

Article

Energy Consumption Patterns and Load Forecasting with Profiled CNN-LSTM Networks

Kareem Al-Saudi  ^{1,*}, Viktoriya Degeler  ¹ and Michel Medema  ¹

¹ Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence; University of Groningen, Netherlands

* Correspondence: KareemAlSaudi13@gmail.com

1 Abstract: By virtue of the steady societal shift to smart technologies, built on the increasingly popular smart grid framework, we have noticed an increase in the need to analyze household electricity consumption at the individual level. In order to work efficiently, these technologies rely on load forecasting to be able to optimize operations that are related to energy consumption (such as household appliance scheduling). This paper proposes a novel load forecasting method that utilizes a clustering step prior to the forecasting step to group together days that exhibit similar patterns in energy consumption. Following that, we attempt to classify new days into one of the pre-generated clusters by making use of the available context information (day of the week, month, predicted weather). Finally, using available historical data (with regards to energy consumption) alongside meteorological and temporal variables, we train a CNN-LSTM model on a per-cluster basis that each specializes in forecasting based on the energy profiles present within each cluster. This method leads to improvements in forecasting performance (upwards of a 10% increase in mean absolute percentage error scores) and provides us with the added benefit of being able to easily highlight and extract information that allows us to assess which external variables have an effect on the energy consumption of any individual household.

16 Keywords: Pattern Recognition, Energy Profiling, Clustering, Forecasting

1. Introduction

18 Over the years, our reliance on electrical appliances has been slowly increasing.
 19 As our dependence on electrical appliances increases, so too does our consumption of
 20 energy [1] and, subsequently, our need for more sophisticated and advanced solutions
 21 that can accommodate this growth. Thankfully, the convergence of multiple technologies
 22 – such as machine learning, data mining and ubiquitous computing – has led to the rise
 23 of a solution in the form of *smart (electric) grids* as well as *smart environments* and *smart*
 24 *meters* that are slowly but surely taking off in terms of their popularity and availability
 25 [2]. The resulting growth in the prevalence of smart grids gives us the opportunity
 26 to both control and monitor the energy consumption of individual households on a
 27 real-time basis [3], and, through the utilization of applications built upon this framework,
 28 we are capable of achieving an overall reduction in terms of the amount of energy that
 29 we, as the human race, consume. This opens up the possibility to alleviate some of the
 30 inherent risks associated with the growth in energy consumption, whether that be our
 31 overall environmental footprint on the planet or, on a much smaller scale, the financial
 32 impact on both suppliers as well as consumers due to instabilities present in current,
 33 outdated power grid systems [4].

34 Existing solutions developed under the increasingly popular smart grid framework, such
 35 as the Home Energy Management System (HEMS) and Battery Energy Management
 36 System (BEMS), aim to provide the end-user with the means to schedule, or otherwise
 37 manage, daily appliance operations, taking into consideration external factors such as

Citation: Al-Saudi, K.; Degeler, V.; Medema, M.; Energy Consumption Patterns and Load Forecasting with Profiled CNN-LSTM Networks.

Processes **2021**, *1*, 0. 0

Received:

Accepted: -

Published: -

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Submitted to *Processes* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (CC BY 4.0).

39 weather conditions, utility tariff rates alongside any other personal preferences [3]. To
40 operate efficiently, these solutions rely on our ability to capably forecast future trends
41 in energy consumption at the individual household level. This information is required
42 to appropriately and sufficiently control and supply the correct energy load to the end-
43 user [5,6]. This has lead to a shift in interest within the realm of load forecasting, in
44 which prior research has predominantly been focused at the large-scale, regional level
45 [7] where an amalgamation of available data spanning numerous households provides
46 more obvious patterns as a result of the underlying diversity between households being
47 lost [8] towards the individual household level. Furthermore, owing to the operational
48 characteristics of both HEMS and BEMS and similar applications, load forecasting in the
49 very short term (anywhere from a few minutes to a couple of hours), oftentimes referred
50 to as very short-term load forecasting (VSTLF), are more relevant than the substantially
51 studied longer term horizons that are predominantly associated with long-term network
52 planning and operations [3].

53
54 When exploring energy consumption at the individual household level, the diversity
55 and complexity associated with human behavior leads to extremely dynamic, volatile
56 patterns that can prove to be highly dissimilar between households. In addition to this,
57 certain households exhibit no clear pattern in energy consumption due to a high level of
58 irregularity in the lifestyle of its occupants [8]. To account for this dissimilarity, current,
59 state-of-the-art methods benefit from a precursory clustering step within the forecasting
60 pipeline [3,4,8]. This precursory clustering step serves to amalgamate days that exhibit
61 a measure of similarity in terms of their energy consumption patterns into the same
62 cluster. By training individual forecasting models on a per-cluster basis we should, in
63 theory, see an improvement in load forecasting performance as each of the respective
64 models specializes in predicting future trends in energy consumption based on patterns
65 present within the energy profile associated with its unique cluster. This is the area of
66 research that this paper seeks to tackle – how can we best construct energy profiles out of
67 historical data that truly capture repeated patterns with regards to energy consumption
68 and what are the effects of a clustering step in the performance of a forecasting pipeline.

69 2. Use Case Description

70 At our disposal are a number of publicly available data sets that contain historical
71 data with regards to energy consumption. These include the data collected by the
72 Engineering and Physical Sciences Research Council via the project entitled "*Personalised*
73 *Retrofit Decision Support Tools for UK Homes using Smart Home Technology (REFIT)*" [9]
74 which is a collaboration among the Universities of Strathclyde, Loughborough and East
75 Anglia, as well as the "*Individual Household Electric Power Consumption*" data set [10]
76 that is part of the University of California, Irvine Machine Learning Repository and that
77 will henceforth be acronymized as the "*UCI data set (UCID)*". This section will serve to
78 briefly describe the main aspects of each of these individual data sets so that we may
79 be better able to draw comparisons between them and highlight any key differences.
80 Additionally, we aim to append meteorological features (e.g., temperature, wind speed,
81 cloud coverage, precipitation) to each of our respective data sets – an overview of this
82 process and the data that we will be utilizing will also be presented in this section.

83 2.1. REFIT

84 The REFIT Electrical Load Measurements data set [9] includes cleaned electrical
85 consumption data, in watts, for a total of 20 households labeled *House 1 - House 21*
86 (skipping House 14) located in the Loughborough area, a town in England, over the
87 period of 2013 through early 2015. The electrical consumption data is collected at both
88 the aggregate level as well as the appliance level with each household containing a total
89 of 10 power sensors that comprise of a current clamp for the household aggregate labeled
90 as *Aggregate* in the data set as well as 9 individual appliance monitors (IAM) labelled as

91 Appliance 1 - Appliance 9 in the data set. The appliance list associated with each of the
92 IAMs differs between households and comprise a measure of ambiguity as applicants
93 may have switched appliances around during the duration of the data collection and
94 the installation team responsible for setting up the power sensors did not always collect
95 relevant data associated with said IAMs. The consequences of this is of course that we
96 do not know with 100% certainty whether an appliance or set of appliances associated
97 with an IAM is the same throughout the entirety of the data set. Additionally, some
98 labels are inherently ambiguous taking, for example, the *television site* label which could
99 comprise of any number of appliances including: a television, DvD player, computer,
100 speakers etc. Finally, the makes and models of the appliances that were meant to be
101 collected by the installation team are not always present, further compounding on the
102 previously mentioned uncertainties.

103
104 The documentation associated with the data set states that active power is collected, and
105 subsequently recorded, at an interval of 8 seconds; however, a cursory glance at the data
106 demonstrates that this is not always the case. A potential reason for this could be the
107 fact that the aforementioned power sensors are not synchronized with the associated
108 collection script which polls within a range of 6 to 8 seconds leaving us with a margin for
109 error in the intervals between recorded data samples. Moreover, the data set is riddled
110 with long periods of missing data making it exceptionally difficult to work with. All of
111 that said, the data collection team made an attempt to pre-process or otherwise *clean* the
112 data set by:

- 113 1. Correcting the time to account for the United Kingdom daylight savings.
- 114 2. Merging timestamp duplicates.
- 115 3. Moving sections of IAM columns to correctly match the appliance they were
116 recording when said appliance was reset or otherwise moved.
- 117 4. Forward filling NaN values or zeroing them depending on the duration of the time
118 gap.
- 119 5. Removing spikes of greater than 4,000 watts from the IAM values and replacing
120 them with zeros.
- 121 6. Appending an additional issues columns that is set to 1 if the sum of the sub-
122 metering IAMs is greater than that of the household aggregate – in this case, data
123 should either be discarded or, at the very least, the discrepancy must be noted.

