S.-H. Kim, "Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems," *Power Systems, IEEE Transactions on*, vol. 10, pp. 1534 – 1539, 09 1995.

[127] A. Khotanzad, R. Afkhami-Rohani, Tsun-Liang Lu, A. Abaye, M. Davis, and D. J. Maratukulam, "Annstlf-a neural-network-based electric load forecasting system," *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 835–846, July 1997.

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

[129] C. Chen and D. Cook, "Energy outlier detection in smart environments," in *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Smarter Living: The Conquest of Complexity*, ser. AAAIWS'11-07. AAAI Press, 2011, pp. 9–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=2908658.2908660

[130] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on*

Power Systems, vol. 28, no. 4, pp. 4356–4364, Nov 2013.

[131] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *Expert Systems with Applications*, vol. 41, no. 13, pp. 6047 – 6056, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414001912

[132] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian, "Short-term electric load forecasting using echo state networks and pca decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, October 2015.

[133] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," *CoRR*, vol. abs/1610.09460, 2016. [Online]. Available: http://arxiv.org/abs/1610.09460

[134] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan 2019.

[135] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, "An overview and comparative analysis of recurrent neural networks for short term load forecasting," *CoRR*, vol. abs/1705.04378, 2017. [Online]. Available: http://arxiv.org/abs/1705.04378

[136] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa, "Deep dictionary learning," *IEEE Access*, vol. 4, pp. 10 096–10 109, December 2016.

[137] V. Singhal, J. Maggu, and A. Majumdar, "Simultaneous detection of multiple appliances from smart-meter measurements via multi-label consistent deep dictionary learning and deep transform learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2969–2978, May 2019.

[138] F. L. Quilumba, W. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911–918, March 2015.

[139] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *Smart Grid, IEEE Transactions on*, vol. 5, pp. 411–419, 01 2014.

[140] Y. Goude, R. Nedellec, and N. Kong, "Local short and middle term electricity load forecasting with semi-parametric additive models," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 440–446, Jan 2014.

[141] V. Thouvenot, A. Pichavant, Y. Goude, A. Antoniadis, and J. Poggi, "Electricity forecasting using multi-stage estimators of nonlinear additive models," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3665–3673, Sep. 2016.

[142] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914

– 938, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207015001508

[143] E. Esser, "Applications of lagrangian-based alternating direction methods and connections to split bregman," *CAM Rep*, vol. 9, 01 2009.

[144] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality," *Math. Oper. Res.*, vol. 35, pp. 438–457, 2010.

[145] J. M. Maggu and A. Majumdar, "Unsupervised deep transform learning," 04 2018, pp. 6782–6786.

[146] B. Liu, J. Nowotarski, T. Hong, and R. Weron, "Probabilistic load forecasting via quantile regression averaging on sister forecasts," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 730–737, March 2017.

[147] S. Ben Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2448–2455, Sep. 2016.

[148] P. H. John J Conti, "Annual energy outlook 2016." US EIA, 2016. [Online]. Available: http://www.eia.gov/forecasts/aeo/pdf/0383(2016).pdf

[149] "Annual Report 2016-2017." Bureau of Energy Efficiency, Govt of India, Ministry of Power, 2016. [Online]. Available: http://www.beeindia.gov.in/content/annual-report

[150] "Energy Efficiency in Commercial Buildings." Energy Star, US Environmental Protection Agency, 2016. [Online]. Available: https://www.energystar.gov/ia/partners/publications/pubdocs/C+I_brochure.pdf

[151] "Estimated U.S. Energy Use in 2012." Lawrence Livermore National Laboratory (LLNL) and Department of Energy (DOE), May 2013. [Online]. Available: https://flowcharts.llnl.gov/commodities/energy

[152] Y. Zhang, W. Chen, and J. Black, "Anomaly detection in premise energy consumption data," in *2011 IEEE Power and Energy Society General Meeting*, July 2011, pp. 1–8.

[153] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[154] T. Fawcett and F. Provost, "Adaptive fraud detection," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 291–316, Sep 1997. [Online]. Available: https://doi.org/10.1023/A:1009700419189

[155] W. DuMouchel and M. Schonlau, "A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities," in *KDD*. AAAI Press, 1998, pp. 189–193.

[156] J. Tang, Z. Chen, A. W. Fu, and D. W. Cheung, "Capabilities of outlier detection schemes in large datasets, framework and methodologies,"

Knowledge and Information Systems, vol. 11, no. 1, pp. 45–84, Jan 2007. [Online]. Available: https://doi.org/10.1007/s10115-005-0233-6

[157] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environmental Modelling & Software*, vol. 25, no. 9, pp. 1014 – 1022, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815209002321

[158] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[159] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1257–1272, Nov 2012.

[160] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in ami using customers consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, Jan 2016.

