AI Forecasting of Dissagregated Demand

Neil Darragh

Master of Science
Data Science
School of Informatics
University of Edinburgh
2021

Abstract

We present a novel approach to the application of Demand Side Response (DSR) and the associated requirement for short term load forecasting (STLF). Unlike previous approaches forecasting either household level or grid level aggregated load, we forecast the aggregated power load but dissagregated by appliance type. Utilising the REFIT dataset, we choose three appliances of interest - refrigeration, dishwasher and washing machine and for each, train deep-learning LSTM-based forecasting models to point forecast communal power consumption in 30 minute intervals over a 24 hour forecast horizon. We find we can learn to forecast the power load variations that are unique to each appliance type and which vary according to both behavioural and environmental factors. However, evaluating the accuracy of our models on a day to day basis using traditional regression metrics such as MAPE is challenging. We show this is due to the Aggregation Error [1] which dominates because we are in the "scaling regime" with the limited number of appliances we have in each aggregation. For the first time in the literature we report an Aggregation Error Curve (AEC) at the appliance level and show, via an appliance simulation, the level of aggregation that would be needed to reduce this error to below that of the model error.

Acknowledgements

I would like to express my gratitude to my supervisors, Dr Nigel Goddard and Dr Sasa Djokic for their time, guidance and support throughout this project. Additionally, I would also like to thank Mingzhe Zou for the many helpful discussions and suggestions regarding the deep-learning implementation. Finally, I would also like to thank my family for their understanding, love and support throughout this period.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Neil Darragh*)

Contents

Chapter 1

Introduction

Demand Side Response (DSR) is a smart grid technology that enables electrical loads to be advanced or deferred to reduce peak demand on the electrical grid or shift the load to a period of lower cost electricity generation (perhaps when there is an abundance of renewable energy available) [2]. In the domestic setting, examples of such loads are electric vehicle charging, dish washing, clothes washing and drying and refrigeration. The expectation is that occupants in a smart-grid enabled home will willingly allow a change of schedule of exactly when these appliances draw their power load (over a short term horizon of a few hours - perhaps overnight or during the workday) in return for reduced energy costs [3]. The societal benefit is a reduction in the environmental impacts of fossil fuel based energy generation and the construction of the associated infrastructure.

For DSR to work successfully, accurate forecasting of the anticipated short-term load available to be shifted is required. In the DSR marketplace, this typically involves making a 24-hour ahead forecast so that an Aggregator (a commercial enterprise that organises DSR products[3]) can offer a specific quantity of load shift for a specific period of time in the next 24 hours to the grid operator [4]. For practical and efficient management, the DSR market is only available for Aggregators able to control loads above a certain size and within a specific timeframe (currently, in the UK this is 3MW of capacity and the ability to respond within 20 minutes [5]). While there has been successful commercial sector deployment [6, 7, 8], such requirements have so far made it impractical for DSR to be employed at the domestic level despite this being a sizeable portion of the overall energy market - 30% and growing [9]. In the smart grid future, however, Aggregators operating in the domestic sector are expected to emerge, enabled by advances in IOT connected appliances and appliance monitoring and control infrastructure [3].

Various approaches to the short-term load forecasting and DSR control requirement have been proposed. The first is to use non-intrusive load management (NILM) to dissagregate the whole-house (smart meter) power signal into individual appliance power loads and then utilise forecasters for each to make a by-household, by-appliance prediction of when each appliance of interest would be in use by that household. These forecasts are then aggregated by the Aggregator who then makes decisions and asserts control over the appliances centrally [10, 11, 12]. The second approach also utilises by-household by-appliance forecasters but the control decision of which loads to shift is made locally by a Home Energy Management System (HEMS) responding to a more generic load shift request from the Aggregator [13, 14]. Yet others have suggested that the entire demand response could be voluntary - occupants responding to ad-hoc requests from the utility directly or responding to economic incentives in how their tariff is structured [15, 16].

We propose a different approach, which to the best of our knowledge is unique. We propose a single forecast model by appliance type, forecasting the aggregated load of that appliance across a community of homes. DSR control could then be asserted centrally by the Aggregator for that appliance type across the community. The advantages of this approach are that while the data collection requirements remain the same, only one forecaster is required (per appliance type), substantially reducing the compute requirements (indeed, some have suggested a forecasting model per appliance, per home is impractical [17]). Additionally, a per-household HEMS would no longer be required, and, based on previous work, forecasting an aggregated signal can be done more accurately than forecasting dissagregated signals [18]. We believe our approach offers a more cost effective, practical and accurate approach compared to the previously suggested solutions.

In this work we explore the ability to forecast short-term aggregated load, disaggregated by appliance type to facilitate this approach.

The novel contributions of this work are:

1. Using the REFIT dataset [19], we build and train deep-learning models to forecast aggregated power demand, disaggregated by appliance type, which, to our knowledge, has not been reported on previously.

2. We demonstrate that we can sufficiently forecast for the application, the power load driven by the first level of various communal behavioural and environmental

factors identified in our EDA work.

3. We compare our forecasting results to previously reported works regarding household level and grid level short term load forecasting, and demonstrate that these initial aggregated appliance-level results, as with household level forecasting, are dominated by Aggregation Error [1] rather than model forecasting error.

4. We present new results showing an appliance-level Aggregation Error Curve (AEC) and determine empirically how many appliances are needed in a refrigeration aggregation to move from the "scaling regime" to the "saturation regime" (as defined by [1]) thereby removing the aggregation error from the reported metrics.

5. We present a successful initial forecasting framework that we believe enables a more practical domestic demand side response implementation than previously suggested approaches.

This dissertation is presented in four main sections. In chapter 2 we summarise the previous approaches to load forecasting, identifying the methods and practices most appropriate to our application. We identify and analyse candidate datasets for our investigation, select the one most applicable and perform exploratory data analysis. In chapter 3 we present the methodologies we adopt to prepare the data for the task, the evaluation metrics we will use, establish naive baselines against which we will evaluate our models and present details of the forecasting algorithm. In chapter 4 we present our experimental results, provide a detailed analysis and compare to existing literature. Finally, chapter 5 summarises our findings and presents guidance on further work.

Chapter 2

Background

2.1 Previous work

We draw on the existing literature regarding short-term load forecasting (STLF) at various levels of aggregation to understand what the state-of-art forecasting methods are and the results they achieve.

STLF at the grid level is a long established discipline [20, 21]. Researchers are motivated to produce more accurate forecasting models which can be used by the grid operators to better forecast power needs over the coming hours and bring dispatchable generation onto the grid in time for when it's needed. Recent state-of-art approaches employ a variety of deep learning architectures and can achieve accuracies, as reported by Mean Absolute Percentage Error (MAPE) in the 1%-2% range. For example, with Recurrent Neural Networks (RNN) [22], Long Short Term Memory (LSTM) [23], Convolutional Neural Networks (CNN) coupled with LSTM [24] and LSTM coupled with Gated Recurrent Unit (GRU) [25] which, as an exemplar, achieved a best-in-class forecasting error of 1.85% on the EUNITE load forecasting competition dataset.

Recently, household-level STLF has become an extremely popular area of research with the proliferation of data from smart metering infrastructure becoming available publicly. The stated motivations for these studies are various. Some researchers are motivated in the belief that by providing home occupants with a forecast of their upcoming energy use, it will motivate them to make voluntary behavioural changes to manually shift it to some time later to take advantage of a cheaper rate [26, 27]. Others have argued explicitly or implicitly it will be needed for a locally-managed DSR implementation (such as a Home Energy Management System) [28, 29, 30]. Others have argued it enables utilities to target high-use customers for DSR [31, 32] while others have argued it will improve grid-level forecasting [33]. And yet others motivations are just to improve household level STLF forecast accuracy without any other stated motivation [34, 35]. While some of these motivations may have some validity, we argue the explosion of research using this data is largely because of the availability of the data itself rather than it necessarily addressing a specific problem. One area, however, where this data has been invaluable is in the development of Non-Intrusive Load Monitoring (NILM) which is the ability to dissagregate a household power signal (post-hoc or in real time) into its constituent appliance components [36].

Contrary to the accuracies achievable with grid level STLF, reported accuracies at household level STLF are much lower - typically in the 30%-45% MAPE regime. This is because the signal at the household-level is disaggregated compared to that at the grid level. Energy use at the individual household level is very variable - driven largely by the individual habits and behaviours of the occupants. Habitual behaviour (both from individuals or from the appliances themselves with pre-programmed schedules) leads to patterns of energy use which can be learnt and predicted. On the other hand, completely random behaviours are not learnable and cannot be predicted. Therefore, the achievable STLF accuracy of any particular home will largely be a function of that households habitual behaviours vs random behaviours rather than a function of any particular forecasting methodology (at the grid level, these random behaviours are "averaged out" of the data, leaving only the deterministic part and a large reason why the achievable forecasting accuracies are so much better).

Nevertheless, many papers have been published on this topic with increasingly complex deep-learning architectures. In [28] the authors employ bi-directional LSTM, RNN and GRU architectures, reporting a best-in-class MAPE of 35% on the latter. In [31] the authors report a MAPE of 44% using a 2-layer LSTM architecture. In [26] the authors report a 40% MAPE by combining an LSTM and CNN model compared to a MAPE of 44% with just the LSTM alone. In [17] the authors propose a CNN with pre-clustering of customers into similar profiles in order to reduce the number of forecasting models required in a practical DSR implementation and report a MAPE of 39%. In [33] the authors achieve MAPE's of 32% to 40% using an echo state network.

Almost all of the papers state they have achieved improved (sometimes, state-of-art) results by comparing their method to some hand-picked lesser method - often ARIMA, a traditional ML (Support Vector Regression or Decision Trees) or even another deep-learning approach. However, the accuracy of the underlying results are still always in the 30% - 45% range due to the random behavioural aspects already mentioned.

In this work we are motivated to demonstrate the validity of our new approach of forecasting aggregated appliance power, disaggregated by appliance type rather than choosing some overly complex model architecture with which to baseline it. Based on this outlook and our review of the related works we therefore choose a LSTM-based model architecture: It is popular in the most recent literature, straightforward to implement and evaluate and demonstrated to produce competitive accuracies on both aggregated and disaggregated loads. We should expect forecasting accuracies, as measured by MAPE, to be better than disaggregated household-level STLF state-of-art (approximately 30-45%) but no-better than aggregated grid-level STLF state-of-art (approximately 1-2%).

2.2 Task

Our task is to produce a 48-point forecast covering a forecast horizon of 24-hours at a resolution of 30 minutes of the aggregated power demand, by appliance type, using historical data and potentially additional explanatory variables.

2.3 Selection of Dataset

We analysed two datasets as candidates for this work - the IDEAL dataset [37] and the REFIT dataset [19] (there is also UKDALE [38] but as it only monitored 5 homes, we did not consider it further). Both studies instrumented 20 or more homes with appliance-level monitoring devices for different appliances within each home. A summary of the monitored appliances included in the IDEAL and REFIT datasets is shown in table 2.1. We have noted those appliances which we believe are DSR-eligible. We argue that each of these could have their load demand advanced or deferred with either no impact to the user or a minimal impact to which the user agrees to in return for reaping the benefits of participating in a DSR program.

Our task involves assembling an aggregated dataset for each DSR-eligible appliance type we wish to consider in our study. The aggregated data needs to be constructed coherently - that is, the period of time is chosen such that the same homes are consistently reporting data for that appliance *for the entire period.* We wish to capture collective appliance-use behaviour on a day to day basis. Therefore each home (and its appliances of interest), which will have a unique behavioural pattern, needs to be present for the entire period of our study. Apart from short periods (which we can deal

Appliance Name	REFIT Qty	REFIT Ave Days	IDEAL Qty	IDEAL Ave Days	DSR Eligible
Washing Machine	20	506	24	183	Yes
Microwave	17	490	28	152	
Fridge Freezer	16	504	28	183	Yes
Dishwasher	15	501	20	194	Yes
Kettle	15	493	28	174	
Freezer	13	519	6	139	Yes
Toaster	10	528	22	76	
Fridge	7	497	5	130	Yes
Tumble Dryer	7	519	3	126	Yes
Heater	4	516	5	64	
Washer Dryer	3	479	9	146	Yes

Table 2.1: Summary of the appliances available in both the REFIT and IDEAL datasets. Those considered DSR-eligible and therefore candidates for this study are identified.

with by imputation) appliances going permenantly offline or new homes being included in the study as it progresses are not behavioural changes and are not patterns (in the first instance) we wish to learn with our forecasting models.

Figure 2.1 shows an example appliance-level heat-map showing power data availability for dishwashers, one of the candidate DSR-eligible appliances available in both datasets. We observe that homes in the IDEAL study were added gradually throughout the study period vs REFIT. We found all the DSR-eligible appliances showed the same trends in both datasets (Appendix A).

We make the following observations regarding the suitability of the two datasets to the task.:

- The coherency of the REFIT dataset is more suited to our task - we need a long period of data coupled with as many appliances reporting over that same period as possible.
- Both datasets show random gaps in the data for individual appliances and also gaps in the data that appear to have affected all appliances. This latter issue is more prevalent in the REFIT data, particularly towards the end of the study (the horizontal white lines in figure 2.1).

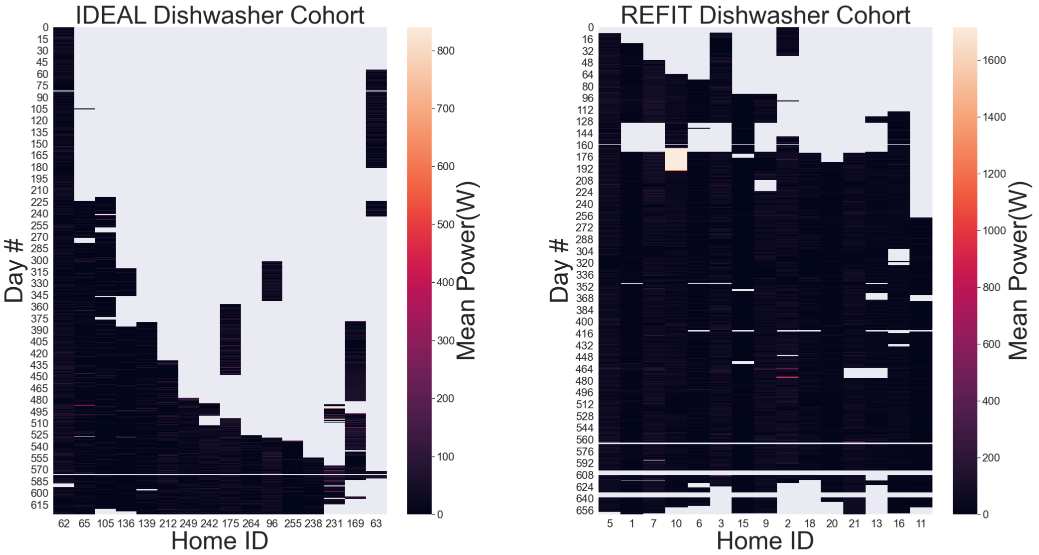


Figure 2.1: Coherency Analysis of an example DSR-Eligible Appliance - Dishwashers.

- The most populous appliances are refrigeration, dishwashers and washing machines in both datasets. The least are tumble dryers and washer-dryers.

Based on the longer period of coherency for a larger number of appliances for each DSR-eligible appliance type, we choose the REFIT dataset and the following appliances - Dishwashers, Refrigeration (all types) and Washing Machines as examples to study (in this study we are aiming to provide a proof of concept of the approach, not exhaustively study all the available DSR-eligible appliances).

2.4 Exploratory Data Analysis

2.4.1 Dissagregated Appliance EDA

2.4.1.1 Short Timeframe Load Profiles

Figure 2.2 shows an example of the power signal for the selected appliances from REFIT home 1 (we deliberately selected a day when all three were in use). We observe that the dishwasher and particularly the washing machine signals are quite noisy when in use. Both typically employ water heaters to heat the incoming mains water. In addition they both have water pumps, control electronics, sensors and in the case of the washing machine, employ a high-power motor to rotate the drum. All these components are

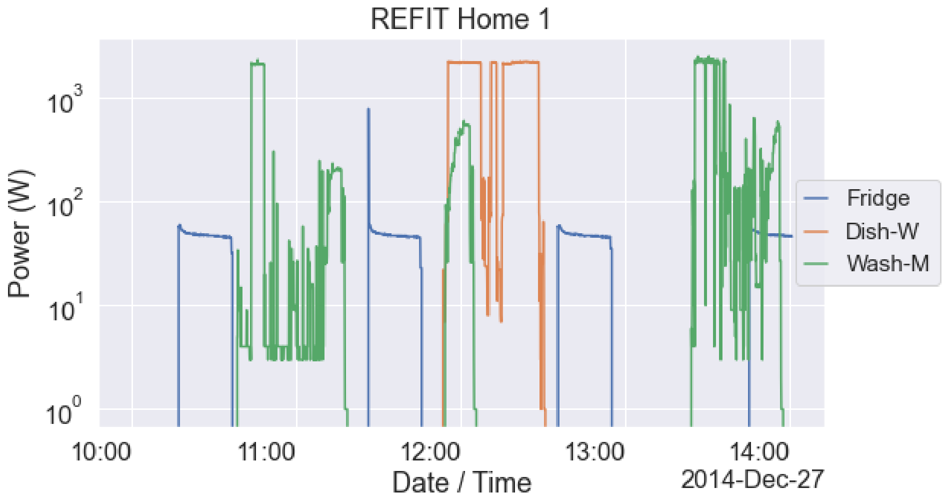


Figure 2.2: Example Daily Load Profile for Dishwasher, Fridge-Freezer and Washing Machine (a day and time span deliberately selected where they were all in use).

operated at various times during the operation of the appliances and are responsible for the complexity of the signals. The flat topped regions exceeding 1kW are likely the periods when the water heater was operating. The last few minutes of each washing machine cycle show a curved rising power profile - likely the washing machine in the spin cycle.