124 2.2. UCID

125 The UCID data set [10] contains a total of 2,075,259 measurements gathered in a
126 single house located in Sceaux, a commune in the southern suburbs of Paris, France. The
127 data within this data set was recorded throughout a duration of 47 months spanning the
128 period between December 2006 and November 2010. Measurements were made approx-
129 imately once a minute and consist of the minute-averaged active power consumption, in
130 kilowatts, within the entire household as well as 3 energy sub-metering measurements
131 that correspond to the kitchen, which includes a dishwasher and microwave, the laundry
132 room that consists of a washing machine and tumble dryer, and the combination of both
133 an electric water-heater as well as an air-conditioner respectively. The UCID data set is
134 not without fault either, containing approximately 25,979 missing measurements which
135 make up roughly 1.25% of the entire data set; however, given the extensive range covered
136 as well as the immense number of total measurements available on hand these missing
137 values can easily be disregarded and subsequently discarded during the preprocessing
138 stage of our forecasting pipeline.

139 2.3. Meteorological Data

140 As an addendum to both the REFIT and UCID data sets we will be incorporating
141 meteorological data as provided by Solcast [11], a company based in Australia that aims
142 to provide high quality and easily-accessible solar data. For the purpose of this master's

¹⁴³ thesis project we requested meteorological data in variable time resolutions (5, 10, 15
¹⁴⁴ minutes) for both the Loughborough area in the United Kingdom for the REFIT data set
¹⁴⁵ as well as meteorological data for the Sceaux commune in the southern suburbs of Paris,
¹⁴⁶ France for the UCID data set. The relevant periods are the 16th of September, 2013 up to
¹⁴⁷ and including the 11th of July, 2015 and the 1st of December, 2006 up to and including
¹⁴⁸ the 30th of November, 2010 for each data set respectively. The provided data is extensive,
¹⁴⁹ covering a wide range of parameters that are listed, and described in detail, in Table 1.

| Parameter | Description |
|---|---|
| Air Temperature | The air temperature (2 meters above ground level). Units in Celsius. |
| Albedo | Average daytime surface reflectivity of visible light, expressed as a value between 0 and 1. 0 represents complete absorption. 1 represents complete reflection. |
| Azimuth | The angle between a line pointing due north to the sun's current position in the sky. Negative to the East. Positive to the West. 0 at due North. Units in degrees. |
| Cloud Opacity | The measurement of how opaque the clouds are to solar radiation in the given location. Units in percentage. |
| Dewpoint | The air dewpoint temperature (2 meters above ground level). Units in Celsius. |
| Direct Normal Irradiance | Solar irradiance arriving in a direct line from the sun as measured on a surface held perpendicular to the sun. Units in W/m ² . |
| Direct (Beam) Horizontal Irradiance | The horizontal component of Direct Normal Irradiance. Units in W/m ² . |
| Global Horizontal Irradiance | The total irradiance received on a horizontal surface. It is the sum of the horizontal components of direct (beam) and diffuse irradiance. Units in W/m ² . |
| Global Tilted Irradiance – Fixed | The total irradiance received on a surface with a fixed tilt. The tilt is set to latitude of the location. Units in W/m ² . |
| Global Tilted Irradiance – Horizontal Single-Axis Tracker | The total irradiance received on a sun-tracking surface. Units in W/m ² . |
| Precipitable Water | The total column precipitable water content. Units in kg/m ² . |
| Relative Humidity | The air relative humidity (2 meters above ground level). Units in percentage. |
| SFC pressure | The air pressure at ground level. Units in hPa. |
| Snow Depth | The snow depth liquid-water-equivalent. Units in cm. |
| Wind Direction | The wind direction (10 meters above ground level). This is the meteorological convention. 0 is a northerly (from the north); 90 is an easterly (from the east); 180 is a southerly (from the south); 270 is a westerly (from the west). Units in degrees. |
| Wind Speed | The wind speed (10 meters above ground level). Units in m/s. |
| Zenith | The angle between a line perpendicular to the earth's surface and the sun (90 deg = sunrise and sunset; 0 deg = sun directly overhead). Units in degrees. |

Table 1: List of meteorological parameters available to us as per the Solcast data sets.

150 3. Proposed Model

151 To attempt to solve the previously outlined problem of VSTLF forecasting at an
152 individual household level, we propose a novel solution that utilizes a combination of
153 statistical knowledge and machine learning techniques to both generate energy profiles
154 that provide us with some measure of insight as to the habits of a household's occupants
155 as well as forecast future trends in their energy consumption. A high level overview of
156 the steps relevant to our proposed model can be seen in Figure 1.

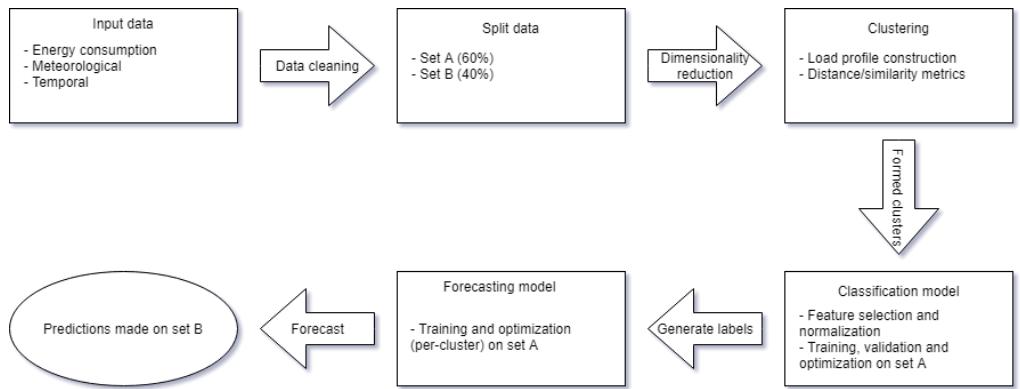


Figure 1. High level overview of the steps pertaining to our model.

157 In short, we devised a method that consists of 3 steps: cluster, classify, forecast – first we
158 cluster historical days based on similarity in terms of active power consumption, then
159 we classify new days into one of the generated clusters and finally we generate forecasts
160 based on models that are trained on a per-cluster basis.

161 4. Related Work

162 Energy management systems, such as the previously introduced HEMS and BEMS,
163 are designed with the intent to both optimize and control the smart grid energy market.
164 As previously stated, to be able to do this, these demand-side management systems
165 require a priori knowledge about the load patterns and, as a result of this, the field
166 of designing computationally intelligent load forecasting systems has expanded quite
167 rapidly in recent years with over 50 research papers related to the subject having been
168 identified in existing literature [12]. In this chapter we will be exploring a compiled
169 subset of this literature that specifically tackle the problem of energy profile construction
170 as well as load forecasting. This is done so as to establish a baseline of understanding
171 as to what has already been done within the field in terms of the two focal points of
172 our forecasting pipeline: the precursory clustering step as well as the state-of-the-art
173 forecasting models. Furthermore, by doing so we will be able to position our paper with
174 respect to the current state-of-the-art and highlight the key differences in our approach.

175 4.1. Clustering and Energy Profile Creation

176 The main issue that this paper seeks to address is that of creating interesting profiles
177 in terms of recurrent patterns in energy consumption. To do this, we will be making
178 use of clustering algorithms that seek to partition our data into a number of clusters
179 so that each of these clusters exhibit some metric of similarity or *goodness*. However, a
180 measure of goodness can inherently be seen as quite subjective with Backer and Jain [13]
181 noting that, "in cluster analysis a group of objects is split up into a number of more or
182 less homogeneous subgroups on the basis of an often subjectively chosen measure of
183 similarity (i. e., chosen subjectively based on its ability to create "interesting" clusters)
184 such that the similarity between objects within a subgroup is larger than the similarity
185 between objects belonging to different subgroups.". We will be exploring papers in the

¹⁸⁶ existing literature that present different takes both in how they define similarity as well
¹⁸⁷ as their chosen clustering methodology.