[161] P. Arjunan, H. D. Khadilkar, T. Ganu, Z. M. Charbiwala, A. Singh, and P. Singh, "Multi-user energy consumption monitoring and anomaly detection with partial context information," in *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, ser. BuildSys '15. New

York, NY, USA: ACM, 2015, pp. 35–44. [Online]. Available: http://doi.acm.org/10.1145/2821650.2821662

[162] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[163] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, Oct 2004. [Online]. Available: https://doi.org/10.1007/s10462-004-4304-y

[164] M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215 – 249, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016516841300515X

[165] M. Markou and S. Singh, "Novelty detection: a review—part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481 – 2497, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168403002020

[166] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with ICDM*, 2003, pp. 171–179.

[167] R. Pincus, "Barnett, v., and lewis t.: Outliers in statistical data. 3rd edition. j. wiley & sons 1994, xvii. 582 pp., Âč49.95," *Biometrical*

Journal, vol. 37, no. 2, pp. 256–256, 1995. [Online]. Available: http://dx.doi.org/10.1002/bimj.4710370219

[168] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 427–438. [Online]. Available: http://doi.acm.org/10.1145/342009.335437

[169] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, ser. VLDB '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 392–403. [Online]. Available: http://dl.acm.org/citation.cfm?id=645924.671334

[170] J. Tang, Z. Chen, A. W. chee Fu, and D. Cheung, "A robust outlier detection scheme for large data sets," in *In 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2001, pp. 6–8.

[171] D. Wettschereck, "A study of distance-based machine learning algorithms," Ph.D. dissertation, Corvallis, OR, USA, 1994, aAI9507711.

[172] B. Rosner, "Percentage points for a generalized esd many-outlier procedure," *Technometrics*, vol. 25, no. 2, pp. 165–172, 1983. [Online]. Available: http://www.jstor.org/stable/1268549

[173] S. Wang and F. Xiao, "Ahu sensor fault diagnosis using principal component analysis method," *Energy and Buildings*, vol. 36, no. 2, pp.

147 – 160, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S03787788803001178

[174] B. Narayanaswamy, B. Balaji, R. Gupta, and Y. Agarwal, "Data driven investigation of faults in hvac systems with model, cluster and compare (mcc)," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York, NY, USA: ACM, 2014, pp. 50–59. [Online]. Available: http://doi.acm.org/10.1145/2674061.2674067

[175] M. Peña, F. Biscarri, J. I. Guerrero, I. Monedero, and C. León, "Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach," *Expert Systems with Applications*, vol. 56, pp. 242 – 255, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416300938

[176] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," 09 2012.

[177] M. J. Desforges, P. J. Jacob, and J. E. Cooper, "Applications of probability density estimation to the detection of abnormal conditions in engineering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 212, no. 8, pp. 687–703, 1998. [Online]. Available: https://doi.org/10.1243/0954406981521448

[178] S. A. Kalogirou, "Applications of artificial neural-networks for energy systems," *Applied Energy*, vol. 67, no. 1, pp. 17 – 35, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261900000052

[179] S. Karatasou, M. Santamouris, and V. Geros, "Modeling and predicting building's energy use with artificial neural networks: Methods and results," *Energy and Buildings*, vol. 38, no. 8, pp. 949 – 958, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778805002161

[180] M. Brown, C. Barrington-Leigh, and Z. Brown, "Kernel regression for real-time building energy analysis," *Journal of Building Performance Simulation*, vol. 5, no. 4, pp. 263–276, 2012. [Online]. Available: https://doi.org/10.1080/19401493.2011.577539

[181] Z. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.

[182] C. C. Aggarwal, "Outlier ensembles: Position paper," *SIGKDD Explor. Newsl.*, vol. 14, no. 2, pp. 49–58, Apr. 2013. [Online]. Available: http://doi.acm.org/10.1145/2481244.2481252

[183] D. B. Araya, K. Grolinger, H. F. Elyamany, M. Capretz, and G. Bitsuamlak, "An ensemble learning framework for anomaly detection in building energy consumption," *Energy and Buildings*, vol. 144, pp. 191 – 206, 2017. [On-

line]. Available: https://ir.lib.uwo.ca/cgi/viewcontent.cgi?referer=https://www.google.ca/&httpsredir=1&article=1153&context=electricalpub

[184] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Applied Energy*, vol. 211, pp. 1123 – 1135, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261917317166

[185] A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4324 – 4338, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415000251

[186] A. Kialashaki and J. R. Reisel, "Modeling of the energy demand of the residential sector in the united states using regression models and artificial neural networks," *Applied Energy*, vol. 108, pp. 271 – 280, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261913002304

[187] P. Singh and P. Dwivedi, "Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem," *Applied Energy*, vol. 217, pp. 537 – 549, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261918302654

[188] J. Chou and A. S. Telaga, "Real-time detection of anomalous power consumption," *Renewable and Sustainable Energy Reviews*, vol. 33, no. Supplement C, pp. 400 – 411, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032114001142

[189] I. Khan, A. Capozzoli, S. P. Corgnati, and T. Cerquitelli, "Fault detection analysis of building energy consumption using data mining techniques," *Energy Procedia*, vol. 42, pp. 557 – 566, 2013, mediterranean Green Energy Forum 2013: Proceedings of an International Conference MGEF-13. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1876610213017591

[190] R. Fontugne, J. Ortiz, N. Tremblay, P. Borgnat, P. Flandrin, K. Fukuda, D. Culler, and H. Esaki, "Strip, bind, and search: A method for identifying abnormal energy consumption in buildings," in *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2013, pp. 129–140.