Refrigeration devices employ a compressor to compress a refrigerant which is then rapidly expanded to provide the cooling effect. A refrigerator / freezer is typically set to maintain a constant internal temperature while the temperature in the environment it is located in is changing (the kitchen, utility room or perhaps even an unheated out-building). The compressor runs to reduce the internal temperature of the fridge to a set level (with hysteresis) and then turns off until the internal temperature rises and triggers it again. We see this cycling effect in the plot of figure 2.2. In this example, each on period is approximately 20 minutes and each off period is approximately 1hr 15 minutes. The sharp spike at the beginning of the 2nd cycle is due to the high start-up current of the compressor. Usually only lasting a few seconds, this is randomly sampled by the power monitor. As the external temperature changes, the compressor has to come on more often to cool the appliance internally (due to thermal losses), leading to a higher duty cycle (on time vs off time) and more average power being consumed. When the compressor is running, the appliance in this example consumes approximately 80W.

2.4.1.2 Anomalous Appliance Behaviour

The raw datasets for the chosen appliances were analysed for anomalous appliance behaviour utilising timeseries plots to find periods of unusual temporal behaviour and boxplot distributions to identify unusual appliance power distributions. These plots and the investigations are documented in appendices B and C. The only issues we found that resulted in individual appliance removal from the dataset were that the appliance labelled as a freezer in home 13 was very unlikely to be correct. Furthermore, we determined that homes 1 and 7 from the freezer dataset reported no active power values, only zero's throughout and were also removed. While they weren't affecting any aggregated profile, their presence did affect the computation of the mean power per appliance in the aggregation.

We note in figure 2.1 the white lines across all appliances in the dishwasher dataset. These were present in all the REFIT appliances at the same times and are periods where there is no data reported at all, or, in the case of a few appliances, data which is "stuck" at the last observed value just before the same period. We term these as periods of "system-wide failure" and we will deal with them with a detection, cleaning and imputation strategy to be described later.

2.4.2 Aggregated Appliance EDA

We turn now to looking at the aggregated loads of the appliances. For each appliance, we sum the load across each identical 30-minute interval for all dwellings and then compute a simple mean by dividing by the number of dwellings (a constant for each appliance type).

Figure 2.3 shows the mean power load of each appliance over two different timescale views - hour of day (by day of week) and month of year. In each plot, we pivot on the calendar / time variable and compute the mean value in that interval for all the data. We observe there is a distinct pattern of use for each of the appliances over the different timescale views.

Dishwasher load is lowest around 5am, rising sharply in the morning, showing significant peaks at 8am, 7pm and two more lesser peaks at 1pm and 1am. We note that the behavioural use of dishwashers by hour is markedly different at weekends vs weekdays. Specifically, at weekends the morning peak starts later and the use is more spread out over the course of the day. We argue that this is consistent with expected household behavioural differences at the weekends vs weekdays as occupants engage in

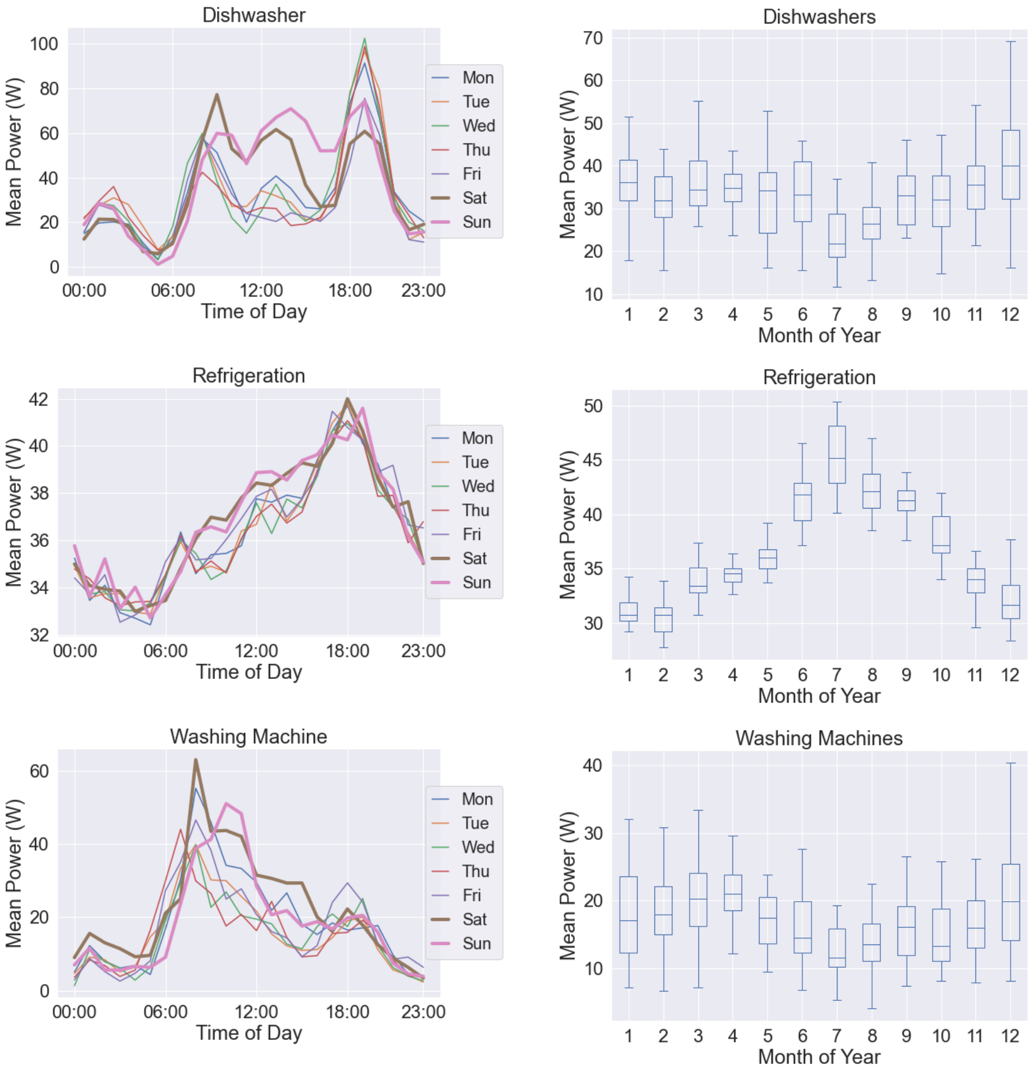


Figure 2.3: Analysis of selected appliances. Left: Daily load profiles by day of week. Right: Hourly power distribution by month.

leisure vs work / school routines during those times, respectively. Over the course of the year dishwasher load is lowest during the summer months and highest in December. This could be behavioural or perhaps more likely due to the incoming water temperature (which the appliance must heat to a target internal temperature) which varies over the year according to the annual seasons.

Refrigeration load rises from its low at 5am, peaking at around 6pm before declining again over the evening and overnight. We see 3 small disturbances in load around 7am, 12-1pm and 6pm which we ascribe to behavioural factors - increased door opening activity and perhaps loading of warm items to be cooled, all around common mealtimes in the UK. During the week, the average load remains fairly constant, even at weekends although we see perhaps some evidence of spread out use at the weekend in the daily load profile (the two highest lines between the morning and evening peaks are the two weekend days). Over the course of the year refrigeration load peaks approximately 40% higher in the summer months vs winter coinciding with peak vs low environment temperatures.

The peak of washing machine load is the morning hours, from 8-9am with a lesser peak around 6-7pm. Over the course of the week, the load varies somewhat, peaking on Saturdays, Mondays and Sundays respectively. As with dishwashers, we see a later morning peak on Sunday and also that the load is lowest during the summer months providing perhaps further evidence that the higher incoming water temperature leads to less heating power being required (although behavioural factors can't be eliminated entirely based on this observation alone).

Overall we see three seasonalities in the data - daily, weekly and annually, which we infer with domain knowledge (we only have just over 1 year of data but we infer some of the change over the year to be the effects of environment temperature on the power required to refrigerate or heat water). Note that we adopt the terminology defined in [39] where seasonality refers to any recurring pattern with a fixed frequency, explicitly distinguishing it from cycles and trends.

Some of the features we observe are due to behavioural patterns and others are due to environment (temperature). It's perhaps important to note that the behavioural patterns are likely to be community and culturally specific. The REFIT study collected data from 20 homes in the Loughborough region of the UK. The UK has a typical north European climate, a particular school/working week vs weekend cultural schedule and particular cultural and public holidays. Other communities and climates would perhaps have their own unique factors that would drive their behavioural use of particular appliances differently. Indeed, the actual appliances used themselves may be quite different from different communities and cultures. This in itself presents an interesting area of study, perhaps leading to ways of implementing transfer learning between culturally similar communities separated geographically, or, to simply increase the level of aggregation. This, however, is not the focus of our initial study and we do not consider it further here.

Figure 2.4 presents a visualisation of the daily load profile for each appliance over the course of a year. Presenting the data as an image this way allows us to more easily spot patterns and trends and how they change over the course of a year, Firstly, we see the peaks in use at certain times of day for each appliance as horizontal lines of lighter colours.

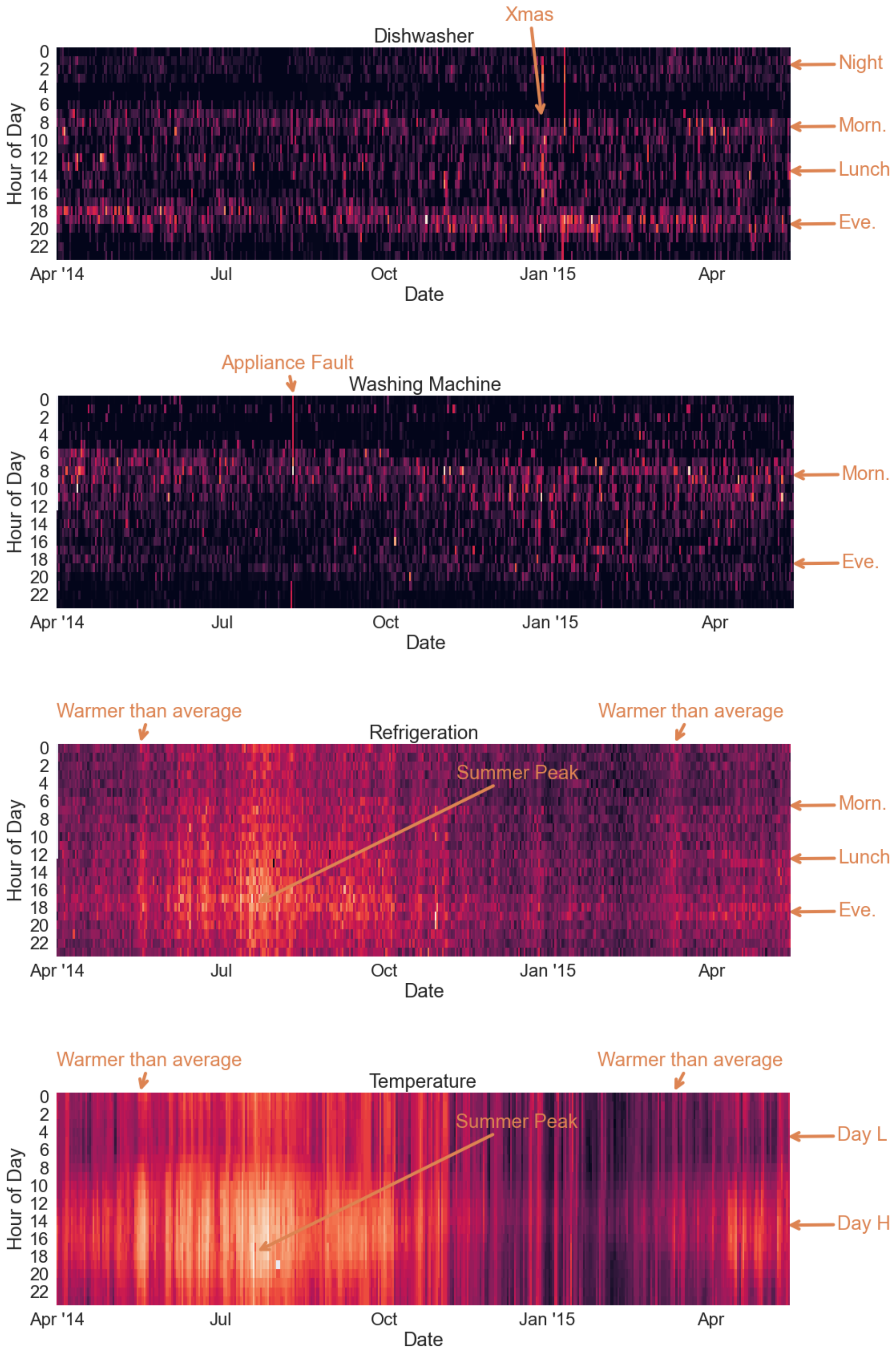


Figure 2.4: Heatmaps of appliance daily load vs date.

Periods of high use on particular dates show up as bright vertical lines in figure 2.4. We note the two bright lines in dishwasher use correspond to December 25th and, interestingly, Monday January 5th 2015 into Tuesday January 6th. The high use of dishwashers on December 25th is understandable given that it's Christmas Day and is typically celebrated culturally in the UK as a feast. Investigation revealed January 5th/6th was found to be an excessive use from a single home. Washing machines show a particularly high power on August 8th which corresponds to an appliance "fault" anomaly we identified in the appliance-level EDA work (appendix B). Refrigeration shows many vertical (somewhat wider) lines spread throughout the year and also variation over the course of the day forming some quite bright regions. We find these observations to be explainable by temperature - both as it varies throughout the day and how it varies throughout the year. The bottom image in figure 2.4 is a heatmap of the half-hourly temperature for the Loughborough area over the same period as the REFIT data collection [40]. One can see qualitatively how well correlated they are in terms of the bright regions and bright vertical lines (the Pearson cross-correlation coefficient is 0.62).

Figure 2.5 shows the daily load profiles for different types of days that we felt might be explanatory for differences in power load: weekday, weekend day, public holiday and school holiday. The Pearson cross-correlation matrix between the daily load profiles is shown in the lower figures. We see that for dishwashers, public holidays are highly similar to weekend days and dissimilar to weekdays and school holidays (which look the same). Refrigeration load shows very little sensitivity to day type and they are all very highly correlated to each other. Washing Machine load doesn't change much between day types - the highest correlated are school holidays and week days which at 0.97 are essentially identical. The least correlated are weekdays and public holidays at 0.83 which is still quite high - we note the range of differences in correlation coefficient is quite small. We conclude that weekdays, weekend days and public holidays are useful markers, depending on appliance. School holidays do not appear to be useful markers.

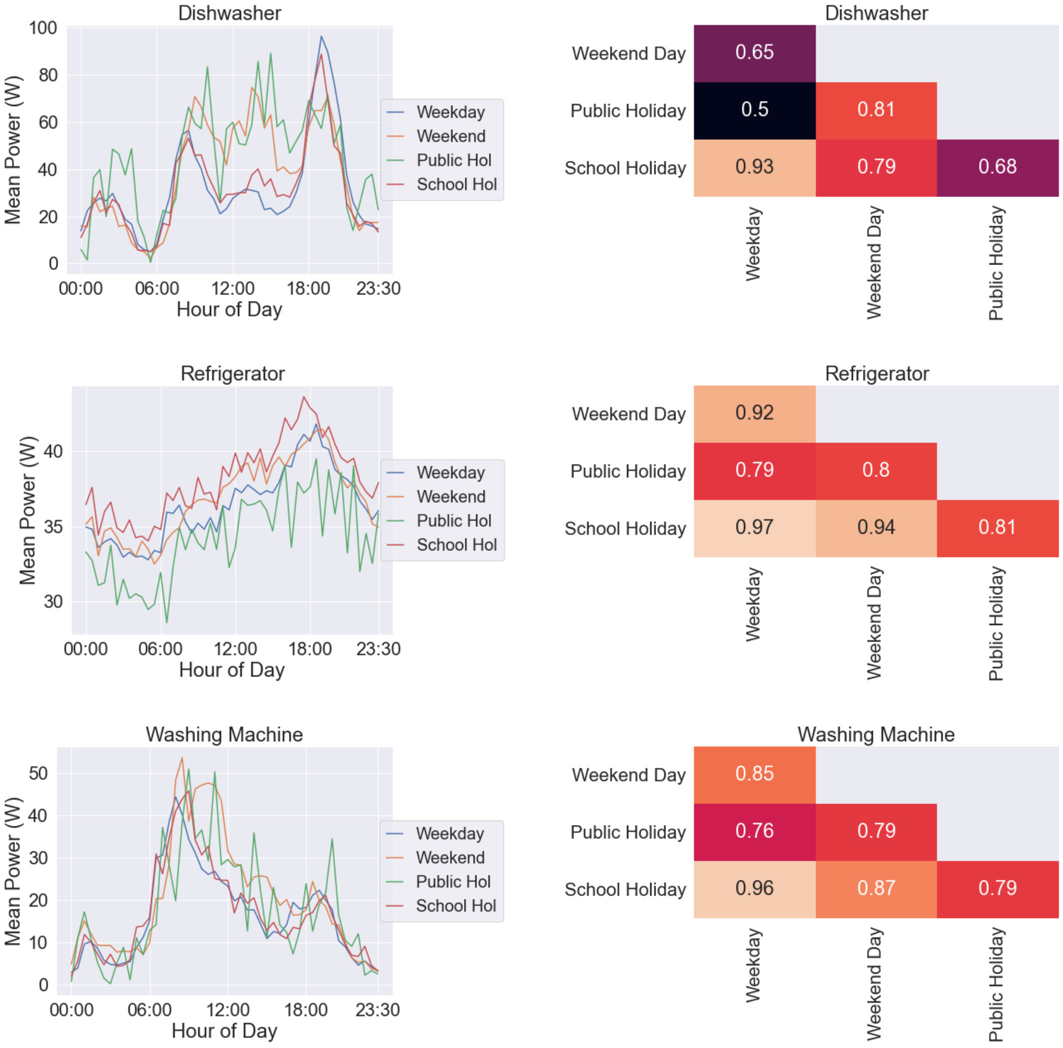


Figure 2.5: Public holidays and school holidays vs non-holiday weekdays and weekend days.