¹⁸⁸

¹⁸⁹ Kong *et al.* [8] attempted to justify the observations made by Stephen *et al.* [14] by using
¹⁹⁰ a density-based clustering technique known as Density Based Spatial Clustering of
¹⁹¹ Applications with Noise (DBSCAN) [15] to evaluate consistency in short-term load
¹⁹² profiles. They remark on the benefits of using DBSCAN, stating that, as it does not
¹⁹³ require knowing the number of clusters in the data ahead of time and as it contains the
¹⁹⁴ notion of outliers, it would be an ideal clustering technique to identify consumption
¹⁹⁵ patterns that repeat with a measure of noise akin to what is loosely defined by Practice
¹⁹⁶ Theory. Their findings are that the number of clusters as well as outliers varies greatly
¹⁹⁷ between households with some households exhibiting no clearly discernible patterns and
¹⁹⁸ some households (mostly) following fairly consistent daily profiles. This is visualized in
¹⁹⁹ Figures 2a and 2b.

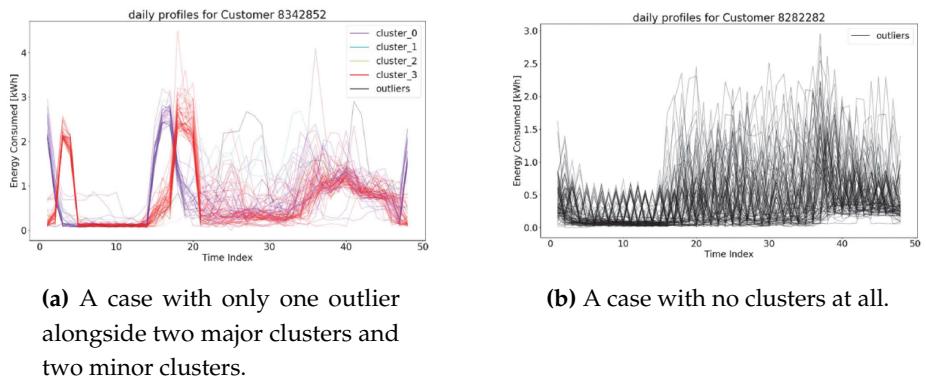


Figure 2. Two widely different scenarios in the application of DBSCAN. Images source: [8] © 2019, IEEE.

²⁰⁰ Yildiz *et al.* [3] expand on traditional load forecasting techniques, such as the Smart
²⁰¹ Meter Based Model (SMBM) that they had previously presented [16], and present their
²⁰² own take in the form of a Cluster-Classify-Forecast (CCF) model. In traditional SMBMs,
²⁰³ a chosen model, whether that be of a statistical variant or from the plethora of existing
²⁰⁴ machine learning models, learns the relationship between the target forecasted loads
²⁰⁵ when presented with some input data which, in our case, consists of some historical
²⁰⁶ lags in terms of energy consumption, data with regards to the weather and temporal
²⁰⁷ information with respect to the time, calendar date etc. The CCF takes this a step
²⁰⁸ further by making use of both K-means and Kohonen's Self-Organizing Map [17] to
²⁰⁹ group profiles that are most similar to each other. After obtaining and validating the
²¹⁰ output of their chosen clustering techniques they investigate the relationship between
²¹¹ the clustering output and other temporal variables, such as the weather, by using a
²¹² Classification and Regression Tree [18].

²¹³ 4.2. Forecasting Models

²¹⁴ Numerous studies have been conducted with the intent to forecast energy consump-
²¹⁵ tion which range from methods the likes assessed by Fumo and Rafe Biswas [19] in the
²¹⁶ form of multiple linear regression to methods such as novel deep pooling Recurrent
²¹⁷ Neural Network introduced by Shi *et al.* [20]. The majority of these forecasting models,
²¹⁸ whether they be statistical, machine learning or deep learning based, can be classified
²¹⁹ into 2 main categories: single technique models in which only a single, heuristic algo-
²²⁰ rithm (e.g., a Multi-Layer Perceptron or Support Vector Machine) is used as the primary
²²¹ forecasting method and hybrid methods that encapsulate 2 or more algorithms [12] such
²²² as the Convolutional Neural Network Long Short-Term Memory (CNN-LSTM).

223 Kong *et al.* [8] employ the use of a Long Short-Term Memory (LSTM) network as it is
224 generally the ideal candidate when attempting to learn temporal correlations within
225 time series data sets; however, their final results are not very promising boasting a mean
226 absolute percentage error (MAPE) of approximately 44% over variable time steps. This
227 could be a result of poor hyperparameter tuning stating that, "*tuning 69 models for each of*
228 *the candidate methods is very time-consuming for this proof-of-concept paper*" leading us to
229 believe that there is definitely room for improvements to be made on the core concepts
230 of their work.

231
232 Yildiz *et al.* [3] use the clusters they formed as described earlier alongside their assign-
233 ments to build SMBMs, in this case through the use of a Support Vector Regression
234 model, and find that, alongside improvements to load forecasting accuracy, they are able
235 to reveal vital information on the habitual load profiles of the households they were
236 exploring. Unfortunately, they do not indicate any potential reasoning as to why they
237 chose to use K-means and Kohonen's SOMs in place of potentially more effective cluster-
238 ing methods citing only that K-means is the most popular clustering technique [18] and
239 that SOMs is generally used as an extension to neural networks for the purposes of clus-
240 tering. Additionally, their results only include values that are indicative of their chosen
241 technique's performance on their specific data set presenting performance metrics such
242 as normalized root mean square error (NRMSE) and normalized mean absolute error
243 (NMAE) rendering us unable to compare the performance of their proposed method.

244
245 Kim and Cho [21] present a more modern take on load-forecasting proposing a hybrid
246 CNN-LSTM network that is capable of extracting both temporal and spatial features
247 present in the data. The use of convolutional layers within the realm of load forecast-
248 ing is brilliant allowing for the network to take into account the correlation between
249 multivariate variables while minimizing noise that can eventually be fed into the LSTM
250 section of the network that finally generates predictions. Their paper proposes such a
251 network citing that the major difficulties with such an approach mainly boil down to
252 hyperparameter tuning which can be remedied through a variety of means that include
253 the likes of genetic algorithms or through the use of packages such as Keras Tuner
254 maintained by O'Malley *et al.* [22]. Furthermore, Kim and Cho [21] did not explore the
255 possibility of implementing a precursory clustering step which could have lead them to
256 substantial improvements in their final MAPE.

257 5. Materials and Methods

258 This paper proposes a forecasting method that utilizes dimensionality reduction
259 and clustering techniques to group days that exhibit similarity in terms of electric
260 consumption behavior. Days that are grouped into the same cluster are thought to
261 contain shared features, whether those features be temporal, as seen in Table 2, or
262 meteorological or otherwise, that cause this similarity in behavior. The formed clusters
263 (per household) are used for 2 purposes: firstly, they will be used to train a classification
264 model that utilizes available context information to assign a new day to the correct cluster.
265 Secondly, and finally, a novel deep learning method will be applied on a per-cluster basis
266 to forecast future energy consumption.