[191] B. Amidan, T. Ferryman, and S. Cooley, "Data outlier detection using the chebyshev theorem," April 2005, pp. 3814 – 3819.

[192] M. Ali, Y. S. Kwon, C. Lee, J. Kim, and Y. Kim, *Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer Publishing Company, Incorporated, 2015.

[193] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[194] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[195] H. Rashid, P. Arjunan, P. Singh, and A. Singh, "Collect, compare, and score: A generic data-driven anomaly detection method for buildings," in *Proceedings of the Seventh International Conference on Future Energy Systems Poster Sessions*, ser. e-Energy '16. New York, NY, USA: ACM, 2016, pp. 12:1–12:2. [Online]. Available: http://doi.acm.org/10.1145/2939912.2942354

[196] J. Eisses, "Anomaly detection in electricity consumption data of buildings using predictive models," University of Amsterdam, 2014, p. 17.

[197] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Moving object detection via robust low rank matrix decomposition with irls scheme," *Advances in Visual Computing*, pp. 665–674, 2012.

[198] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *CoRR*, vol. abs/0912.3599, 2009. [Online]. Available: http://arxiv.org/abs/0912.3599

[199] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and

automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[200] Q. Ke and T. Kanade, "Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 739–746.

[201] R. Gnanadesikan and J. R. Kettenring, "Robust estimates, residuals, and outlier detection with multiresponse data," *Biometrics*, vol. 28, no. 1, pp. 81–124, 1972. [Online]. Available: http://www.jstor.org/stable/2528963

[202] T. Bouwmans and E. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2013.11.009

[203] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2080–2088.

[204] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *CoRR*, vol. abs/1010.4237, 2010. [Online]. Available: http://arxiv.org/abs/1010.4237

[205] C. Guyon, T. Bouwmans, E.-h. Zahzah *et al.*, "Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis," *Principal Component Analysis, P. Sanguansat, Ed*, 2012.

[206] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low rank matrix factorization including spatial constraint with iterative reweighted regression," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2805–2808.

[207] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *Trans. Img. Proc.*, vol. 20, no. 12, pp. 3419–3430, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TIP.2011.2156801

[208] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, Aug 2012.

[209] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *in International Conference on Machine Learning*, 2011.

[210] Z. Wang and X. Xie, "An efficient face recognition algorithm based on robust principal component analysis," in *Proceedings of the Second International Conference on Internet Multimedia Computing and Service*, ser. ICIMCS '10. New York, NY, USA: ACM, 2010, pp. 99–102. [Online]. Available: http://doi.acm.org/10.1145/1937728.1937752

[211] X. He, S. Yan, Y. Hu, P. Niyogi, and H. jiang Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 328–340, 2005.

[212] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[213] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997. [Online]. Available: http://dx.doi.org/10.1109/34.598228

[214] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, no. 2, pp. 217 – 235, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000000917112

[215] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[216] P. Rodriguez and B. Wohlberg, "A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," *2014 IEEE International Conference on Image Processing, ICIP 2014*, pp. 3414–3416, 01 2015.

[217] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20 – 30, 2006, special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379505003812

[218] P. Jokar, N. Arianpoo, and V. Leung, "Electricity theft detection in ami using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, pp. 1–1, 05 2015.

[219] M. PeÃśa, F. Biscarri, J. I. Guerrero, I. Monedero, and C. LeÃşn, "Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach," *Expert Systems with Applications*, vol. 56, pp. 242 – 255, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416300938

[220] R. Moghaddass and J. Wang, "A hierarchical framework for smart grid anomaly detection using large-scale smart meter data," *IEEE Transactions on Smart Grid*, vol. PP, pp. 1–1, 04 2017.

[221] D. Li, K. Sawyer, and S. Dick, "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)*, Aug 2015, pp. 1–5.

[222] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York, NY, USA: ACM, 2014, pp. 80–89. [Online]. Available: http://doi.acm.org/10.1145/2674061.2674064

[223] K. D. Anderson, "Non-Intrusive Load Monitoring: Disaggregation of Energy by Unsupervised Power Consumption Clustering," 12 2014. [Online]. Available: https://kilthub.cmu.edu/articles/Non-Intrusive_Load_Monitoring_Disaggregation_of_Energy_by_Unsupervised_Power_Consumption_Clustering/6720851