Chapter 3

Methodology

3.1 Data Preparation

3.1.1 Cohort Analysis and Assembly

For each selected appliance, we wish to assemble a cohort of homes from which to generate the aggregate signal. We apply the following criteria to determine which homes and how much data from each to include in the cohort assembly:

- The span of time must be at least 1 year to capture at least one example of expected annual seasonality effects.
- The number of appliances included must be as large as possible to smooth out randomness in individual household behaviour.
- Aside from short periods (which shall be imputed), a cohort must have all appliances reporting data for the entire duration of the cohort - it must be coherent.

We implemented a cohort assembly algorithm conforming to these requirements. The algorithm and the selection of homes it picked for each appliance aggregation are detailed in Appendix D. All fridges, freezers and fridge-freezers were merged into one appliance type - refrigeration. The data were re-sampled from the original REFIT datafiles to down-sample them from the original stochastic sampling process (sample rate was approximately 8 seconds, but variable) to a specific alignment (on the half hour) and a fixed period of 30 minutes. The value we compute is the mean of the power measurements recorded in each 30 minute interval. A summary of the cohort assembly process for the three selected appliances is shown in table 3.1.

	DISHWASHERS	REFRIGERATION	WASHING_MACHINES
NUM APPLIANCES	14	31	19
START DATE	2014-03-31	2014-04-02	2014-04-02
END DATE	2015-05-10	2015-05-10	2015-05-10
NUM DAYS	405	403	403
NUM INTERVALS	19455	19366	19366
PCT IMPUTED	3.27	3.25	3.3

Table 3.1: REFIT Appliance Cohort Assembly Summary

3.1.2 Outliers, Anomalies and Missing data

As noted in the appliance-level EDA, a number of issues were identified which need to be addressed before our data can be used for the task.

- We devised an algorithm to detect the region of system-wide failure identified in section 2.4.1.2 and deleted all values within those regions for all appliances. Details of the algorithm and the regions detected and deleted are shown in Appendix F.
- We applied a 99.9% quantile limit (computed separately for each appliance) to the refrigeration data to remove extreme outliers due to compressor start-up transients.
- After aggregation we imputed the missing values from the periods of system-wide failure from the same time interval and day from the week before. A summary of the imputation process is shown in table 3.1.

3.1.3 Train, Validation and Test Split

The dataset for each aggregated appliance was split into training, validation and test sets as summarised in table 3.2. The test-set was held back and not used for any purpose except evaluating finalised models. We chose to maintain strict temporal ordering in our dataset (and not shuffle or use cross-validation) as in a real-world implementation we would wish the model to adapt over time to behavioural or appliance changes which occur at specific dates and remain so thereafter.

SET	START DATE	END DATE	DAYS	INTERVALS	%
TRAIN (DW)	31/03/2014	11/30/2014	245	11760	60%
TRAIN (R & WM)	02/04/2014	30/11/2014	243	11664	60%
VALIDATION SET	01/12/2014	30/01/2015	61	2928	15%
TEST SET	31/01/2015	10/05/2015	100	4800	25%

Table 3.2: Train, Validation and Test Set Constituency

3.1.4 Normalisation

The data were normalised using min-max scaling as computed on only the training period - finding the offset and scale values that convert the training data to values between 0 and 1 and then applying these to all data values including the validation and test periods.

3.1.5 Conversion to Supervised Dataset

Figure 3.1 shows the formation of a single training sample at time t. Each training sample input features are formed of N previous (lagged) observations and (optionally) a one-hot encoded label for each day of the week. The labels are the future 48 intervals from t. During training, t is advanced 30 minutes (1 interval) for each training sample, generating 11664 training samples for dishwashers and washing machines (refrigeration is 11760). The validation set is also evaluated in 30 min intervals (2928 samples). For testing, t is advanced 48 intervals (1 day) at a time since we only compute one 24-hour forecast for each day at 6am and then roll forward to 6am the next day to simulate the application in real-world operation (running a single forecast each day to provide the data for a DSR bid to the grid operator). Care was taken to ensure that there was no overlap of training samples or their associated labels from the validation and training sets.

3.2 Evaluation Metrics

Model forecasting performance will be quantified in terms of how well it predicts the power signal for each of the 30 minute intervals over the forecast horizon of 24 hours from 6am each day. For each set of 48 intervals over a single forecasted 24-hr period we

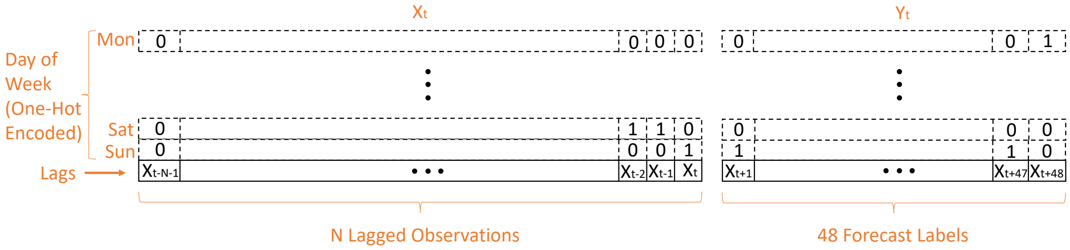


Figure 3.1: Conversion to a supervised dataset for model training.

compute the most commonly reported metrics in the literature, RMSE (eqn 3.1), MAE (eqn 3.3) and MAPE (eqn 3.4) between the forecasted signal and the actual signal.

These measurements are then repeated for each day in the test period. We summarise our overall performance by averaging each metric over all the days in the test (or validation) period as the example for RMSE is shown in eqn 3.2. We will use the average RMSE computed over the entire validation set as the means to select hyper-parameters and save the weights from the best observed model. We will also use the regression metrics computed over the entire test set to compare our results to the existing literature.

3.2.1 Root Mean Square Error (RMSE)

RMSE for a single test day is given as:

$$RMSE_d = \sqrt{\frac{1}{48}\sum_{i=1}^{48}(\widehat{y_{di}} - y_{di})^2} \tag{3.1}$$

where i is the interval in the forecast horizon, d is day in the test period, $\widehat{y}$ is the interval forecasted power, y is the actual interval power. RMSE for the entire test period is then:

$$RMSE = \frac{1}{D}\sum_{d=1}^{D} RMSE_d \tag{3.2}$$

where D is the total number of days in the test period.

3.2.2 Mean Absolute Error (MAE)

$$MAE_d = \frac{1}{48}\sum_{i=1}^{48}|\widehat{y_{di}} - y_{di}| \tag{3.3}$$

	Dishwashers	Refrigeration	Washing Machines
Num Weeks	19	6	15
Validation Set RMSE	54.47	6.04	29.45

Table 3.3: The optimal number of weeks to compute the Recent Day of Week & Interval of Day Means over for the Naive Baseline.

3.2.3 Mean Absolute Percentage Error (MAPE)

$$MAPE_d = \frac{100}{48} \sum_{i=1}^{48} \frac{|\widehat{y_{di}} - y_{di}|}{y_{di}} \tag{3.4}$$

3.3 Naive Forecast Baseline

It's important in any machine learning task to establish a naive forecast baseline against which the forecasting results of candidate models can be compared. In our dataset, which we have shown to have daily, weekly and, by inference, annual seasonality, there are a number of season-aware naive forecasts we could consider for each forecast interval of the forecast horizon. We note that while it is likely our dataset has an annual seasonality, it is not possible to construct any annual seasonal naive forecasts as we do not have data spanning previous multiple years to compute such a forecast from. We therefore chose a seasonal naive forecast which is the recent mean of the same interval of day from the same day of the week, going back several weeks. The naive forecast was optimised by sweeping the number of previous weeks to compute the mean over, monitoring the RMSE on the validation set and using the value that yields the lowest value for each appliance. The findings are summarised in table 3.3.

Full details of all the naive forecasts we considered and their evaluation results on the test period is given in Appendix G.

3.4 Forecast Model

The LSTM architecture is well suited to time-series forecasting problems as it directly encapsulates the ordered, temporal nature of the input data [41, 42, 43]. The architecture of our particular implementation is shown in figure 3.2.

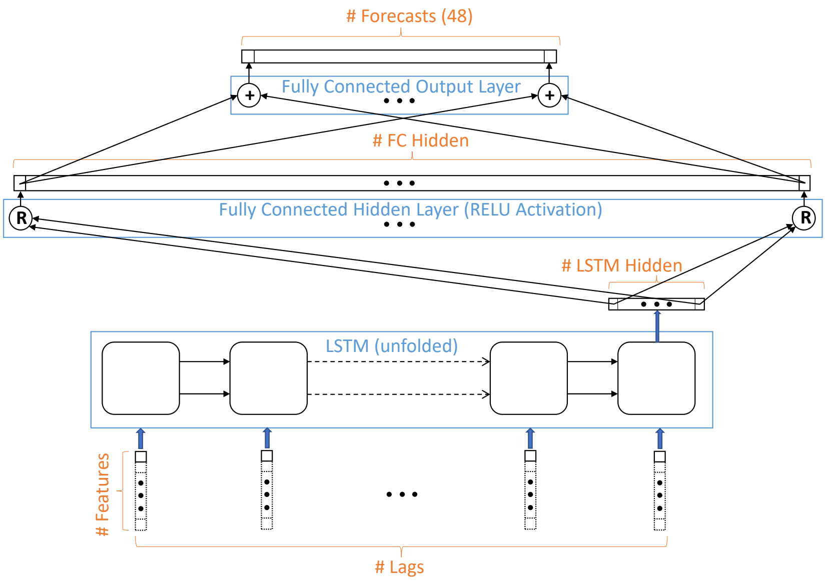

# Forecasts (48)
Fully Connected Output Layer
+
# FC Hidden
Fully Connected Hidden Layer (RELU Activation)
R
# LSTM Hidden
LSTM (unfolded)
# Features
# Lags

Figure 3.2: Model Architecture

3.4.1 Implementation Details

The architecture and setup parameters common to all the forecasting models are shown in table 3.4.

3.4.2 Hardware / Software

We utilised a Windows based PC with the following hardware and software to perform all the experiments in this report. Processor: Intel 16-core i7 10700K 3.8GHz, GPU: NVIDIA GeForce RTX3080, Development Environment: Jupyter Notebooks running Anaconda 4.10.1 for Windows (10), Deep-Learning Library: Tensorflow-gpu v2.5.0.

3.4.3 Hyper-Parameter Tuning

For hyper-parameter tuning we adopt the methodology described in [46] of tuning the hyper-parameters using random search. Such an approach allows a larger area of the parameter space to be searched per compute resource unit. Table 3.5 shows the possible values for each of the hyper-parameters being tuned. For each trial, we sample a value for each with uniform probability from each list.

The efficacy of each combination of hyper-parameter values are quantified by training the models on the training set and evaluating them using RMSE evaluated on the validation set. At the end of each epoch (one complete pass of the training data

PARAMETER	VALUE
LSTM LAYERS	1
LSTM ACTIVATION	RELU [44]
FULLY-CONNECTED (FC) HIDDEN LAYERS	1
FC HIDDEN ACTIVATION	RELU
BATCH SIZE	250
LOSS FUNCTION	MSE
OPTIMISER:	ADAM [45]
ADAM LEARNING RATE	1E-3
ADAM LEARNING RATE DECAY	NONE
EARLY-STOPPING PATIENCE	100

Table 3.4: Common model architecture and configuration

PARAMETER	VALUES
LAGGED OBSERVATIONS	6, 12, 24, 48, 96, 192, 384
LSTM HIDDEN SIZE	32, 64, 128, 256, 512, 1024
FULLY-CONNECTED HIDDEN SIZE	32, 64, 128 ,256, 512, 1024

Table 3.5: Random hyper-parameter search values

through the model) the model is used to compute the forecast interval squared error for each interval in the validation set. The square root of the mean of these is then computed for a single evaluation metric for that epoch to yield its RMSE. If the RMSE is the best yet seen during training, the model weights are saved to disk. Training then continues through multiple epochs (typically 100-200) whereby at the end of the training we have a best-seen RMSE and model saved to disk for that particular set of hyper-parameters. Furthermore, we monitor the validation RMSE and employ an early stopping strategy as recommended by [47]. If, after 100 epochs (the "patience"), the validation RMSE has not improved from the best yet seen, we terminate training. This strategy allows more combinations hyper-parameter settings to be evaluated efficiently within our available compute resource by reducing our time over-training our models. This procedure is then repeated for all the hyper-parameter settings to be evaluated. We select the hyper-parameters that yielded the lowest RMSE on the validation period for final model evaluation on the test set.

3.4.4 Model Evaluation

After hyper-parameter tuning we evaluate trained models using the selected hyper-parameters on the held-out test period. First, we retrain ten models with different known seeds (1000 to 1009, step 1), each time using the minimum RMSE on the validation period as a means to save the best model weights and terminate with early-stopping. Finally, we run the held-out test set through each of the trained models and report the metrics along with standard deviations to quantify the repeatability of the model training process.

Chapter 4

Experimental Results and Discussion

Our EDA work informs us the appliances we have chosen have different power signatures over the three seasonalities and it is likely that they will need different treatment when building and evaluating our forecasting models. We will consider each separately.

4.1 Refrigeration Forecasting

We noted from our EDA that we have two strong seasonalities in our refrigeration data - a daily and an annual one, both of which are strongly correlated to temperature. Furthermore, we also detected an additional behavioural pattern in the daily cycle - that of increased power consumption around mealtimes as users interact with the fridge.

4.1.1 Hyper-parameter Tuning and Test Results

The results of the random hyper-parameter search as evaluated on the validation period are summarised in table 4.1. Here we are showing the top performing configuration (by RMSE) in 65 random trials which took approximately 18 hours to complete. Figure 4.1 shows a typical training / validation curve where we can see, in this case, the model is under-constrained - there is a point of best validation RMSE after which the validation RMSE worsens as the model starts to over-fit to the training data.

The result of the test set evaluation of the best performing model from the hyper-parameter search is shown in table 4.2. We see that the model has beaten the Naive baseline for RMSE and MAE. improving it by approximately 10% in each case and MAPE by 9.3%. The standard deviation over the 10 retrained models with different seeds is very small (<0.55%) indicating that the model training process with this

TRIAL RANK	NUM LAGS	LSTM HIDDEN	DENSE HIDDEN	VALIDATION RMSE	BEST EPOCH	FIT TIME (SECS)
1	192	32	512	5.340	317	5010

Table 4.1: Refrigeration - Best performing settings from random hyper-parameter search trials.

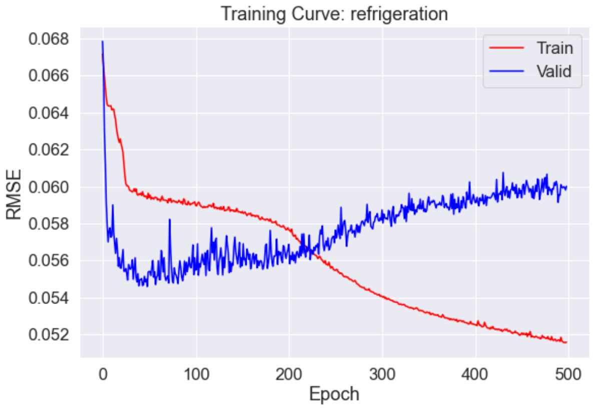


Figure 4.1: Example training and validation loss (MSE) curve showing the model, in this case, to be under-constrained (early-stopping is disabled in this example).

particular set of hyper-parameters is quite repeatable.

EVALUATION METRIC:	RMSE	MAE	MAPE(%)
NAIVE BASELINE	5.820	4.695	14.488
FORECAST MODEL 10X MEAN	5.230	4.192	13.140
FORECAST MODEL 10X STDDEV(%)	0.24	0.19	0.54
VS BASELINE	-10.1%	-10.7%	-9.3%

Table 4.2: Refrigeration Model vs Naive Baseline Test Period Evaluation

Figure 4.2 (left) shows an example forecasted day (the first day of the test period). The 95% confidence interval for each forecast interval is computed from the residuals of the training period as per [39]. We observe that the confidence intervals remain somewhat constant vs the forecast over the forecast horizon (tracking with the forecast, but not diverging). This is one of the benefits of a non-linear model over a linear model, such as ARIMA, where there the confidence interval always grows over the forecast horizon due to having to bootstrap previously forecasted values to get the next interval prediction [39]. We observe in figure 4.2 (left) how erratic the actual signal is over the forecast horizon. On the other hand the forecast is quite smooth and the naive baseline is somewhat in between. Comparing the forecast to the expected daily load profile that we determined in the EDA work (figure 2.3) we can see that the model forecasts a peak around 6.30-7pm and there is some evidence of increased power at around 7am and 1pm (although, it's not strong). While the actual signal and naive baseline clearly show the daily seasonality also, they seem too erratic to make out the mealtime behavioural peaks at all.

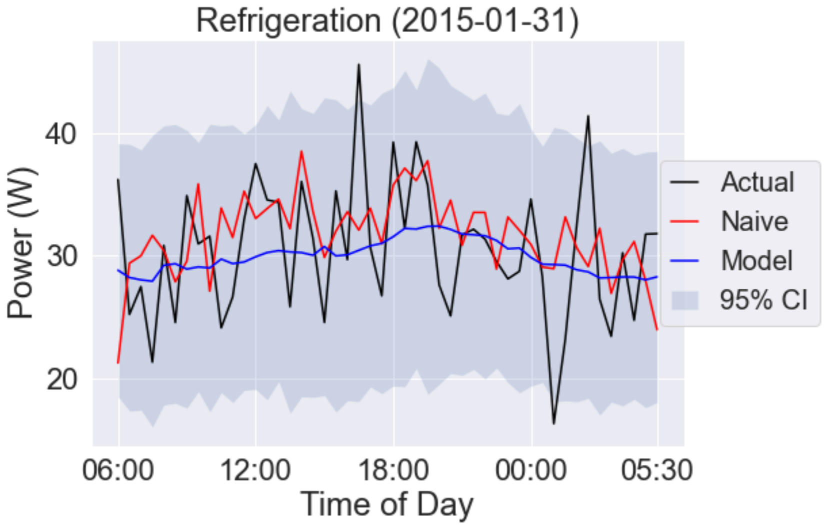


Figure 4.2: Model Forecast, Naive Baseline Forecast and Actual over the forecast interval for the first day in the test period. The RMSE between the forecasts and the actual signal for this day is 5.33W vs 5.93W for the model and naive baseline respectively.