267 In short, we start off by resampling the data present in both the REFIT data set as well
268 as the UCID into a common resolution in order be able to directly compare results. In
269 this case we will be resampling both data sets to a common resolution of 15 minutes per
270 sample as this lines up well with the native resolution of the meteorological data we
271 have on hand that was provided to us by Solcast. Following this, we clean both data
272 sets and rid them of any days that contain an incomplete number of records (incomplete
273 here referring to any days that contain less than 96 records given we split each day into a
274 total of 96 chunks). After this, we take a subset of each of our data sets (60% of the total
275 data for each of the REFIT as well as the UCID data sets henceforth referred to as *Set A*)

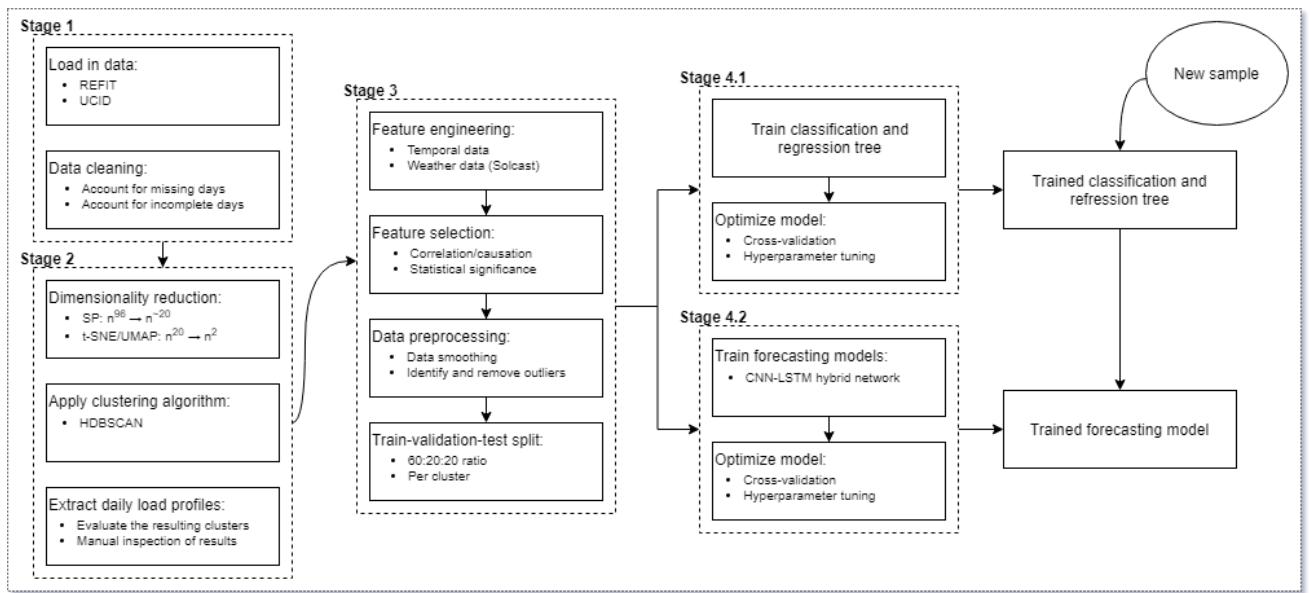


Figure 3. Proposed daily profile extraction and load forecasting model.

and leave out the remaining 40% (henceforth referred to as *Set B*) in order to validate the results of our forecasting model. We then reduce the overall dimensionality of a single day, going from a total of 96 features to a much more manageable 2 through a combination of statistical and machine learning techniques and generate clusters based on the new, 2-dimensional data set by utilizing a density-based clustering algorithm. Following this, we train and optimize a classifier on Set A and use it to generate cluster labels for the previously withheld Set B. Finally, we train our forecasting models, one per cluster, on the data present in Set A and use the data present in Set B to act as both a validation set as well as a training set with which to obtain final results. In this Chapter, we will provide a working example alongside in-depth explanations of each of the steps previously discussed that make up our overall forecasting pipeline.

5.1. Stage 1 - Data Collection and Cleaning

As mentioned previously, this paper utilizes available historical data with regards to energy consumption on an individual household basis. In reality, as part of stage 1 of our forecasting pipeline, time series data of daily electricity consumption would need to be collected from an individual household meter for an adequate amount of time at an ideal resolution so as to obtain acceptable results. After collecting, or in our case loading in, the data, we perform common preprocessing techniques to account for noisy or otherwise missing data that occurred during the transmission of the data from the meters. In our case, the available data was resampled into a resolution of 15 minutes and any days that contained less than 96 values (given that there are 96 15 minute chunks in a day) were dropped from our data set. All other days that contained NaN values were also not considered and subsequently dropped from our data set.

5.2. Stage 2 - Dimensionality Reduction and Clustering

Given that each day in our data set is represented by 96 dimensions, each dimension comprising mean active power consumption over a time period of 15 minutes, the first logical step to undertake would be to transform the data in a manner that enables our clustering techniques to more efficiently determine which days exhibit similarity in terms of electric consumption behavior. This *dimensionality reduction* step comprises 2 parts: To start things off we divide each day into 5 different periods (as per the work of Yildiz *et al.* [3]) as follows:

307 1. Morning: 06:00 - 11:00
308 2. Late morning/afternoon: 11:00 - 15:00
309 3. Late afternoon/early evening: 15:00 - 20:30
310 4. Evening: 20:30 - 23:30
311 5. Late evening/early morning: 23:30 - 06:00

312 Following that, we represent each period by its respective mean, minimum, maximum
313 value as well as its standard deviation. The outcome of performing this is that each
314 day is now represented by a total of 20 dimensions rather than the initial 96 which is
315 a reduction of $\sim 80\%$. We can reduce this even further, and even visualize our data
316 in 2 or 3 dimensions, by making use of either of the t-Distributed Stochastic Neighbor
317 Embedding (t-SNE) [23] or Uniform Manifold Approximation and Projection (UMAP)
318 [24] algorithms. The most important hyperparameter to tune for either algorithm is
319 the *perplexity* hyperparameter for the t-SNE algorithm and the equivalent $n_{neighbors}$
320 hyperparameter for the UMAP algorithm. During our research, we found that an
321 optimal value for either of these hyperparameters is $N^{\frac{1}{2}}$ where N is the number of
322 samples present in the data set. To better understand each of the steps of our proposed
323 model, we will now begin a series of visualizations showcasing each step as performed
324 on the UCID data set throughout the entirety of stage 2 as well as the remainder of the
325 stages that make up our overall forecasting pipeline. We start off by presenting a scatter
326 plot of the 2-dimensional output obtained as a result of performing the t-SNE algorithm
327 which can be seen in Figure 4a that allows us to clearly visualize the 2-dimensional
328 interpretation of the samples present in the UCID data set. Each of the points found on
329 the 2-dimensional surface in Figure 4a represents a single day, and given that the UMAP
330 algorithm claims to preserve both local as well as most of the global structure present in
331 the data we can safely assume that distances between the samples are conducive to the
332 similarity in terms of energy consumption as per the previously segmented (into various
333 periods) interpretation of the data.

334

335 After performing the dimensionality reduction step on our data, we proceed to cluster
336 the resulting output by applying the Hierarchical Density Based Spatial Clustering
337 of Applications with Noise (HDBSCAN) algorithm. As previously mentioned, the
338 only important parameters that need to be passed to the HDBSCAN algorithm are the
339 minimum size we expect each cluster to be. In this case we set that value to $\frac{1}{10}(N)$ where
340 N is the number of samples present in the data set. Our reasoning for selecting this
341 value is predominantly based on the adequate results observed by Kong *et al.* [8] in
342 their implementation of the DBSCAN algorithm in a similar setting whilst utilizing a
343 similar selection in terms of hyperparameter settings. The other hyperparameter we
344 choose to tune is the *min_samples* hyperparameter, which, in layman's terms, denotes
345 how conservative we would like to be with our clustering in terms of restricting clusters
346 to progressively more dense areas and classifying samples from our data set as noise. In
347 our case, an arbitrary value of 15 was selected, in contrast to the default value that sets
348 *min_samples* = *min_cluster_size*. The results of performing the HDBSCAN algorithm
349 on our 2-dimensional representation of the UCID data set (shown in Figure 4a) can be
350 seen in Figure 4b.

351

352 For the sake of comparison, we present the output of applying the k-means clustering
353 algorithm (assuming $k = 3$) on the same 2-dimensional representation of the UCID
354 data set. This can be seen in Figure 4c. We note immediately the capability of the
355 HDBSCAN algorithm in capturing a better representation of the clusters present in
356 our 2-dimensional representation of the UCID data set. The representation of outliers
357 as noise points and not having to have a priori knowledge on the number of clusters
358 present in the data we are working with is a definite pro as well further compounding
359 our choice of clustering algorithm in our proposed model.