The left and right plots of figure 4.3 show the MAE and MAPE respectively over the forecast interval for the same forecasted day as figure 4.2. Of course, the MAE and MAPE values reflect the erratic behaviour of the actual signal vs forecast throughout the forecast horizon but the main thing to note here is how flat the overall trend is - while the underlying signal against which MAE and MAPE are both computed rises and then falls over the daily cycle (figure 4.2), the errors (apart from some individual peaks) do not - the forecast, naive baseline and actual signal all follow the same daily seasonality and the variance of the error is not changing throughout the forecast horizon. We examined many examples of forecasted days in the test period and we found the observations we have described here to be common to all of them.

Our reported MAPE over the entire test period is 13.14% (table 4.2). As we noted in section 2.1 state-of-art STLF for a single home achieves MAPE's in the 30%-45% range whereas state-of-art grid-level STLF (an aggregation of thousands of homes) achieves MAPE's in the 1%-2% range. Here we have an aggregation of 31 appliances (somewhat in between 1 and thousands) and so our MAPE accuracies perhaps fall within the range that we would expect between these two extremes. We will discuss this further in section 4.4 after reporting our dishwasher and washing machine results.

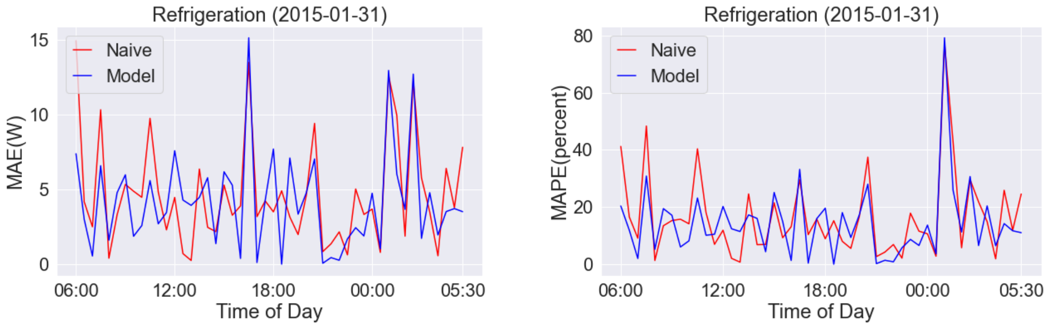


Figure 4.3: MAE (left) and MAPE (right) over the forecast interval for the first day in the test period. The mean MAE between the forecasts and the actual signal is 4.18W vs 4.70W for the model and naive forecasts respectively. The MAPE between the forecasts and the actual signal is 14.34% and 16.68% for the model and naive forecasts respectively.

4.1.2 Exploring the Learnt Features

We can explore further some of the important features we would expect our model to have learnt by averaging the forecasted and actual signals over different timescale dimensions as we did in our EDA work. For example, we can average the 48 forecasts for each day to obtain the mean daily forecasted power compared to the mean daily actual power and then view this over all the days in the test period to see how well the forecast model tracks the temperature change in the weather over a few months. We can also average each forecast interval over all the days in the test period and then look across the average forecast horizon to see how well the model forecasts the power changes due to daily temperature change and the behavioural interactions with the fridge.

These two views are shown in figure 4.4. The left plot shows the mean daily power of the actual signal, the naive baseline and the model forecasts over the test set period.

We observe that the model tracks closely with the actual mean daily power vs the naive baseline, which doesn't so well. The mean daily power is changing over this timescale due to changes in the temperature due to the weather. The naive baseline is essentially a 6 week moving average which is unable to track changes which occur much faster than this. The weather in the period around 7th March 2015 was particularly warmer than the surrounding period. This particular event can also be seen in the original EDA work in figure 2.4 around the same dates (the rightmost vertical arrow in the refrigeration and temperature plots). This divergence between the ability of the forecast model and the naive baseline to follow the change in power due to the weather is the primary reason the forecast model reports significantly better metrics than the naive baseline in table 4.2.

The apparent slight delay between the actual and the forecast power is due to the model forecasting the next 24 hours based on lagged power observations only. Since it doesn't have any other information, the model will forecast a daily mean power according to these historical values. Since the temperature of each day is variable from day to day, there is an error between what the model forecasts (based on the last few days) and the actual power based on what the actual temperature (of the future forecasted day) was. This error could be mitigated by providing the model with a weather forecast value of "tomorrows" expected temperature as an additional input feature to the model.

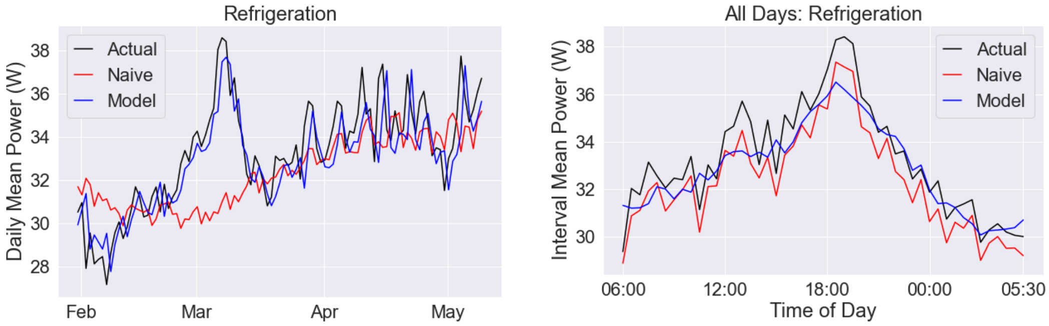


Figure 4.4: Left: Model forecast, Naive forecast and Actual daily mean power vs test day. The RMSE of the Model and Naive forecasts to the Actual daily mean are 1.33W vs 2.38W respectively. Right: Model forecast, Naive forecast and Actual interval mean power over the forecast horizon. The RMSE of the Model and Naive interval mean forecasts to the Actual interval mean are 1.0W vs 1.03W respectively.

Figure 4.4 (right) shows the average of the forecast intervals over the forecast horizon for all the days in the test period. We can see qualitatively from this figure that overall the model forecasts the mean daily load profile quite well. There is also stronger evidence that it forecasts the behavioural peaks around mealtimes (8am, 12:30pm and 7:30pm) but also that it does not capture it fully as in each of these regions we under forecast the actual signal. At the interval of peak load we note that we under-forecast the actual power by approximately 5% (36.5W vs 38.5W).

4.2 Dishwasher Forecasting

We observed in our EDA work (figure 2.3) that aggregated Dishwashers have a strong behavioural daily seasonality (the time of day that they are typically used is after meals and overnight) and, unlike refrigeration, a strong weekly seasonality in that we see weekend use is markedly different from weekdays.

4.2.1 Hyper-parameter Tuning and Test Results

We follow the same methodology for training and evaluating a model as we did for refrigeration. We use the same model architecture, settings and possible choices for the random hyper-parameter search (tables 3.4 and 3.5). Table 4.3 shows the top performing hyper-parameter configuration from the random hyper-parameter search.

RANK	NUM LAGS	LSTM HIDDEN	DENSE HIDDEN	VALIDATION RMSE	BEST EPOCH	FIT TIME (SECS)
1	192	512	512	57.924	76	1270

Table 4.3: Dishwasher - Top performing hyper-parameter settings.

The result of the test set evaluation on ten models trained with different seeds with the best performing hyper-parameter settings in table 4.3 is shown in table 4.4. We see that the model is worse than the naive baseline for both parameters by 5-8%. MAPE is not shown as it could not be computed on a daily basis as the actual aggregated signal goes to zero at some points during the course of each day due to the very low number of appliances in the aggregation (there's no base load). The standard deviation over the 10 models is small indicating that the model training process with different seeds but with this particular set of hyper-parameters is quite repeatable.

EVALUATION METRIC:	RMSE	MAE
NAIVE BASELINE	47.38	36.15
MODEL (NO DOW FEAT)	50.05	39.18
STDDEV(%)	0.86	2.72
VS BASELINE	+5.6%	+8.4%
MODEL (W/ DOW FEAT)	48.88	37.75
STDDEV(%)	0.45	0.8
VS BASELINE	+3.2%	+4.4%

Table 4.4: Dishwasher Test Set Evaluation without and with the Day of Week input feature.

Figure 4.5 shows an example forecasted day (the first day of the test period). As with the refrigeration example, we see that the actual signal is very erratic compared to the naive baseline and model forecasts. Comparing our RMSE and MAE results (50.05W and 39.18W respectively) to the refrigeration test results (5.23W and 4.19W respectively) we can see that the metrics are approximately an order of magnitude worse. On the other hand, the signal is only of the order of 2-3 times higher at the peak (80W for the dishwasher aggregation and 30W for the refrigeration aggregation for this example day). We ascribe the relative degradation in accuracy metrics to the lower number of appliances in the aggregation coupled with the behavioural differences of dishwashers vs refrigeration loads (which are more consistent over the day).

4.2.2 Exploring the learnt features

As we did with refrigeration, we now explore what salient features of the aggregated dishwasher load the model has learnt to accommodate in the forecast. The two timescale averaged views are shown in fig 4.6. We can see in the left plot that unlike refrigeration, there is no trend over the test period in mean daily power. We can see that the forecast model does not follow the excursions in the actual signal very much at all, presenting more of an average signal through the entire test period. This is likely day to day random behaviour in our aggregation which we are observing because of the very limited number of appliances coupled with the pure behavioural nature of dishwashers.

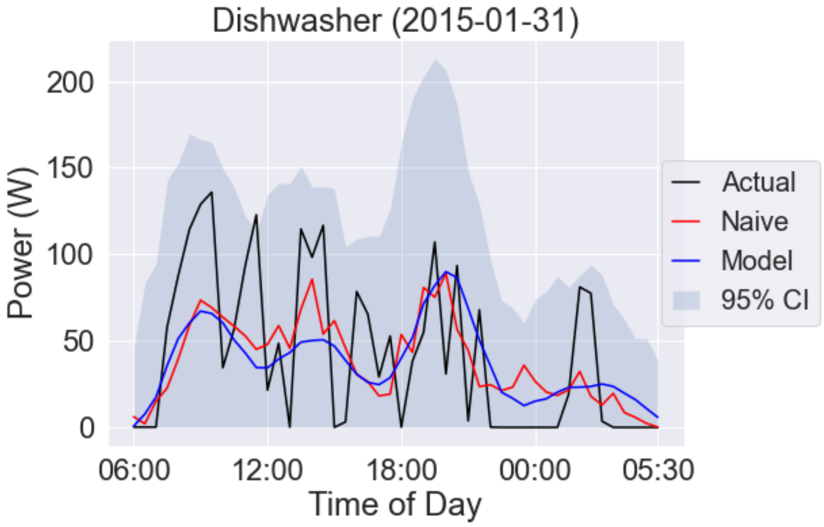


Figure 4.5: Dishwasher Model Forecast, Naive Baseline Forecast and Actual over the forecast interval for the first day in the test period. The RMSE between the forecasts and the actual signal is W vs W for the model and naive baseline respectively.

In the right plot we see that the forecasted average day tracks the actual signal very well for both the model and the naive baseline. We note that we under-forecast the interval of maximum demand by approximately 10% (80W vs 88W). The fact that the daily plot shows quite a lot of erratic behaviour and the interval plot is relatively smooth indicates that the times that dishwashers are likely to get used is typically the same (around mealtimes and overnight) but day to day the number of homes running them at all in the aggregation varies somewhat randomly (unlike in refrigeration where we mostly assume they are on all the time in all the homes).

Figure 4.7 shows the result of adding a one-hot-encoded day of week set of features to the model (as described in figure 3.1). We can see that with only lagged observations the model was not able to distinguish any temporal characteristics in the days leading up to Saturdays (in particular) or Sundays (mostly, although there is a little more separation as Saturday's are in fact distinctly different than workdays and so would provide some temporal difference in the lagged observations to generate a different forecast for Sundays). The right plot shows that with the addition of the day of week feature, the model is now able to forecast both weekend days vs weekdays quite differently and is qualitatively similar to the profiles we see in the EDA work (figure 2.3).

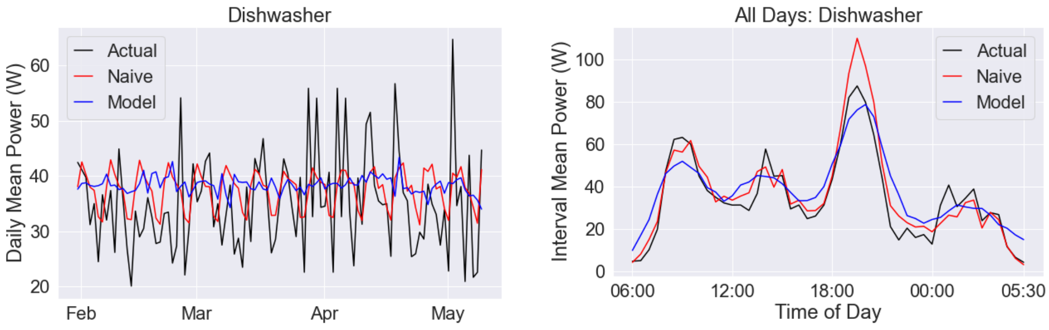


Figure 4.6: Left: Model forecast, Naive forecast and Actual daily mean power vs test day for aggregated dishwashers. The RMSE of the Model and Naive forecasts to the Actual daily mean are 9.71W vs 8.59W respectively. Right: Model forecast, Naive forecast and Actual interval mean power over the forecast horizon. The RMSE of the Model and Naive interval mean forecasts to the actual interval mean are 9.58W vs 7.15W respectively.

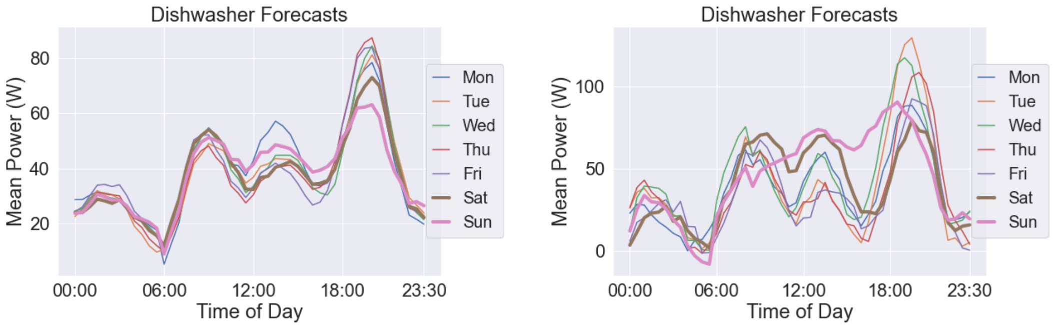


Figure 4.7: Left: Forecast average daily load profile without day-of-week feature. Right: Forecast average daily load profile with day-of-week feature. Adding the day-of-week feature to the model enables it to learn a different daily load profile for weekends vs weekdays and compares well with our expectations from the EDA work (figure 2.3).

4.3 Washing Machine Forecasting

Finally, we present our results for Washing Machines. Once again,we use the same training, hyper-parameter tuning and test period evaluation procedure. The best model found has the parameters shown in table 4.5 and the test evaluation results are presented in table 4.6. Here we are only showing results with the day of week feature also included (as, with the dishwasher, we observed it helped the model capture the weekend vs weekday differences).

A single forecasted day example is shown in figure 4.8. As was the case with the other two appliances we see a somewhat smooth model forecast, a very erratic actual

RANK	NUM LAGS	LSTM HIDDEN	DENSE HIDDEN	VALIDATION RMSE	BEST EPOCH	FIT TIME (SECS)
1	384	1024	256	30.356	10	2614

Table 4.5: Washing Machine - Top performing hyper-parameter settings.

EVALUATION METRIC:	RMSE	MAE
NAIVE BASELINE	28.99	20.15
MODEL (W/ DOW FEAT)	28.91	20.20
STDDEV(%)	0.36	0.77
VS BASELINE	-0.28%	+0.25%

Table 4.6: Washing machine Test Set Evaluation (with the Day of Week input feature).

signal with the naive baseline somewhere in between.

Our two averaged timescale views are shown in figure 4.9. Over the test period (left) we can see a strong periodicity in the actual mean daily power signal which the naive baseline follows very well and the forecast model less so. This is the strong weekly seasonality of washing machine use where the mean power of washing machines goes up significantly at weekends (particularly Saturdays). Adding the day of week feature helped the forecast model considerably here but it still has a residual error in this seasonality that needs further work to address. We make the following observations from this data compared to dishwashers (the other behaviourally driven appliance). Unlike dishwashers which tend to get used quite frequently throughout the week, washing machines tend to get used most at weekends. They both exhibit different times of use at weekends vs weekdays but the difference with washing machines is that they get used by a lot more households in the aggregation during the weekend vs the weekday, not just at different times (as with dishwashers). The right plot shows the model forecast smooths out the peak load of day, under forecasting it by approximately 10% (48W vs 53W) on average and also shifting the predicted time of peak load to about 1 hour later. From figure 2.3 we can see that the timing of the morning peak load for washing machines varies quite a bit day to day, even during the workweek. This is not the case for the other appliances so in this case we see a smoothing out of the predicted peak load time that we did not see with the other two appliances.