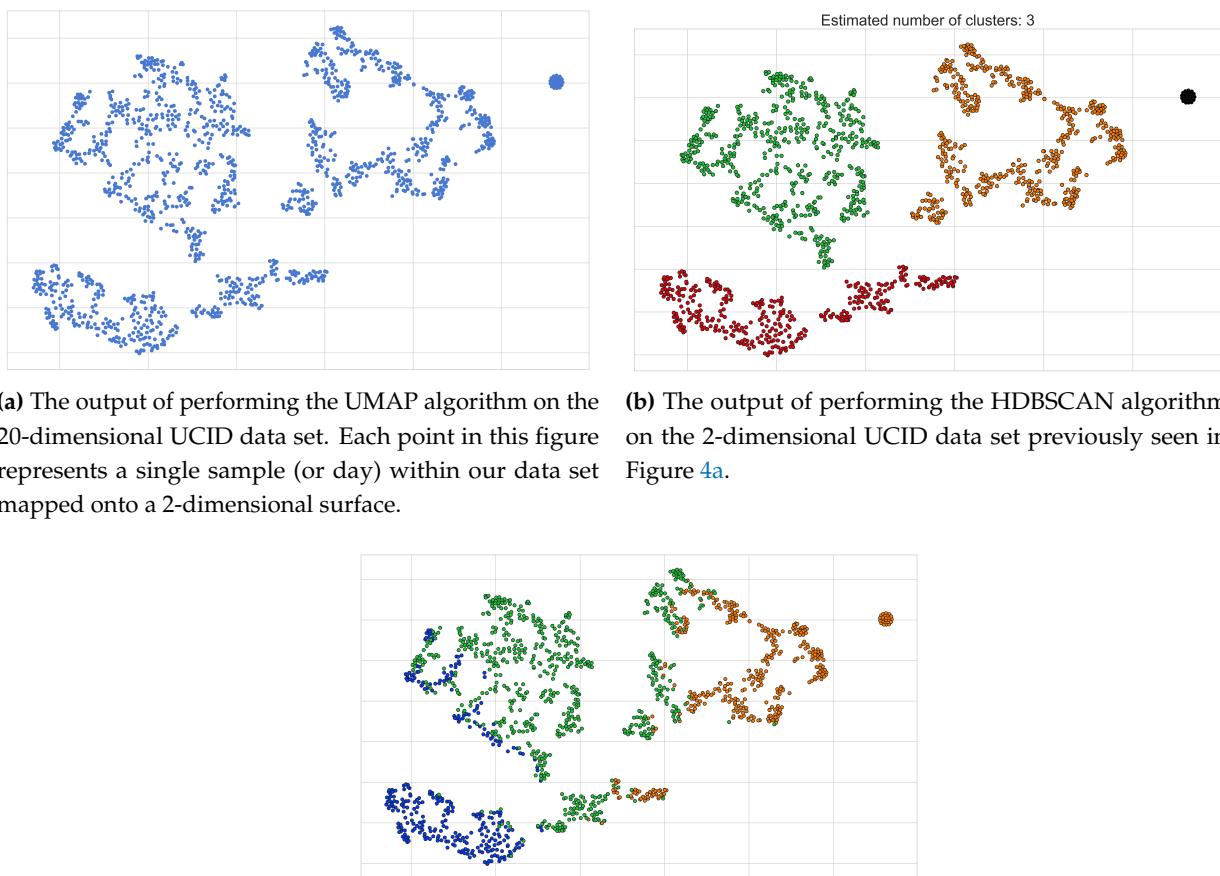
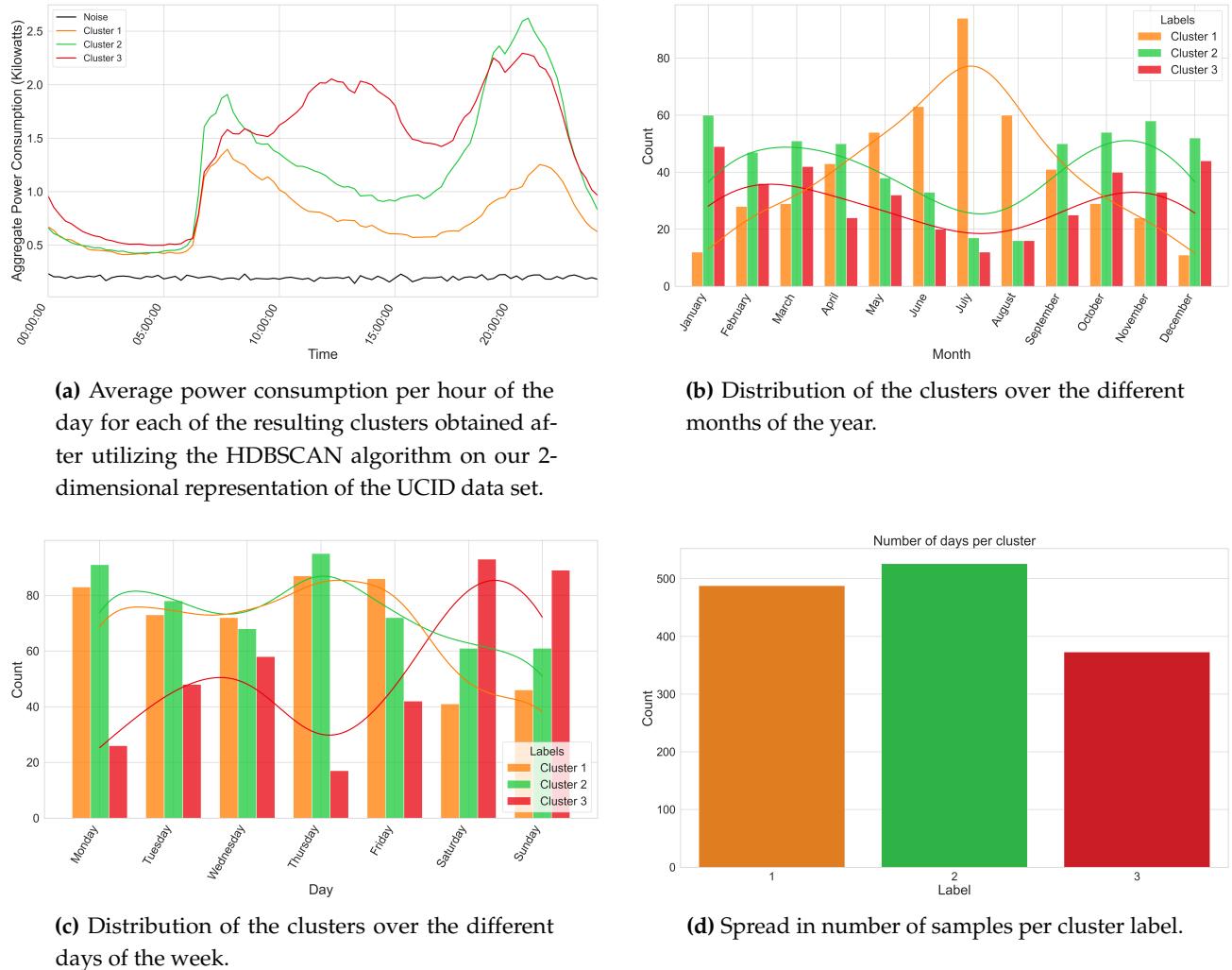


Figure 4. Output of performing the various steps associated with Stage 2 of our model.