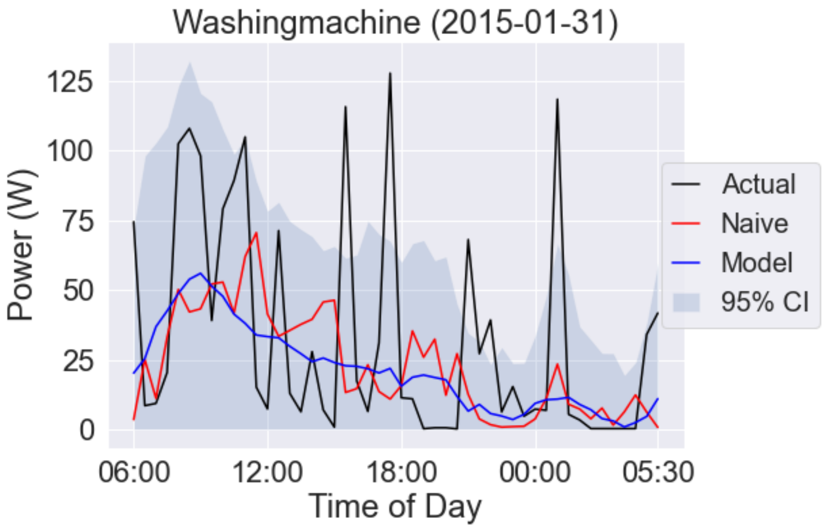


Figure 4.8: Washing Machine Model Forecast, Naive Baseline Forecast and Actual over the forecast interval for the first day in the test period. The RMSE between the forecasts and the actual signal is 6.31W vs 5.76W for the model and naive baseline respectively.

4.4 Discussion

We have shown that the forecasting models can learn the first level of important seasonal features that drive aggregated power consumption by appliance type for three different appliances. We summarise the ability to forecast, on average, the peak load of day in table 4.7. Here we see that on average the model under-forecasts the peak load of day consistently for all three appliances. Although this was not a design feature of the model, this is actually a benefit for DSR as over-forecasting (and then not being able to deliver the committed load shift) comes with significant financial penalties. To design for this aspect we would need to use an differentiable asymmetric loss function and a full understanding of the commercial costs of under vs over forecasting to set the parameters correctly [48].

We ran into two practical issues evaluating the models. First, erratic behaviour on the actual signal prevented us from evaluating the accuracy of the model as you would conventionally do on a forecasted day-to-day basis. Instead we had to resort to averaging both our forecasted and actual signal over different timescale views and then compare them in order to demonstrate model forecasting accuracy to specific seasonal features. Secondly, we were unable to compute MAPE for dishwashers and washing machines due to the actual signal dropping to zero or near zero during the forecast. We

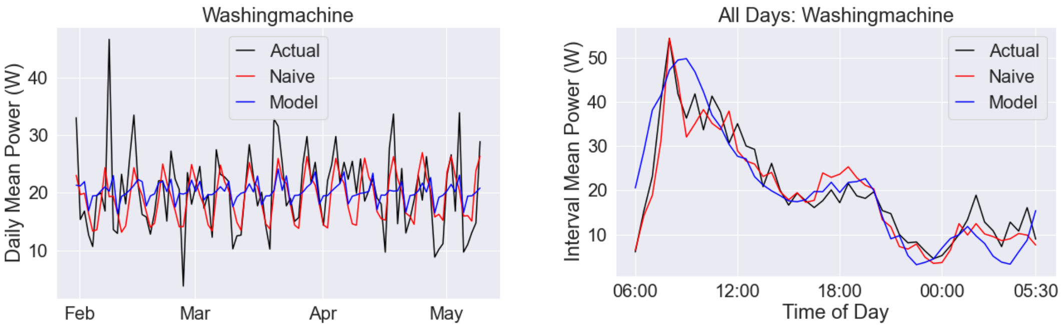


Figure 4.9: Model forecast, Naive forecast and Actual daily mean power vs test day for aggregated washing machines. The RMSE of the Model and Naive forecasts to the Actual daily mean are 6.31W vs 5.76W respectively. Right: Model forecast, Naive forecast and Actual interval mean power over the forecast horizon. The RMSE of the Model and Naive interval mean forecasts to the actual interval mean are 5.92W vs 3.70W respectively.

APPLIANCE	ACTUAL PEAK(W)	FORECAST PEAK(W)	ERROR
DISHWASHER	88	80	-9.1%
REFRIGERATION	38.5	36.5	-5.2%
WASHING MACHINE	53	48	-9.4%

Table 4.7: Average Peak Load of Day Forecast Summary.

argue that these two issues result entirely from the limited number of appliances we were able to assemble in each aggregation for this work, and would not be an issue in a real-world implementation.

The problem of accuracy varying in short-term load forecasting due to limited levels of aggregation is discussed in [1] and [18] where the authors introduce the concept of the Aggregation Error Curve (AEC). As aggregation levels change from low to high, the effects of individual random behaviours (either human or appliance driven) are, at least initially, reduced by the square root of the number of signals in the aggregation (the so-called law of large numbers). Consequently, the accuracy of models forecasting such aggregated signals follow the same law. The authors term this behaviour as being within the "scaling regime". At very high levels of aggregation, however, the AEC curve starts to diverge from the law of large numbers and eventually saturates to a fixed error (2% in their particular case study of 180,000 homes) - the "saturation regime". Whilst this study was focused on aggregated household-level STLF we argue that we are observing the same phenomena in our study. Indeed, at only 14-31 appliances in each aggregation, we are well inside the "scaling regime" according to their data.

When we averaged across the two different timescales to see if the models had learnt the seasonal features of interest we were, essentially, increasing our aggregation by performing an *aggregation in time*. By averaging the signal across the 48 intervals of the forecast horizon we were producing a point forecast of the daily mean load which had 6.9 times (square root of 48) less aggregation error than any individual interval forecast. Similarly, when we averaged the same forecast interval over all 100 days in the test period we are reducing the aggregation error by a factor of 10 times. With only 14 dishwashers, 19 washing machines and 31 refrigeration appliances in each of our aggregations, the aggregation error provides a floor to the forecasting accuracy we can report, specific to the number of appliances in the aggregation. By performing aggregation in time across the two timescale dimensions we were able to reduce the aggregation error as if we had 48 or 100 times as many appliances. Aggregation in time, of course, can only be applied when looking for specific, pre-conceived seasonal features over which to aggregate a different seasonality. Table 4.8 shows a summary of the reduction in RMSE when averaging across the two timescales for the 3 appliances. We can see that the RMSE has reduced by approximately 5 times in each case. That it didn't reduce by 6.9 or 10 times indicates that the RMSE error is now likely due to the model, not the aggregation error, which has been reduced to lower levels.

APPLIANCE	DAY-TO-DAY RMSE	DAY MEAN RMSE	INTERVAL MEAN RMSE
DISHWASHER	48.88	9.58	9.71
REFRIGERATION	5.23	1.00	1.33
WASHING MACHINE	28.91	5.92	6.31

Table 4.8: Forecast RMSE summary and estimates of model forecast error by averaging across the two seasonal timescales.

In a real-world implementation we would need to assemble enough appliances in our aggregations such that we were either in the saturation regime or far enough down the AEC such that the aggregation error is lower than errors arising from the model's limitations itself. This latter point is very important. We argue that *the models we implemented have lower forecasting error than the aggregation error allows us to measure.* We were able to demonstrate this by aggregating in time both the model forecast and actual signals and then comparing the aggregation in time results. To further illustrate this and to provide, for the first time, an Aggregation Error Curve for an appliance-level aggregation, we implemented a refrigeration appliance simulator from which we can construct an arbitrary aggregation of appliances.

The details of the simulator are given in Appendix H . For each simulated appliance the cycling nature of the refrigeration compressor is modelled (running power, frequency, phase and duty cycle) and the duty cycle is varied sinusoidally over a 24-hour period to simulate the daily seasonality due to the change in ambient temperature. We also include a simulation of the random interaction with the appliance by users around mealtimes (by increasing the duty cycle a random amount for a half-hour period around these times). The simulator parameters were derived and calibrated from analysing the raw refrigeration data files and performing both time-series and spectral analysis as described in the appendix.

We made aggregations of 1, 10, 31, 100, 500, 1k and 5k refrigeration appliances generated using our appliance simulator. We then ran these waveforms through the *same model* we trained in section 4.1 and evaluated the performance of the model over the same test period and using the same metrics and methodology as previously described. The results are shown in table 4.9. We can clearly see the effect of the aggregation error reducing beyond the 31 appliance aggregation we were limited to in our study. The trained model has considerably less error than the aggregation error at 31 appliances - at 5000 appliances, for example, it is able to forecast the signal to a MAPE accuracy of 2.87%. We show this result in the form of Aggregation Error Curve (AEC) in the same form as in [1] for direct comparison. We can see that in the case of our refrigeration aggregation we are firmly in the "scaling regime" up until approximately a 100 appliance aggregation after which we start to asymptote towards the "saturation regime".

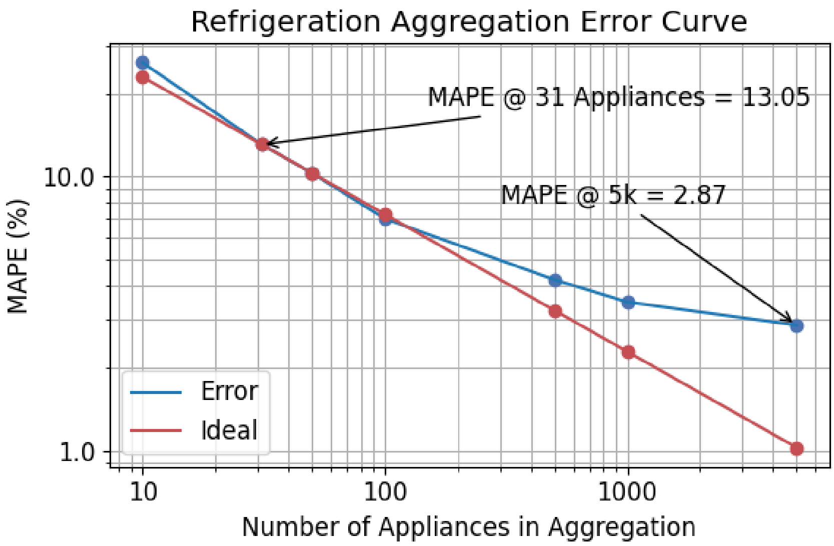


Figure 4.10: Refrigeration Aggregation Error Curve (AEC).

Evaluation Metric:	RMSE	MAE	MAPE(%)
Naive Baseline	5.820	4.695	14.488
Forecast Model 10x mean	5.230	4.192	13.140
Simulated 31 Appliances	5.60	4.61	13.05
Simulated 100 Appliances	3.13	2.56	7.00
Simulated 1000 Appliances	1.66	1.31	3.48
Simulated 5000 Appliances	1.41	1.09	2.87

Table 4.9: Test set results with higher levels of aggregation utilising the refrigeration appliance simulator. Actual forecasting results on REFIT 31 appliance aggregation are reprinted from table 4.2 for ease of comparison.

Chapter 5

Conclusions and Future Work

In this work we proposed an alternative approach to domestic demand side response (DSR) than those previously presented - that we monitor and control power load by appliance type, aggregated over many homes. This work presented exploratory data analysis (EDA) of aggregated appliance loads for three different DSR-eligible appliances - refrigeration, dishwasher and washing machine from the REFIT dataset. We found that for this particular community of homes, each of these appliances had unique patterns of power consumption over the course of a day, week and year and that there were both behavioural and environmentally (temperature) driven factors. We concluded that these differences could be used to advantage in the demand side response approach we have proposed.

We demonstrated that LSTM-based deep-learning forecasting models can be successfully trained to forecast the aggregated appliance loads to facilitate our DSR approach. We obtained initial accuracies on the aggregated forecasts higher than those previously reported on household level short term load forecasting but not as good as grid-level (highly aggregated) forecasting work. We were able to show that the error in our forecasting results was not limited by model forecasting error but rather by aggregation error [1] as a result of having too few appliances in our aggregations. We demonstrated this limitation in two ways:

Firstly we showed that we could take advantage of the seasonalities in our data and *aggregate in time* to produce more accurate estimates of the model forecasting ability. We provided this analysis over two seasonalities and we found our model had approximately 5 times less RMSE across these timescales than the aggregation error was allowing us to measure in the day-to-day forecasts. Using this approach, we were able to demonstrate that the model can forecast the average peak daily load (a point of particular interest for DSR) to an accuracy of approximately 5%-10% depending on appliance. We were also able to show that the model consistently under-forecasts the peak load of day, which is preferable in a DSR application.

Secondly, we implemented a refrigeration appliance simulator to allow us to construct aggregations of an arbitrary number of appliances. By testing such aggregations with the refrigeration model trained on the REFIT dataset, we were able to show that the MAPE forecasting accuracy of the model was approximately 3% rather than the 13% that the initial results had indicated. We were further able to show an appliance-level aggregation error curve [1] for refrigeration. The curve showed divergence of the model from the ideal curve at around 100 appliances which indicates the minimum number of refrigeration appliances needed in an aggregation to start to see sensitivity to model performance itself.

The next level of features that would be interesting to explore would be the effect that public holidays have both on themselves and the days around them. Similarly, periods of festivals (Christmas, Easter etc) are of interest in how they impact the behavioural use of appliances. These were all sparse in our dataset as it only covered a single year and multiple years of such data would likely be required to get enough examples in the dataset to see meaningful effects there.

The biggest issue though hindering the investigation of learnt features (pre-conceived or not - the very reason for utilising a deep-learning architecture) is the aggregation error which must be reduced. For that we need to assemble larger cohorts of appliances. We could look at combining data from different datasets but as we pointed out, these need to come from culturally similar homes. Clustering analysis could be useful here to identify similar communities and then combine.

Other appliances could be studied, perhaps from datasets where there are a plethora of data from different homes even if the appliance itself is not DSR-eligible. Filtering the data to reduce the signal noise would be another area of interest but care must be taken here as the results arbitrarily improve as more and more filtering is applied to the point where all the temporal information is removed from the signal and we are simply predicting the mean of the entire dataset - very accurately.

Probably the most fruitful area of future research would be to combine NILM with the project, obtaining the household-level data from a large number of homes and then disaggregating it into appliance-level signals to re-aggregate into appliance aggregations. There are a number of very large household level datasets (e.g. as in [1]) which would certainly provide enough data to reduce the aggregation error to the "saturated regime".

Bibliography

[1] Ra Sevlian and Ram Rajagopal. Electrical Power and Energy Systems A scaling law for short term load forecasting on varying levels of aggregation. 98(October 2017):350–361, 2018.

[2] Hassan Farhangi. The path of the smart grid. *IEEE Power and Energy Magazine*, 8(1):18–28, jan 2010.

[3] Özge Okur, Petra Heijnen, and Zofia Lukszo. Aggregator's business models in residential and service sectors: A review of operational and financial aspects, apr 2021.

[4] Electrical system operator - balancing services. https://www.nationalgrideso.com/industry-information/balancing-services. Accessed: 2021-04-26.

[5] National grid eso - stor technical requirements. https://www.nationalgrideso.com/industry-information/balancing-services/Reserve-Services/Short-term-operating-reserve/Technical-Requirements. Accessed: 2021-04-26.

[6] Tesco: How a new partnership is saving energy using the internet of things. https://ktn-uk.org/casestudy/tesco/. Accessed: 2021-04-26.

[7] Power responsive - demand side flexibility. http://powerresponsive.com/. Accessed: 2021-04-19.

[8] The association for decentralised energy. https://www.theade.co.uk/resources/what-is-demand-side-response. Accessed: 2021-04-19.

[9] Digest of uk energy statistics (dukes): Electricity. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/

attachment_data/file/904805/DUKES_2020_Chapter_5.pdf. Accessed: 2021-04-11.

[10] Maximilian Wurm and Vlad C. Coroamă. Poster abstract: grid-level short-term load forecasting based on disaggregated smart meter data. *Computer Science - Research and Development*, 33(1-2):265–266, feb 2018.

[11] Chinthaka Dinesh, Stephen Makonin, and Ivan V. Bajic. Residential Power Forecasting Using Load Identification and Graph Spectral Clustering. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(11):1900–1904, 2019.

[12] Chinthaka Dinesh, Stephen Makonin, and Ivan V. Bajic. Residential Power Forecasting Based on Affinity Aggregation Spectral Clustering. *IEEE Access*, 8:99431–99444, 2020.

[13] Sergio Iván Elizondo-González and Sergio Iván. Market-based coordination for domestic demand response in low-carbon electricity grids. 2017.

[14] Yuting Ji, Elizabeth Buechler, and Ram Rajagopal. Data-Driven Load Modeling and Forecasting of Residential Appliances. *IEEE Transactions on Smart Grid*, 11(3):2652–2661, may 2020.

[15] Ye Hong, Yingjie Zhou, Qibin Li, Wenzheng Xu, and Xiujuan Zheng. A deep learning method for short-term residential load forecasting in smart grid. *IEEE Access*, 8:55785–55797, 2020.

[16] Aida Mehdipour Pirbazari, Mina Farmanbar, Antorweep Chakravorty, and Chunming Rong. Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques. *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 2020.

[17] Khursheed Aurangzeb, Musaed Alhussein, Kumail Javaid, and Syed Irtaza Haider. A Pyramid-CNN based deep learning model for power load forecasting of similar-profile energy customers based on clustering. *IEEE Access*, 9:14992–15003, 2021.

[18] Raffi Sevlian and Ram Rajagopal. Short Term Electricity Load Forecasting on Varying Levels of Aggregation. mar 2014.

[19] David Murray, Lina Stankovic, and Vladimir Stankovic. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. `https://doi.org/10.1038/sdata.2016.122`.

[20] W R Christiaanse. Short-term load forecasting using general exponential smoothing. *IEEE Transactions on Power Apparatus and Systems*, PAS-90(2):900–911, 1971.

[21] Hesham K. Alfares and Mohammad Nazeeruddin. Electric load forecasting: Literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, jan 2002.

[22] Philipp Theile, Anna Linnea Towle, Kaustubh Karnataki, Alessandro Crosara, Kaveh Paridari, Graham Turk, and Lars Nordstrom. Day-ahead electricity consumption prediction of a population of households: Analyzing different machine learning techniques based on real data from RTE in France. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018*, 2018.

[23] Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani. Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7), 2018.

[24] Chujie Tian, Jian Ma, Chunhong Zhang, and Panpan Zhan. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies*, 11(12), dec 2018.

[25] Xianlun Tang, Yuyan Dai, Qing Liu, Xiaoyuan Dang, and Jin Xu. Application of Bidirectional Recurrent Neural Network Combined with Deep Belief Network in Short-Term Load Forecasting. *IEEE Access*, 7:160660–160670, 2019.

[26] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access*, 8:180544–180557, 2020.

[27] Heng Shi, Minghao Xu, and Ran Li. Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN. *IEEE Transactions on Smart Grid*, 9(5):5271–5280, sep 2018.

[28] Abdul Wahab, Muhammad Anas Tahir, Naveed Iqbal, Faisal Shafait, Syed Muhammad, and Raza Kazmi. Short-term load forecasting using Bi-directional sequential models and feature engineering for small datasets, nov 2020.

[29] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using Deep Neural Networks. In *IECON Proceedings (Industrial Electronics Conference)*, pages 7046–7051. IEEE Computer Society, dec 2016.

[30] Kasun Amarasinghe, Daniel L. Marino, and Milos Manic. Deep neural networks for energy load forecasting. In *IEEE International Symposium on Industrial Electronics*, pages 1483–1488. Institute of Electrical and Electronics Engineers Inc., aug 2017.

[31] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid*, 10(1):841–851, jan 2019.

[32] Mohamed Chaouch. Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves. *IEEE Transactions on Smart Grid*, 5(1):411–419, jan 2014.

[33] Debneil Saha Roy. Household level electricity load forecasting using echo state network. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2020*. Institute of Electrical and Electronics Engineers Inc., nov 2020.

[34] Sarwan Ali, Haris Mansoor, Imdadullah Khan, Naveed Arshad, Muhammad Asad Khan, and Safiullah Faizullah. Hour-ahead load forecasting using AMI data, dec 2019.

[35] Yu Hsiang Hsiao. Household electricity demand forecast based on context information and user daily schedule analysis from meter data. *IEEE Transactions on Industrial Informatics*, 11(1):33–43, feb 2015.

[36] Cillian Brewitt and Nigel Goddard. Non-intrusive load monitoring with fully convolutional networks, dec 2018.

[37] Nigel Goddard, Jonathan Kilgour, Martin Pullinger, D.K Arvind, Heather Lovell, Johanna Moore, David Shipworth, Charles Sutton, Jan Webb, Niklas Berliner,

Cillian Brewitt, Myroslava Dzikovska, Edmund Farrow, Elaine Farrow, Janek Mann, Evan Morgan, Lynda Webb, and Mingjun Zhong. Ideal household energy dataset. `https://doi.org/10.7488/ds/2836`.

[38] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2(150007), 2015.

[39] Rob J Hyndman and George Athanasopoulis. *Forecasting Principles and Practice*. OTEXTS, 2018.

[40] Historic weather data for loughborough, united kingdom. `https://www.https://www.timeanddate.com/weather/uk/loughborough/historic`. Accessed: 2021-07-19.

[41] B V Vishwas and Patel Ashish. *Hands-on Time Series Analysis with Python*. Apress, 2020.

[42] Francesca Lazzeri. *Machine Learning for Time Series Forecasting with Python*. John Wiley & Sons, 2021.

[43] Jason Brownlee. Deep learning for time series forecasting. `https://machinelearningmastery.com`, 2021.

[44] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[45] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[46] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[48] Korbinian Dress, Stefan Lessmann, and Hans-Jörg von Mettenheim. Residual value forecasting using asymmetric cost functions. *International Journal of Forecasting*, 34(4):551–565, 2018.

Appendix A

Dataset Heatmaps

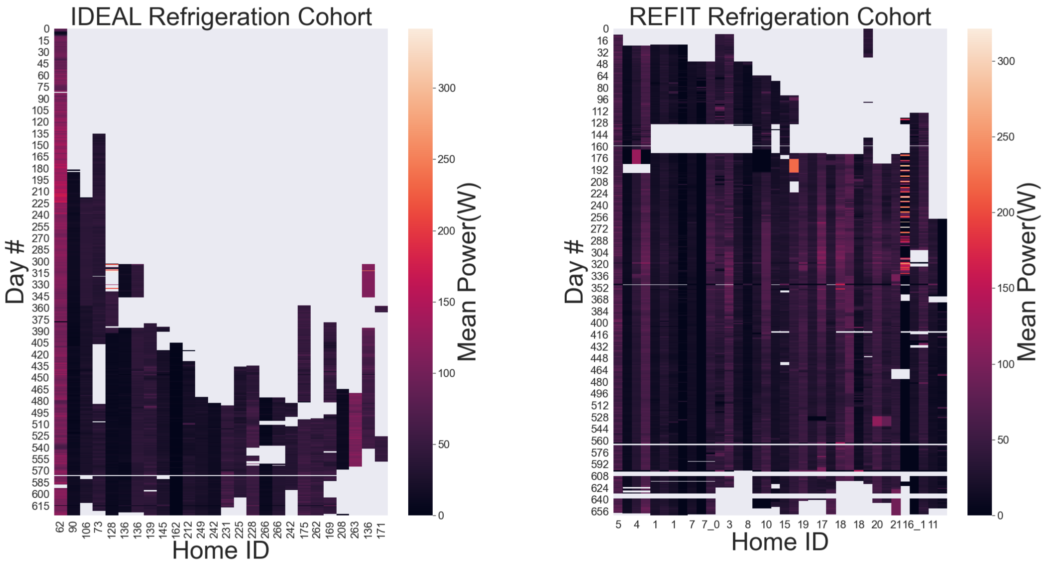


Figure A.1: Dataset Heatmap. Refrigeration. Left: IDEAL. Right: REFIT

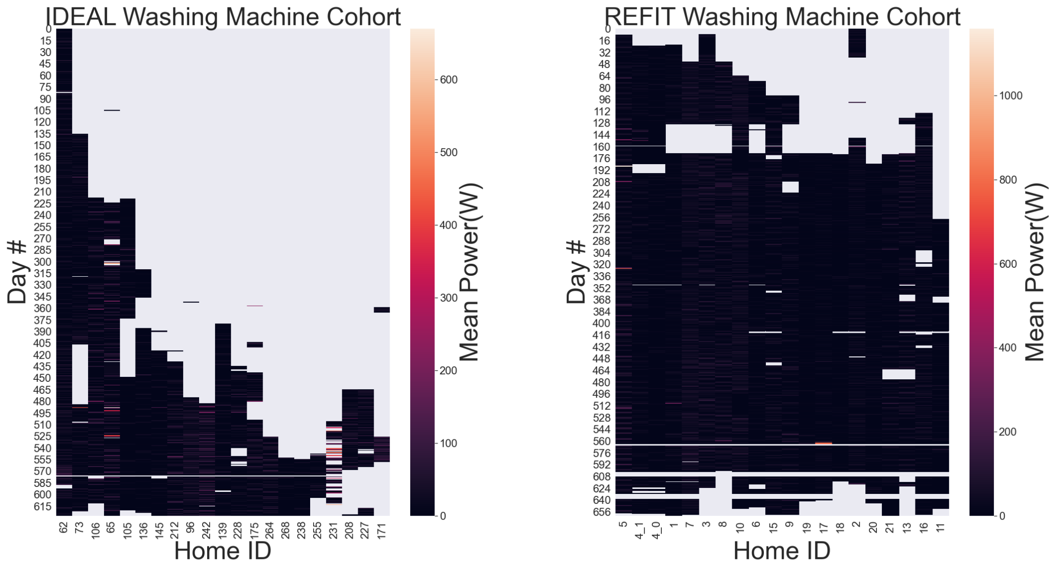


Figure A.2: Dataset Heatmap. Washing Machine. Left: IDEAL. Right: REFIT

Appendix B

Appliance Anomalies - TIMEPLOTS

Timeplots provide us a view of the data where we can see unusual temporal events through the dataset to be investigated further.

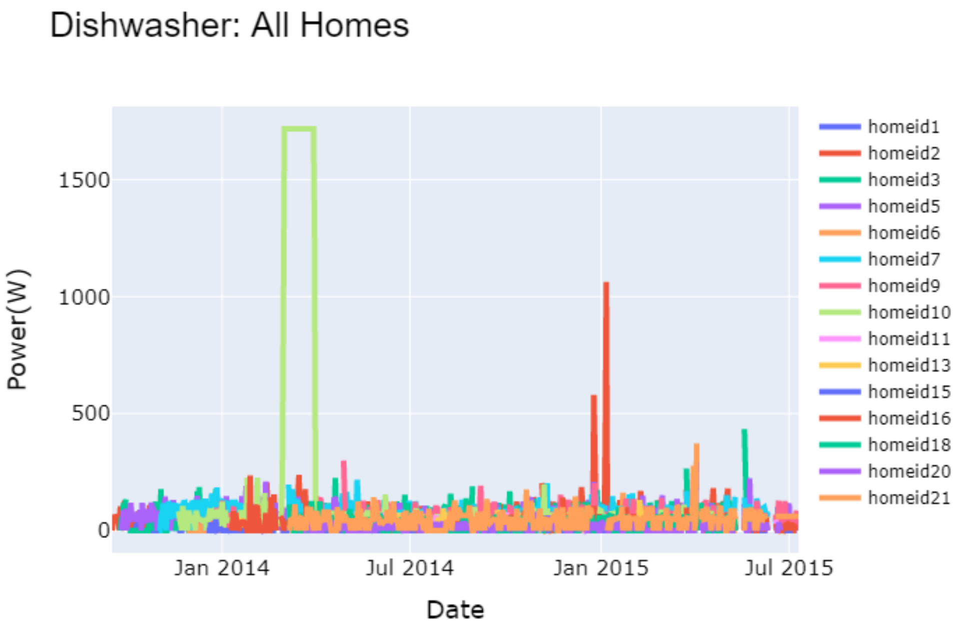


Figure B.1: Dishwasher long-term load profiles at 1 day resolution. The large "spike" from home 10 was in a period before our assembled cohort so it was not investigated. The large spikes from home 16 were investigated and found to be normal operation - in this home the dishwasher ran a lot over the Christmas period and January 5th/6th. No appliances were removed from the dishwasher dataset. The gaps towards the end were periods of system-wide failure.

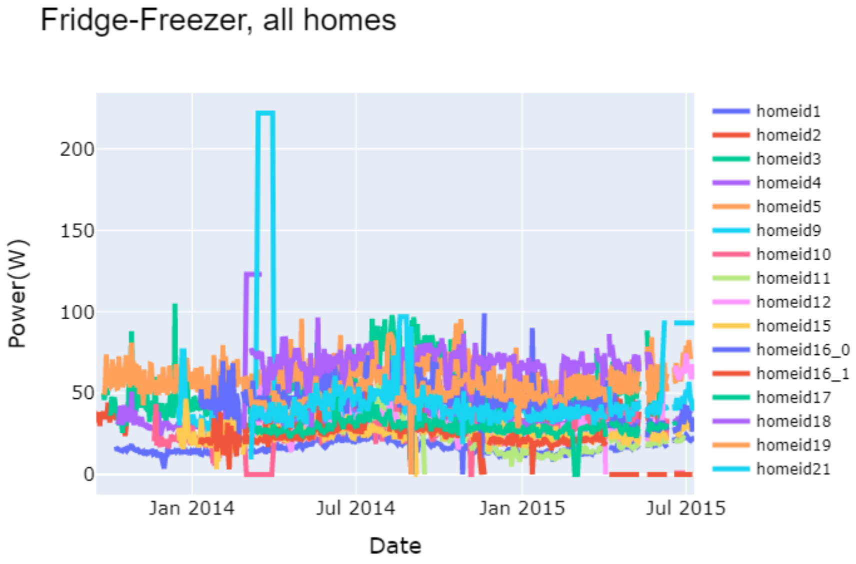


Figure B.2: Fridge-Freezer long-term load profiles at 1 day resolution. The spike in home 4 was before the cohort start date and was not investigated. Actually, so was the spike from home 9 (just) but we didn't know it at the time so it was investigated. See figure B.3.

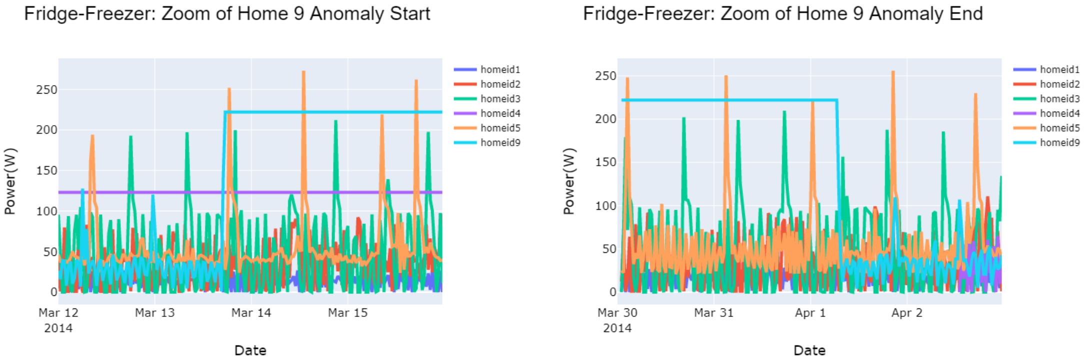


Figure B.3: Fridge-freezer home 9 anomaly. Zoom of the large spike from home 9 in the fridge-freezer plot (middle) in figure B.2. This event lasted 2 weeks (March 15th-31st 2014). Prior to and after the anomaly the appliance is cycling normally and similarly to other fridge-freezers in the plot. During the event, the load is a continuous 220W load with no cycling at all. A possible explanation for this anomaly is that the fridge door was accidentally left open, perhaps over a 2 week holiday when no-one was home, or, that for this period another appliance was plugged in instead. This seems like a normal event that could happen - the appliance was left in the dataset.

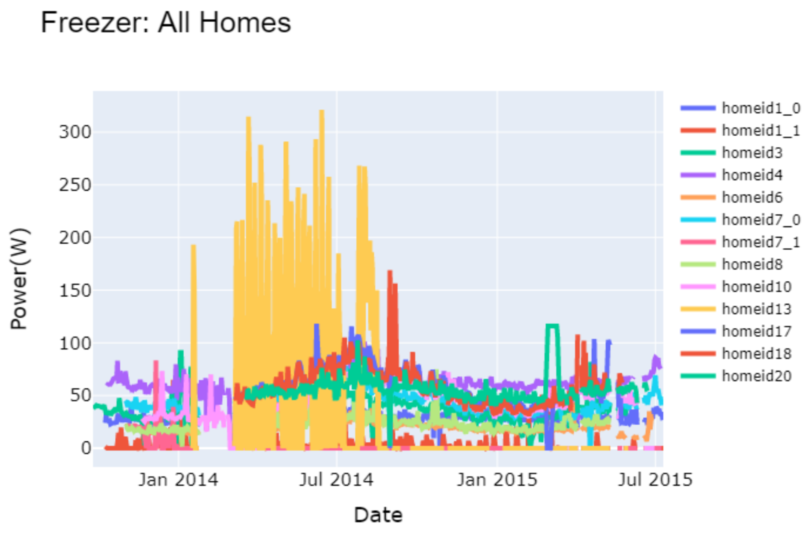


Figure B.4: Freezer long-term load profiles at 1 day resolution. Freezers in homes 13, 1 (freezer 1) and 20 identified for further investigation. For freezer in home 13 see figure B.5.

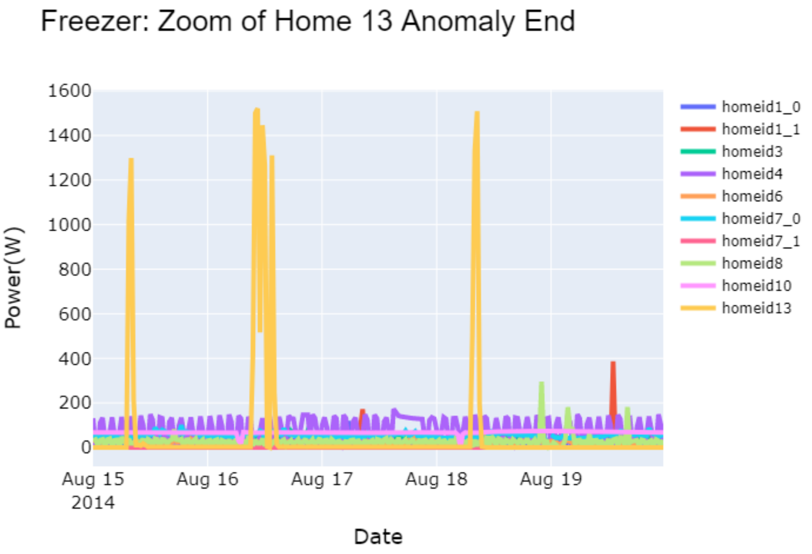


Figure B.5: Freezer home 13 anomaly. 30 Min resolution, zoomed into a few of the last peaks. The peak power values that this appliance is drawing (1.4kW on average, per 30 min) are very much higher than a typical freezer compressor load. Furthermore, the appliance is not cycling as we expect a freezer appliance to cycle over the course of a day (with periodicity in the range of minutes - hours) as can be seen with the other freezer appliances in this plot. We conclude that this appliance is unlikely to be a freezer.