360 Visualizing, or otherwise manually inspecting, the clusters we obtain as a result of our
 361 application of the HDBSCAN algorithm is necessary so that we can better understand
 362 whether our clustering algorithm truly captures the habits of the individuals residing
 363 in the households we are working with. The first step in our analysis of the resulting
 364 clusters would be to plot the averaged power consumption on a per cluster basis so that
 365 we may be able to clearly visualize the patterns in power consumption per cluster. An
 366 example of this, in line with the previous examples showcasing our proposed model
 367 on the UCID data set, can be seen in Figure 5a. We note that, in this example, a subset
 368 of our data (24 samples in total), were recorded as noise by the HDBSCAN algorithm.
 369 Inspecting these samples manually lead us to the confirmation that, of the 4 year's worth
 370 of data, these 24 days were the only days that exhibited no tangible shift in terms of
 371 power consumption throughout the entirety of the day (i.e., the global active power
 372 draw observed was completely stationary throughout this period); however, this is not
 373 explicitly outlined in the documentation of the UCID data set. This can be seen as a more
 374 or less flat line in Figure 5a.
 375 Following this, Figures 5b and 5c help us visualize the distribution of the clusters over
 376 the months of the year as well as the days of the week to ascertain whether any of the
 377 clusters present any correlation with these temporal variables. Given that the initial
 378 spread of the data throughout the months of the year and days of the week of the UCID
 379 were relatively uniform, we should not see any bias towards any particular month or day
 380 in either Figure 5b or Figure 5c respectively. At a glance, we notice that clusters 1 and 2

**Figure 5.** Visualizing the generated clusters.

381 are more likely to occur on the weekdays with cluster 3 taking over the majority share
 382 of the weekend which tends to explain the more consistent draw in power throughout
 383 the entirety of the day for samples that belong to cluster 3. Furthermore, samples in
 384 cluster 1 tend to gravitate towards the warmer, summer months peaking in terms of
 385 number of occurrences in the month of July while samples in clusters 2 and 3 exhibit
 386 a more uniform spread over the remainder of the colder months which could explain
 387 the lower average draw in power present in samples that belong to cluster 1 being a
 388 result of the owners of the home not being in as often or potentially not needing to make
 389 use of appliances to heat up their home (we note that this data was collected in Sceaux,
 390 France that experiences a warm season of ~ 3 months with otherwise, generally, cooler
 391 temperatures).

392
 393 *N.B.: It is worth noting that performing these same steps on households from within the REFIT
 394 data set exhibit similar results.*

395 5.3. Stage 3 - Further Data Preprocessing

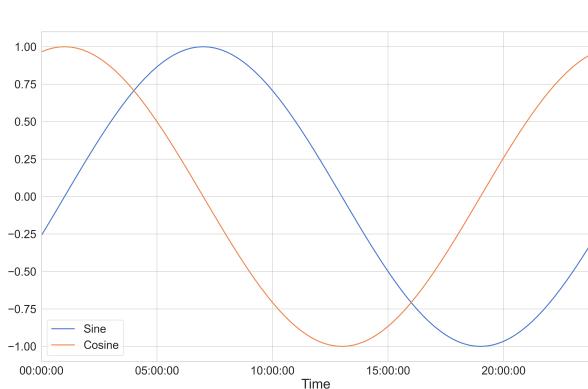
396 The first step undertaken, again on a per-cluster basis, is to append both temporal
 397 data as well as meteorological data to our data sets. Table 2 pertains to the temporal
 398 variables that will be taken into consideration as part of this feature engineering step
 399 while Table 1 pertains to the obtained, historical meteorological data that concern the

regions associated with our data sets. Incidentally, as outlined in Table 2, the temporal variables we have chosen to append do not hold much value given their current format. This is due mostly in part to their cyclical nature (think of how the 23rd hour of the day is rather close to hours 0 and 1). To handle this we can encode our temporal variables (for example, through the use of both the sine and cosine function) in an attempt to transpose our linear interpretation of time into a cyclical state that can be better interpreted by our deep learning model further down the line. The result of performing this so-called encoding can be seen in both Figures 6a and 6b.

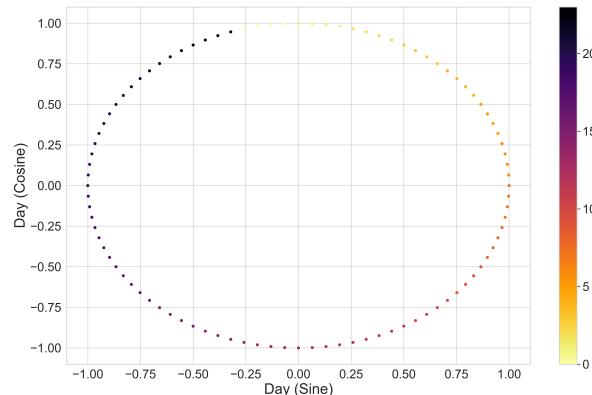
| Variable | Description |
|----------|--|
| Day | An integer value between 1 and 31. |
| Weekday | An integer value between 0 and 6 denoting the different days of the week. |
| Month | An integer value between 1 and 12. |
| Year | An integer value between 2007 and 2010. |
| Hour | An integer value between 0 and 23. |
| Minute | An integer value between 0 and 45 in increments of 15. |
| Season | An integer value between 0 and 3 where 0 denotes Spring, 1 denotes Summer, 2 denotes Fall and 3 denotes Winter. |
| Holiday | A categorical variable that takes on an integer value of 1 when the day concerned is a public holiday and 0 otherwise. |

Table 2: List of temporal variables that are taken into consideration during the feature engineering process as outlined in Section 5.3.

Following the feature engineering process, the feature selection process, heavily revolves around minimizing the overall number of features that do not serve as good predictors of our target variable. To assess which features we will be keeping and which we will be dropping we perform a variety of tests that determine which of our features present a significant level of independent (or combinatorial) correlation as well as causation when considering our target variable for each of the REFIT and UCID data sets. The primary tests conducted revolved around the concepts of Granger Causality (see Figure 7) and mutual information gain (See Figure 8) although other factors (such as a per-variable variance threshold) were also looked into.

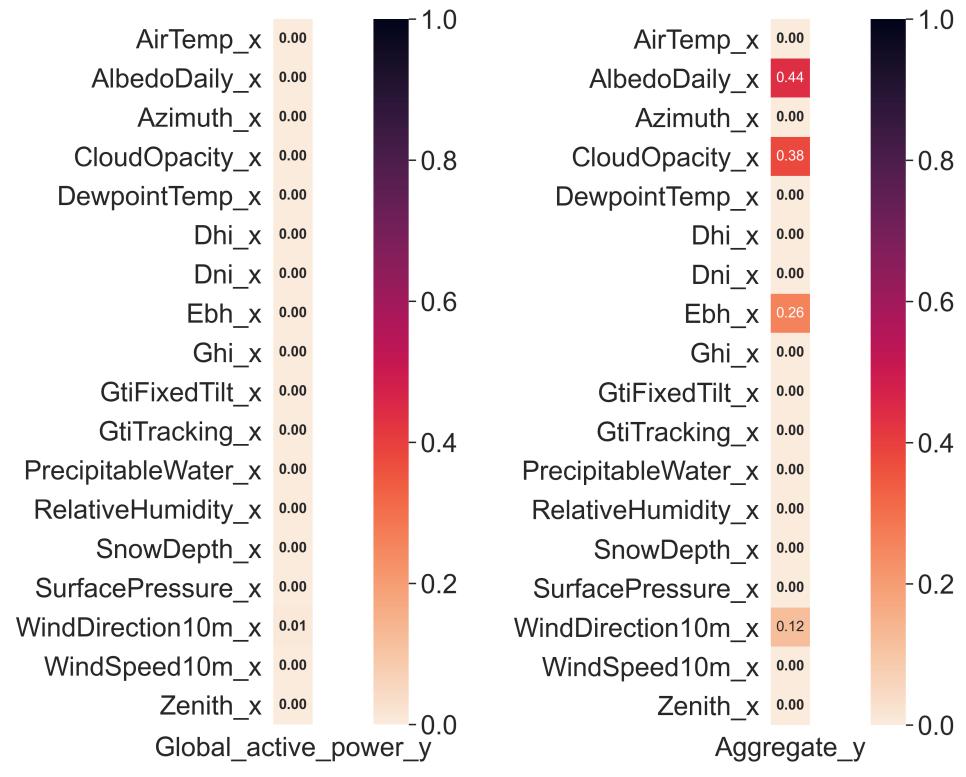


(a) Representing the time of the day as a combination of both sine and cosine waves.



(b) Visualizing our cyclical encoding of the time of day.

Figure 6. By utilizing a combination of the sine function and the cosine function, we eliminate the possibility that two different times would receive the same value had we used either function independently. The combination of both functions can be thought of as an artificial 2-axis coordinate system that represents the time of day.



(a) UCID data set.

(b) REFIT data set.