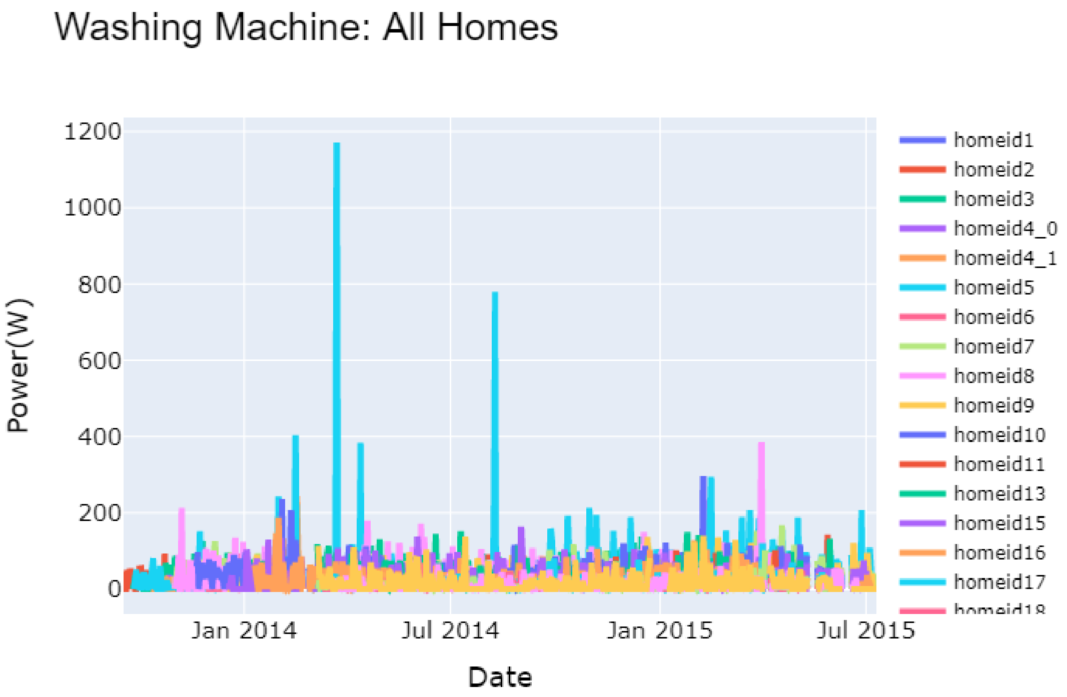


Figure B.6: Washing Machine long-term load profiles at 1 day resolution. Home 5 was identified for further investigation due to the large spikes but found to be operating normally - just more than usual on those particular days. See figure B.7.

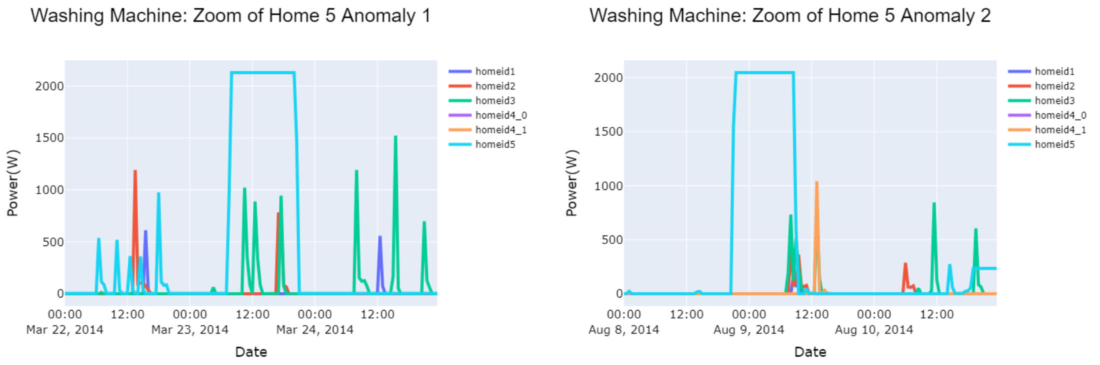


Figure B.7: Washing machine home 5 anomaly.A zoom of two of the spikes from home 5 shown in the washing machine plot (bottom) in figure B.6. In both cases (months apart) there is a continuous load of approximately 2kW for a few hours (and unusual hours for the example from August 2014). We speculate that the amount of power being drawn would indicate an appliance fault - that perhaps the water heater for the washing machine is stuck on for a few hours and then clears.

Appendix C

Appliance Anomalies - BOXPLOTS

Boxplots provide a view of the data allowing us to see appliances with unusual looking distributions to be investigated.

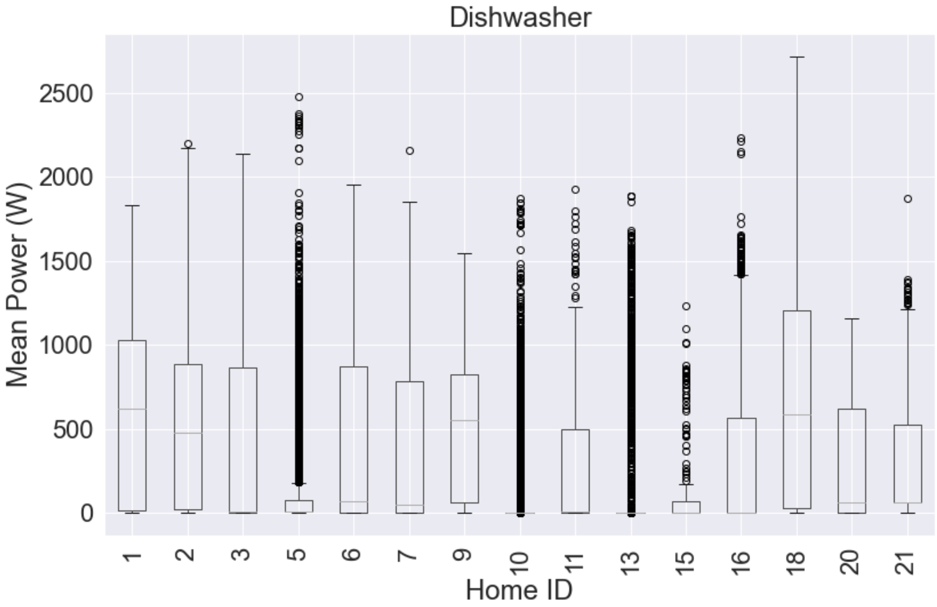


Figure C.1: Dishwasher individual appliance power distributions. Dishwashers in homes 5, 10, 13 have unusual looking distributions and were investigated but found to be operating normally.

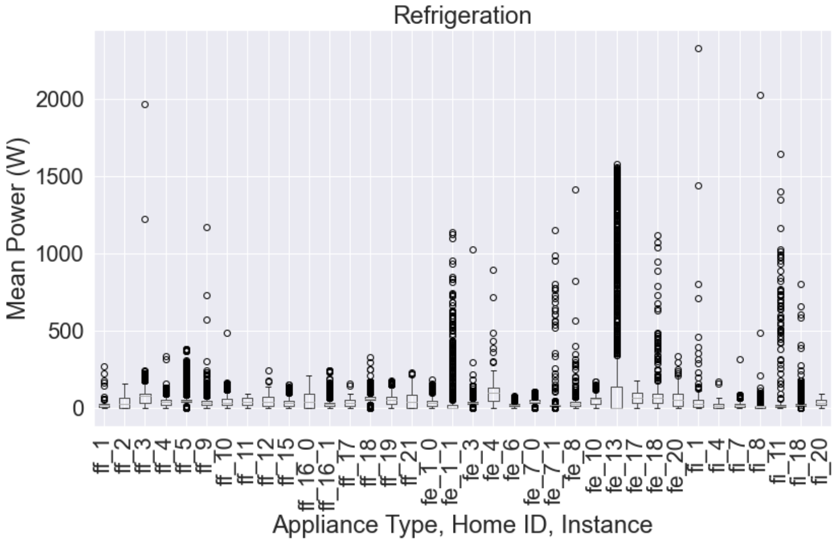


Figure C.2: Refrigeration individual appliance power distributions. ff=Fridge Freezer, fe=Freezer, fi=Fridge. Freezers from home 1 (fe-1-1), 7 (fe-7-1) and 13 (fe-13) were investigated and found to be anomalous. fe-1-1 and fe-7-1 were almost all zero values with just a few readings scattered throughout (the outlier values we see in the plot). fe-13 was determined not to be a freezer appliance. All three were removed from the dataset before aggregation. The fridge in home 11 (fi-11) was also investigated but found to be operating normally.

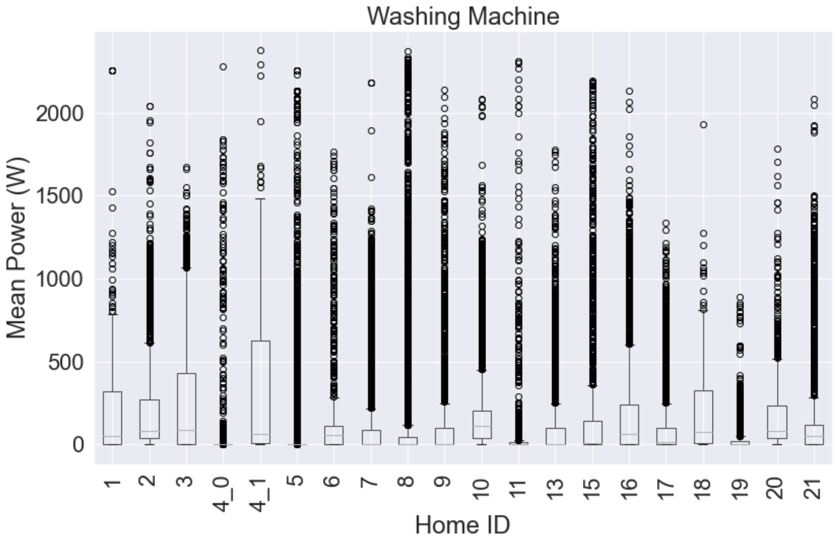


Figure C.3: Washing Machine individual appliance power distributions. Appliances from homes 4, 5, 11 and 19 were investigated but found to be operating normally.

Appendix D

Cohort Assembly Algorithm

Algorithm 1: Appliance Cohort Assembly

```
Input: Appliances Dataframe: $DF_A$, size $N_{Days}$ x $N_{Appliances}$.
Input: Required Minimum Number of Days: $L_{DaysMin}$
Compute $N_{days}$ not $NaN$ for each $A_n$ in $DF_A$
Initialize MergedCohort $DF_M = A_{MAX_D}$ with MAX($N_{days}$)
$R_N = N_{Appliances} - 1$
Drop $A_{MAX_D}$ from $DF_A$
while Length($DF_M$) > $L_{DaysMin}$ do
   for $i = 1$ to $R_N$ do
      $L_M$ = Test Inner Merge Length of $A_i$ with $DF_M$
      if $L_M > L_{best}$ then
         $L_{best} = L_M$, $A_{best} = A_i$
      end if
   end for
   if $L_{best}$ > $L_{DaysMin}$ then
      Execute Inner Merge $A_{best}$ with $DF_M$
      Trim orphaned dates from $DF_M$
      Drop $A_{best}$ from $DF_A$
      $R_N = R_N$ - 1
   end if
end while
```

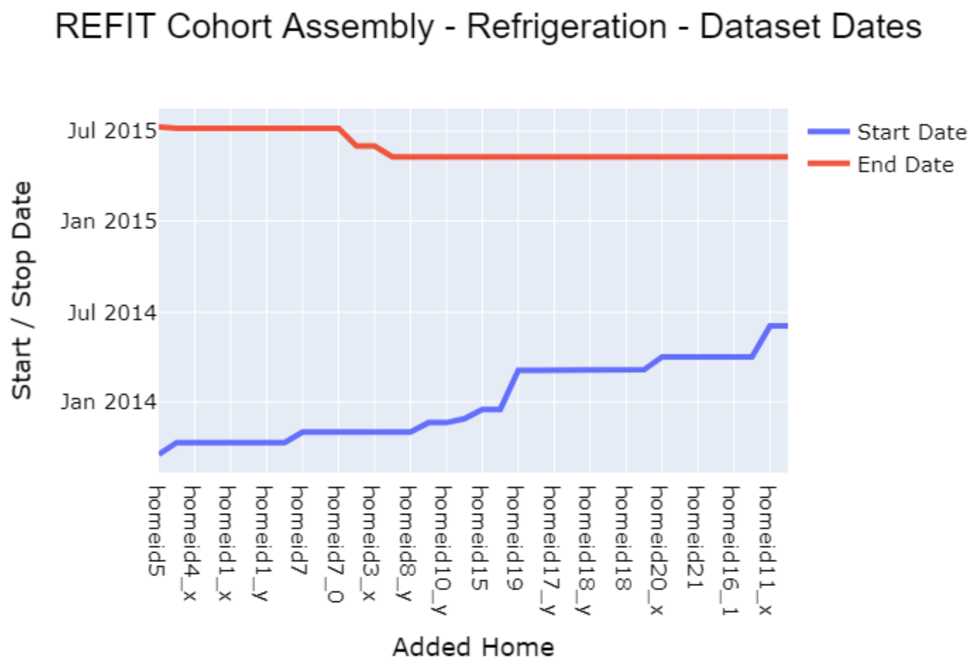


Figure D.1: Example of the cohort assembly process - refrigeration start and end dates vs appliances added.

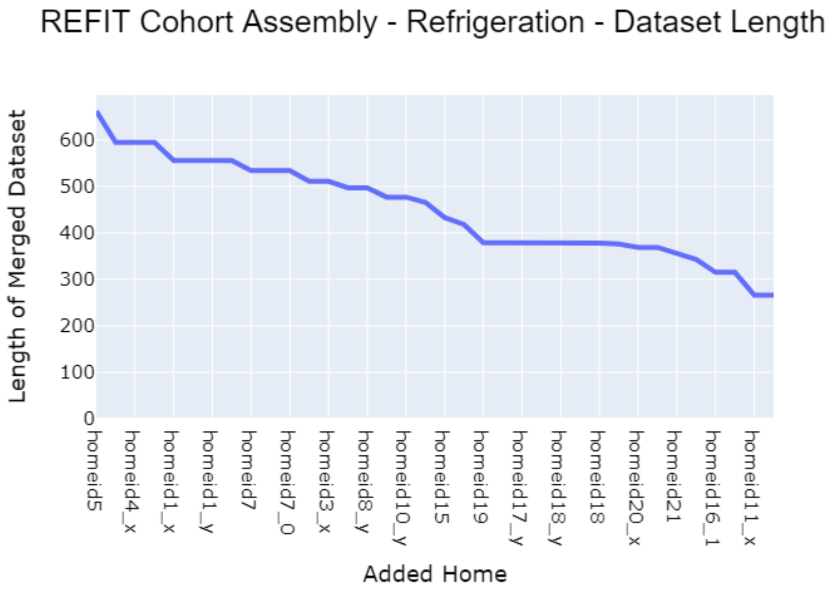


Figure D.2: Example of the cohort assembly process - refrigeration total length of cohort vs appliances added. Using a greedy choice, the algorithm attempts to maximise the length of overlapping days and the number of appliances in the cohort. In order to meet the 365 day requirement, the last 3 appliances were not included in the cohort.

Appendix E

Home ID’s for each Aggregation

Appliance	Home ID’s
Fridge-Freezer	1,2,3,4,5,9,10,12,15,16-0,16-1,17,18,19,21
Fridge	1,4,7,8,18,20
Freezer	1-0,3,4,6,7,8,10,17,18,20
Dishwasher	1,2,3,5,6,7,9,10,13,15,16,18,20,21
Washing Machine	1,2,3,5,4-0,4-1,6,7,8,9,10,13,15,16,17,18,19,20,21

Table E.1: Home ID’s of REFIT homes used in each appliance aggregation. Where there are two appliances of the same type in the home the appliance is futher marked with a dash and enumerated (e.g. Fridge-Freezer 16-0 and 16-1).

Appendix F

Detected Regions of System-wide Failure

The algorithm for finding regions of system-wide failure was simply to look for runs of where all appliances in the cohort had duplicated values for more than 12 hours. Usually these are missing data with no values at all (which could have been detected by just looking for regions where all the values were missing). However, there were instances where most of the appliances had no values and a few were "stuck" at the last observed value before the region commenced. Hence, we used the duplicate method which found both kinds of region. 12 hours was chosen as below that we would start to detect regions where all appliances were occasionally genuinely at zero (e.g. overnight for for washing machines was quite common). The regions detected and cleaned (by imputing the previous weeks values from the same day and interval) are shown in table F.1.

REGION	START	END	LENGTH
1	2014-08-01 18:00	2014-08-02 12:00	0 DAYS 18:00:00
2	2014-08-30 13:00	2014-09-01 08:00	1 DAYS 19:00:00
3	2014-10-27 21:30	2014-10-28 13:00	0 DAYS 15:30:00
4	2014-12-18 10:30	2014-12-19 10:00	0 DAYS 23:30:00
5	2015-03-22 08:30	2015-03-24 14:30	2 DAYS 06:00:00
6	2015-03-31 04:30	2015-04-01 15:30	1 DAYS 11:00:00
7	2015-04-03 02:00	2015-04-07 09:30	4 DAYS 07:30:00

Table F.1: Regions of system-wide failure, detected and cleaned

Appendix G

Naive Forecast Development

These are the seasonal-aware naive forecasts we could considered for each forecast interval of the forecast horizon:

- Persistence: Persist the value from the last observation. (e.g. for the forecast starting at 16:30 on a particular Saturday, persist the last observed value received at 16:00 for all 48 forecast intervals).
- Same Interval Yesterday: Impute the value from the same interval the previous day (e.g. for forecast interval Saturday 16:30, take the known observation from Friday at 16:30).
- Same Interval Last Week: Impute the value from the same interval the same day the previous week (e.g. take the known observation from the previous Saturday at 16:30).
- Day of Week Mean: Impute the mean of all the intervals for the same day in the dataset (e.g. average the observations from all Saturdays).
- Interval of Day Mean: Impute the mean of the same interval for all the days in the dataset (e.g. average the observations at 16:30 for all days).
- Day of Week / Interval of Day Mean: Impute the mean of the same interval of the same day in the dataset. (e.g. average all the observations from Saturdays at 16:30 in the dataset).
- Recent Day of Week / Interval of Day Mean: Impute the mean of the last few weeks of the same interval of the same day in the dataset. (e.g. average the last few observations from Saturday at 16:30 in the dataset).

NAIVE FORECAST	DAILY	WEEKLY	ANNUAL
PERSISTENCE (LAST OBSERVED VALUE)	NO	NO	YES
SAME INTERVAL YESTERDAY	YES	NO	YES
SAME INTERVAL LAST WEEK	YES	YES	YES
DAY OF WEEK MEAN	NO	YES	NO
INTERVAL OF DAY MEAN	YES	NO	NO
DOW / IOD MEAN	YES	YES	NO
RECENT DOW / IOD MEAN	YES	YES	YES

Table G.1: Ability of the naive forecasts to represent daily, weekly or annual seasonality in their computation.

The ability of these naive forecasts to capture the daily, weekly and annual seasonalities is summarised in table G.1.

The naive forecasts were computed over the test set and are shown in table G.2.