Figure 7. Trimmed Granger Causation matrix that display the Granger Causality of our independent features against our target variable for each of the UCID as well as the REFIT data sets.

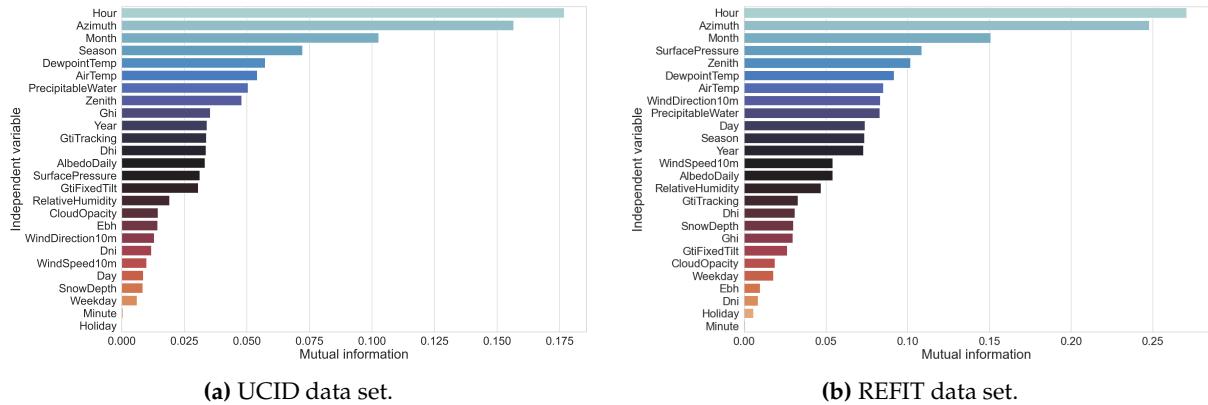


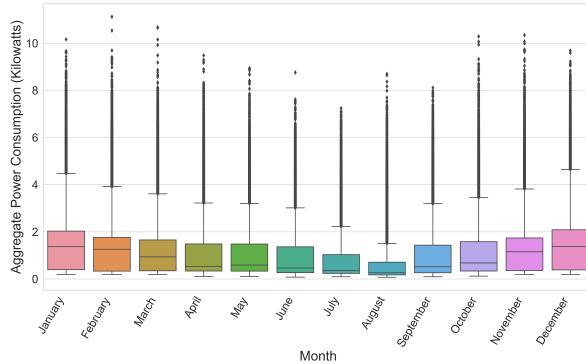
Figure 8. Mutual information gain with regards to our independent features and target variable for each of the UCID as well as the REFIT data sets.

When taking our target variable into consideration, the notion of outliers (and how to deal with them), is inevitable. Leaving them in is one possibility, as some level of noise is unavoidable in the data collection process and training our models on unrealistically curated data does not serve to produce an accurate representation of a real-life scenario in which a model of this caliber could be applied. Alternatively one method, explored during prior, related research [6], worked on the basis of defining an upper and lower bound based on the interquartile range (IQR). The IQR is calculated as the difference between the 75th (Q3) and 25th (Q1) percentiles of the data and comprises the box in a traditional box and whiskers plot. Using the IQR we can define outliers as any values

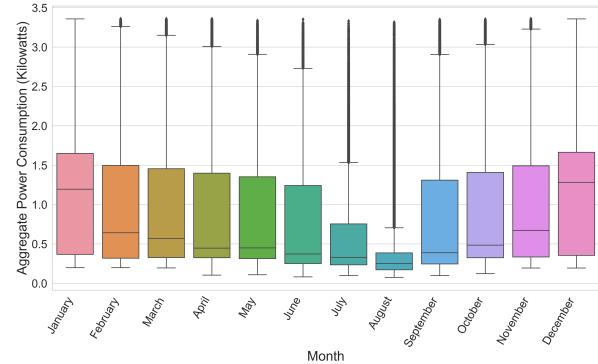
⁴²⁶ that are a pre-defined factor below the 25th percentile or above the 75th percentile as
⁴²⁷ follows:

$$Q1 - (1.5 * IQR) < x < Q3 + (1.5 * IQR) \quad (1)$$

⁴²⁸
⁴²⁹ Figures 9a and 9b represent the distribution of values for our target variable over the
⁴³⁰ different months of the year both before and after removing outliers respectively.



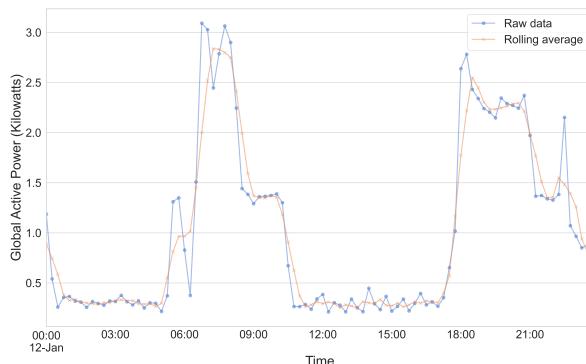
(a) Before removing outliers.



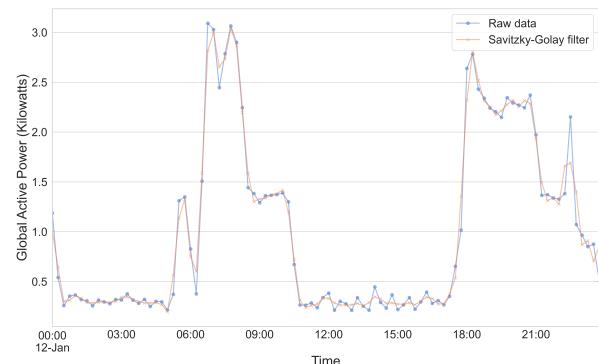
(b) After removing outliers.

Figure 9. Illustrating the distribution of values with respect to the global active power of the UCID data set both before and after removing outlier values as defined by Equation 1.

⁴³¹ *Smoothing*, or otherwise filtering, the data can also be done through the use of a variety
⁴³² of techniques and can help alleviate some of the issues inherent to the noise present
⁴³³ in our data as a byproduct of the data collection process. An example of performing
⁴³⁴ a preliminary smoothing step on energy consumption data can be seen in the work of
⁴³⁵ Hsiao [4] in which a moving (or rolling) average method was utilized.



(a) Application of the moving average method with a window size of 3.



(b) Application of the Savitzky-Golay filter method with a polynomial order of 3 and a window size of 5.

Figure 10. Illustrating the application of both the moving average method as well the Savitzky-Golay filter method in smoothing on a subset of our raw data.

⁴³⁶ With regards to our proposed forecasting pipeline, we will be utilizing Savitzky-Golay
⁴³⁷ filters [25] to smooth our raw, electrical energy consumption data as, when compared to
⁴³⁸ the moving average method, Savitzky-Golay filters tend to do a better job at preserving
⁴³⁹ the integrity of the raw data. Figures 10a and 10b serve to illustrate the application of
⁴⁴⁰ both the moving average method as well as the Savitzky-Golay filter method on a subset
⁴⁴¹ of our (raw) data set in order to better visualize the differences between both methods.

⁴⁴² Similarly, when considering the trend component of our data (obtained through per
⁴⁴³ forming an additive time series decomposition and seen in Figure 11), a preliminary
⁴⁴⁴ smoothing step can be undertaken through the use of Locally Weighted Scatterplot
⁴⁴⁵ Smoothing (LOESS) – this is illustrated in Figure 12.