NAIVE	DISHWASHERS	REFRIGERATION	WASHING MACH.
FORECAST	RMSE	RMSE	RMSE
PERSISTENCE	62.0	7.23	37.4
YESTERDAY	65.9	6.94	40.6
LAST WEEK	58.9	6.79	34.9
DOW MEAN	50.9	6.72	30.6
IOD MEAN	48.2	6.38	29.1
DOW / IOD MEAN	47.3	6.42	**28.5**
RECENT DOW / IOD MEAN	**46.9**	**5.60**	28.7

Table G.2: Summary of RMSE on the test set for all the naive baselines. We observe that Recent DoW & IoD Mean is best in class for dishwashers and refrigeration and marginally 2nd best for washing machines. We therefore adopted this as our default naive forecast baseline.

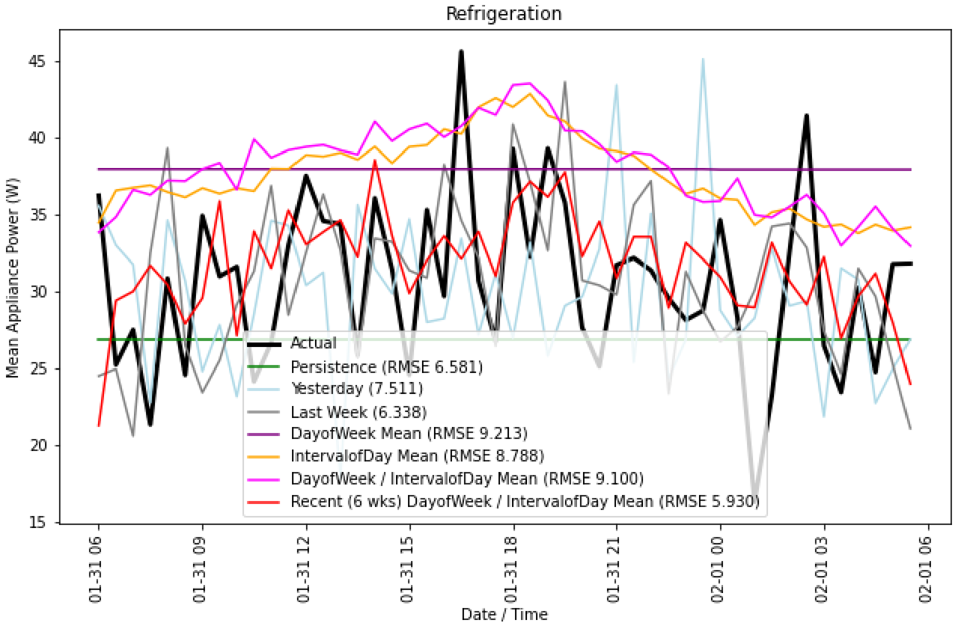


Figure G.1: Example naive forecasts for refrigeration over a 24 hour period (the first day of the test period).

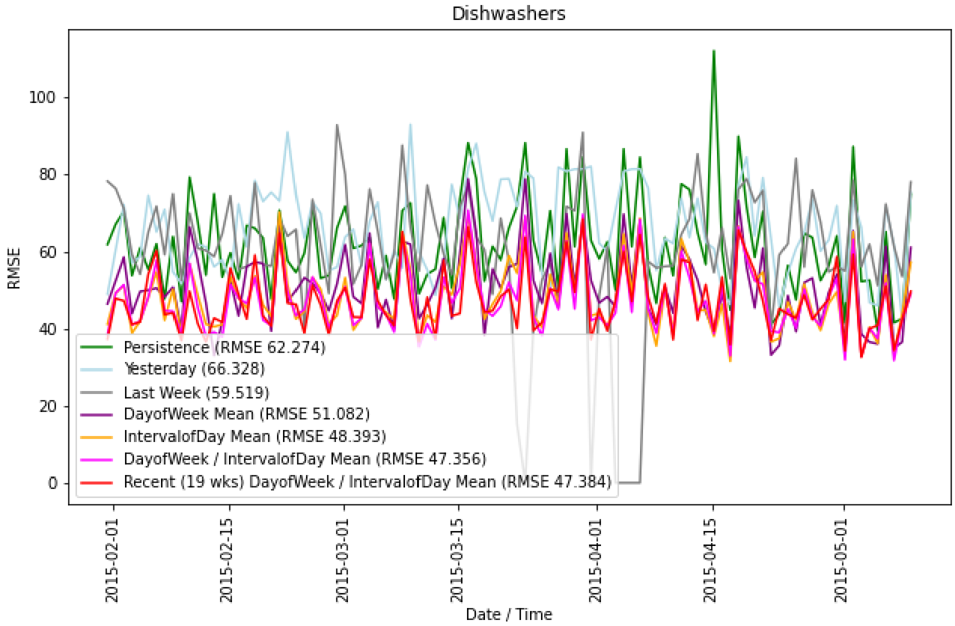


Figure G.2: Dishwasher daily mean RMSE of the naive forecast vs the actual signal over the test set. We note some of the zero values for the "Last Week" forecast. These are an artifact of the imputation strategy where missing data were imputed from the same day, 1 week prior. For these particular dates no data were collected for any of the appliances, the values were imputed from one week prior (hence the zero RMSE) and they all show the same artifact.

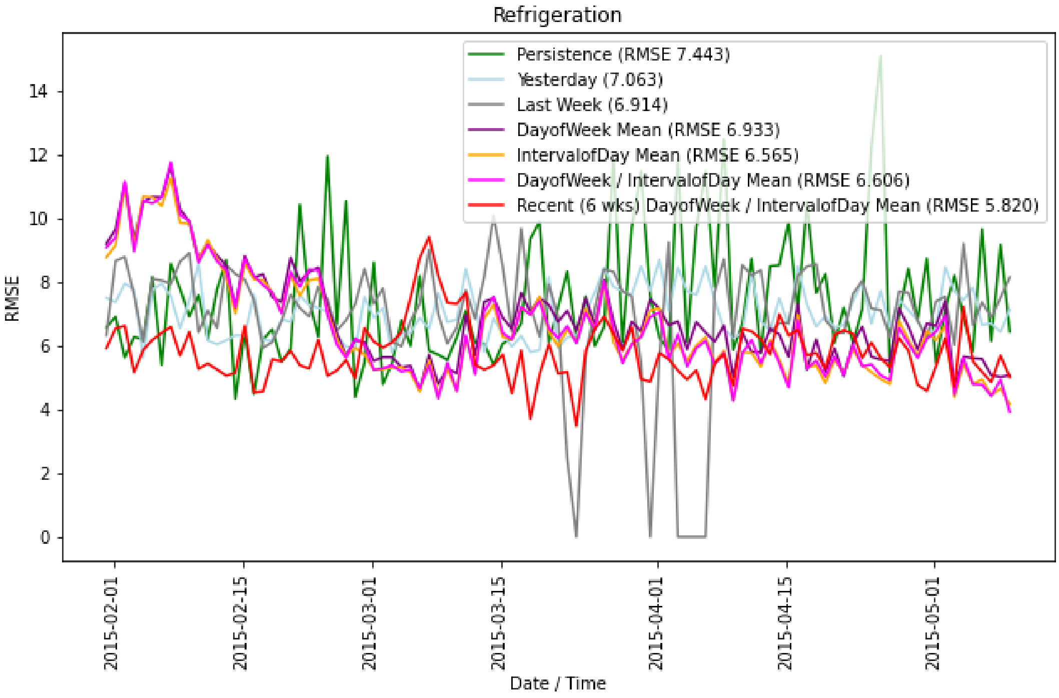


Figure G.3: Refrigeration daily mean RMSE of the naive forecast vs the actual signal over the test set.

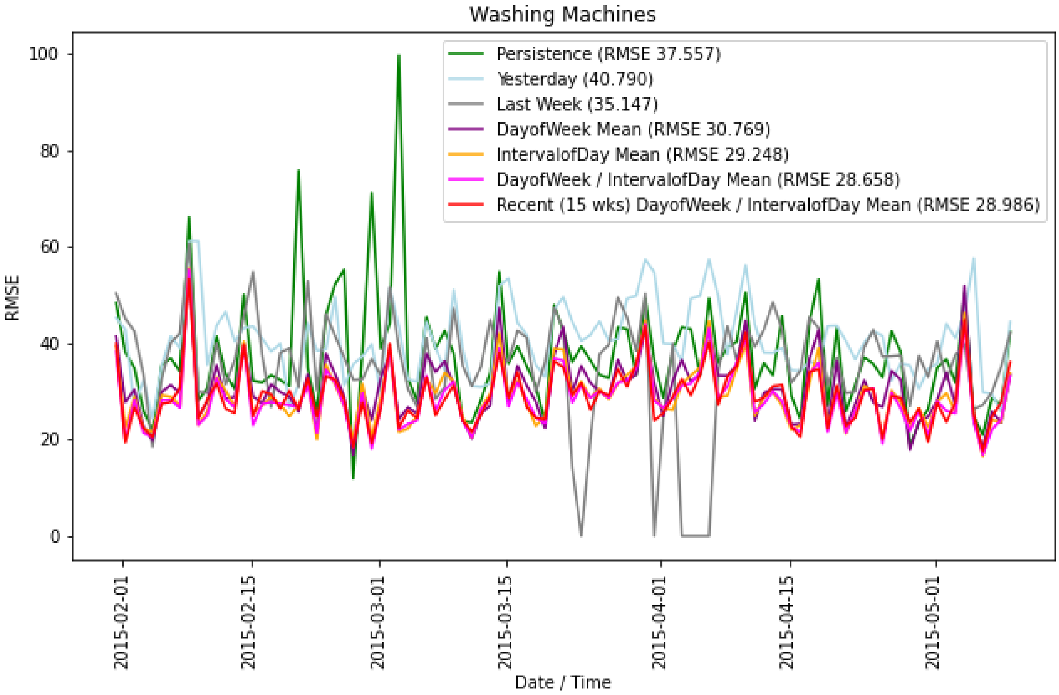


Figure G.4: Washing Machine daily mean RMSE of the naive forecast vs the actual signal over the test set.

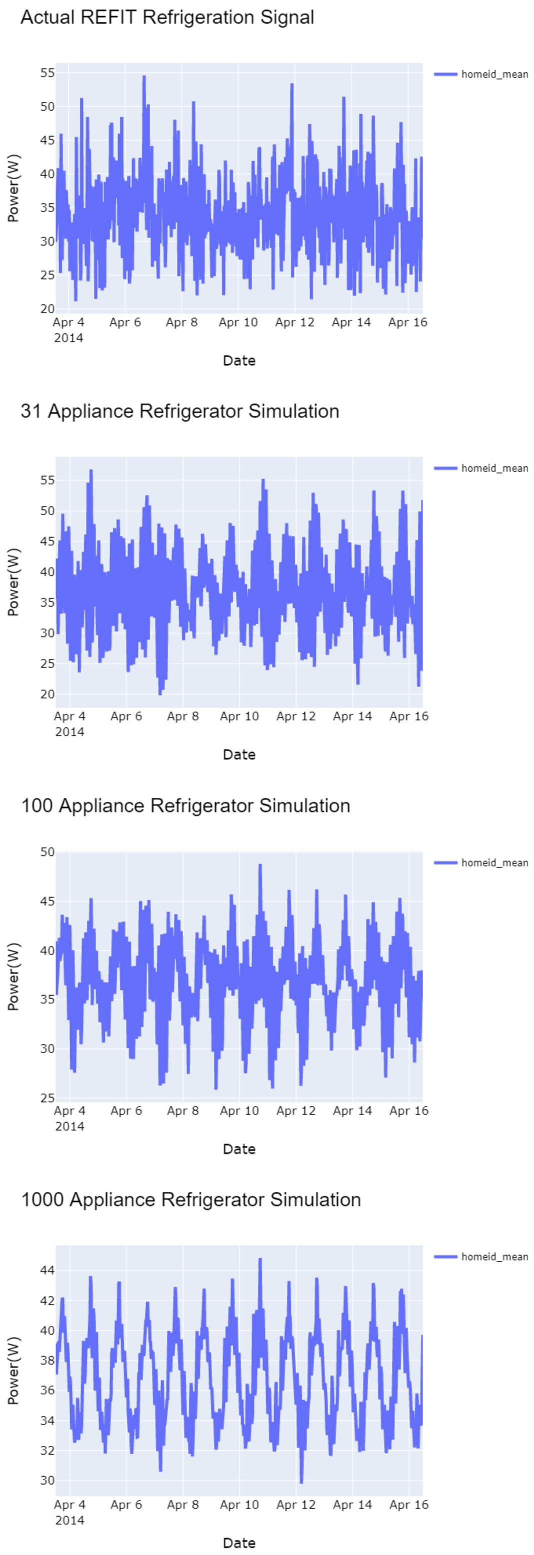
Appendix H

Refrigeration Simulator

Fridges cycle on and off through the day according to the ambient environment temperature (how much power it takes to cool to a fixed internal temperature while the temperature around it is varying) and how much the door is opened and warm things are put inside to cool (according to our EDA, these occur around mealtimes - 7am, 1pm and 6pm). When a fridge compressor is running it consumes approximately a constant power (start-up transients are ignored). When it is off we assume zero power. Different fridges (size, age etc) have different active powers and cycle with different duty cycles. We model each of these by sampling from a normal distribution where we have specified the mean and the standard deviation.

We used the scipy.signal package to generate a square wave with pulse width modulation according to a sinusoid. The frequency of the square wave, the magnitude of the "on" power and the duty cycle were all sampled from normal distributions. The mean and sdev's of these distributions were found empirically by studying the raw time series and with spectral plots. The "blip" in power consumption around mealtimes was modelled by increasing the duty cycle by a "Mealtime Interaction Factor" for 30mins at these specific times.

- Mean active power: 73W, stdev: 0
- Mean Cycle period: 2.5 Hours, Stdev: 0.25 Hours
- Mean Duty Cycle: 0.2, Stdev: 0.1
- Mealtime Interaction Factor: 0.7



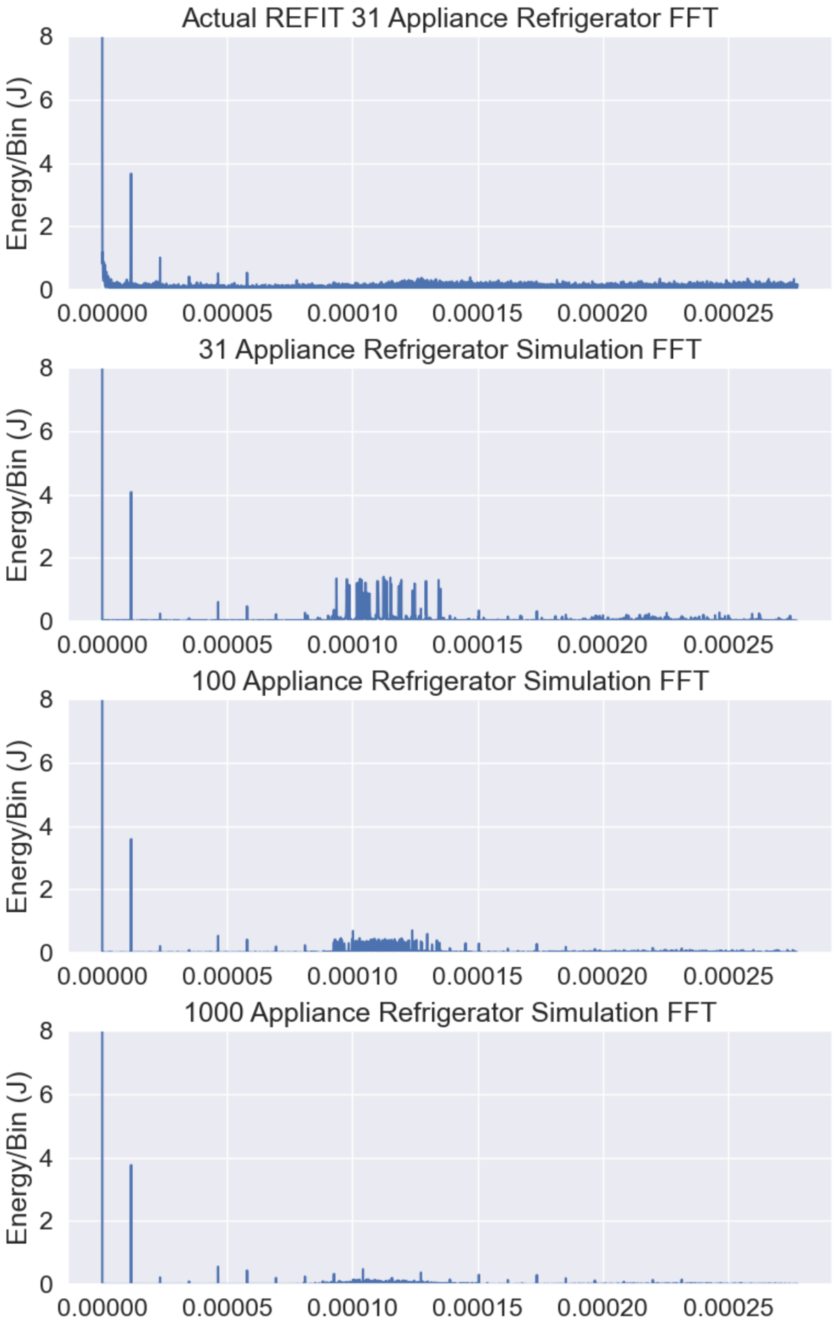
Actual REFIT Refrigeration Signal

31 Appliance Refrigerator Simulation

100 Appliance Refrigerator Simulation

1000 Appliance Refrigerator Simulation

Figure H.1: The actual REFIT signal and then 31, 100 and 1000 Simulation Signals. The aggregation error / noise reduces with the higher number of aggregations to finally reveal the underlying signal we're forecasting.



Actual REFIT 31 Appliance Refrigerator FFT

31 Appliance Refrigerator Simulation FFT

100 Appliance Refrigerator Simulation FFT

1000 Appliance Refrigerator Simulation FFT

Figure H.2: Spectral content of the actual REFIT signal and then 31, 100 and 1000 Simulation Signals.