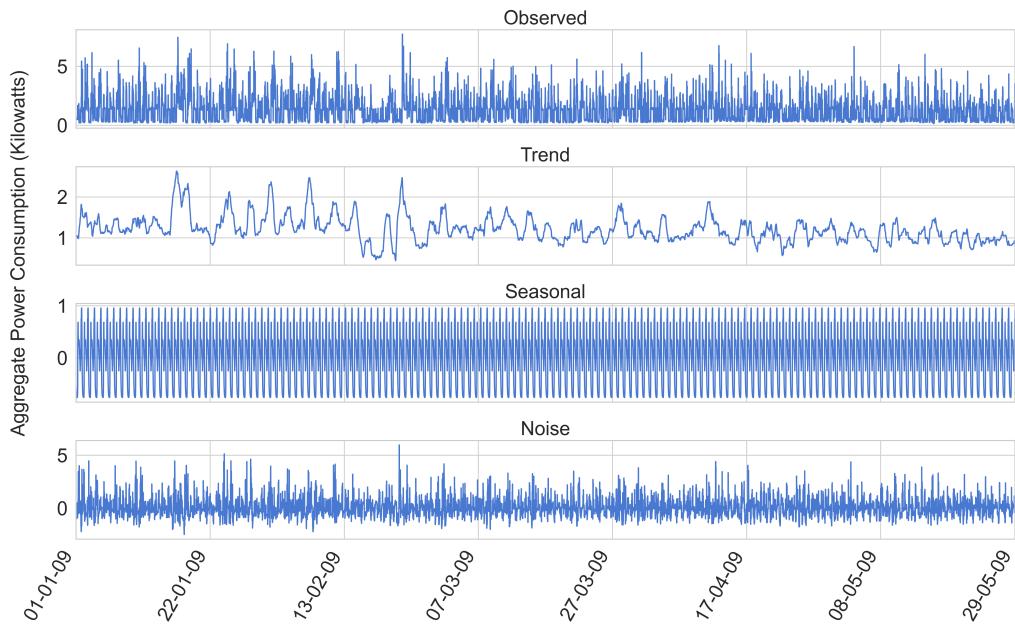


Figure 11. The out of performing time series decomposition on a subset of the UCID data set.

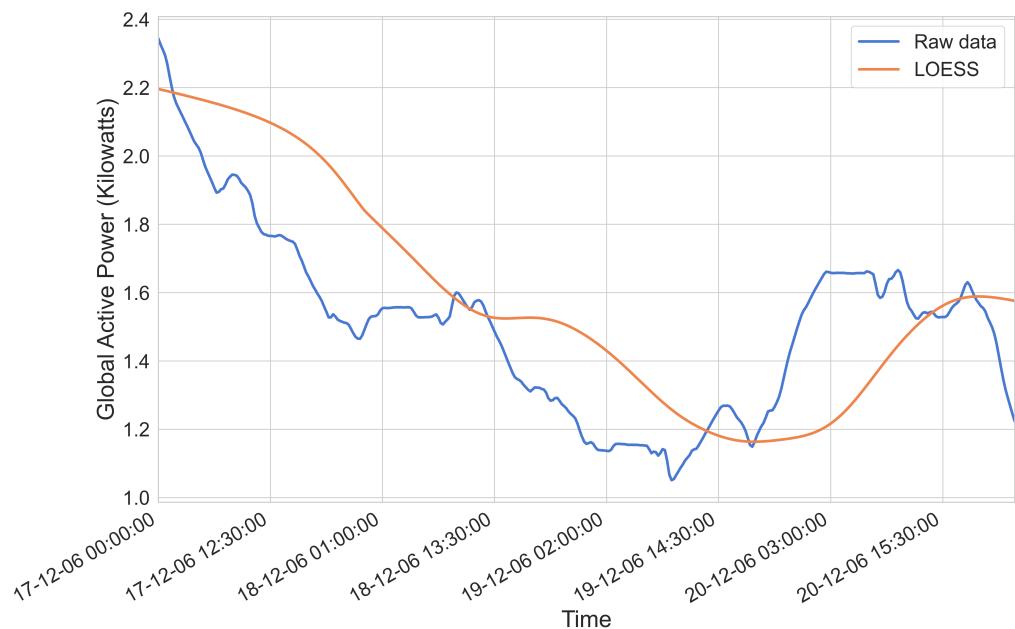


Figure 12. An illustration of the previously obtained trend component both with and without the application of LOESS.

⁴⁴⁶ The final step taken as part of Stage 3 of the forecasting pipeline is to split the data into 3
⁴⁴⁷ subsets that serve to act as training, validation and testing sets that will be fed to both
⁴⁴⁸ our classification tree as well as the CNN-LSTM network that we will be using for the
⁴⁴⁹ purpose of forecasting. A split employing an arbitrarily selected ratio of 60:20:20 is taken.
⁴⁵⁰ Given the nature of our study, we choose not to shuffle the data in either of the generated

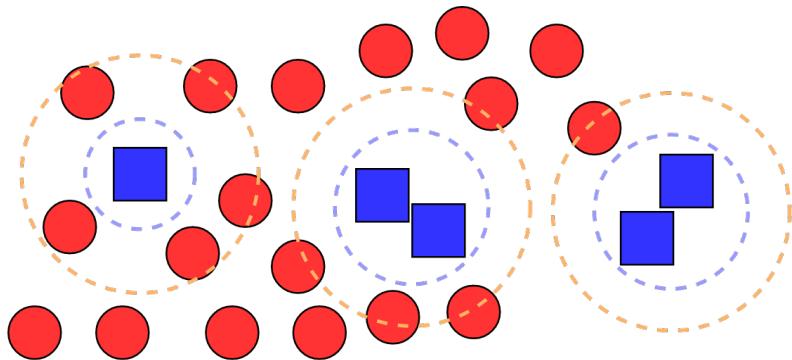
⁴⁵¹ sets as we are primarily interested in our model's capability of forecasting future trends
⁴⁵² in energy consumption given a measure of historically available data.

⁴⁵³ 5.4. Stage 4 - Training and Testing

⁴⁵⁴ In contrast to the earlier stages, stage 4 will be subdivided into Subsections 5.4.1
⁴⁵⁵ and 5.4.2, where Subsection 5.4.1 serves to present an overview of our classification
⁴⁵⁶ model while Subsection 5.4.2 serves to present an overview of our forecasting model.

⁴⁵⁷ 5.4.1. Stage 4.1 - Classification Tree

⁴⁵⁸ Before we can begin attempting to forecast trends in energy consumption we
⁴⁵⁹ will need to establish, or otherwise ascertain, our ability to correctly assign a new
⁴⁶⁰ point (or day) to the *correct* cluster. Given that the previously discussed clustering step
⁴⁶¹ separated the days in our data set on the basis of similarity in terms of patterns in energy
⁴⁶² consumption; this will not be an easy feat as the remaining, available context information
⁴⁶³ may not suffice in providing the relevant information to draw up a decision boundary
⁴⁶⁴ (of sorts) that serves to differentiate individual clusters.



No neighbors of the same class - **Noise**. Surrounded by another class - **Potentially unsafe**. Minimal neighbors from other class - **Safe**.

Figure 13. An illustration of the Synthetic Minority Oversampling Technique (SMOTE) algorithm in the case of 2 classes depicted by blue squares (minority class) and red circles (majority class). The blue square on the far left is isolated from other members of its class and is surrounded by members of the other class and thus is considered to be a noise point. The cluster in the center contains several blue squares surrounded by members from the other class and thus is indicative of potentially *unsafe* points that are unlikely to be random noise. Finally, the cluster in the far right contains predominantly isolated blue squares. The algorithm would then generate new, synthetic samples prioritizing the safer regions.

⁴⁶⁵ The first step in insuring a decently trained classifier is to deal with the glaring problem
⁴⁶⁶ of class imbalance that can be seen in Figure 5d. The results of our clustering step lead
⁴⁶⁷ us to an uneven distribution of days among the different class labels which could lead to
⁴⁶⁸ poor predictive performance as standard classification algorithms are inherently biased
⁴⁶⁹ to the majority class. A common means to alleviate this issue is to either undersample
⁴⁷⁰ the majority class(es) or oversample the minority class(es). In this paper we will be
⁴⁷¹ implementing the SMOTE algorithm, a form of informed oversampling, that works on
⁴⁷² the basis of generalizing the decision region for minority classes and thus provides us
⁴⁷³ with synthetic samples while preventing overfitting. For further explanations as to the
⁴⁷⁴ workings, advantages as well as shortcomings of this algorithm we refer the reader
⁴⁷⁵ to the initial paper by Chawla *et al.* [26] as well as Figure 13 that provides a layman's
⁴⁷⁶ explanation of the algorithm. The results of applying the SMOTE algorithm, and the
⁴⁷⁷ overall negation of the previously mentioned class imbalance, can be seen in Figure 14.

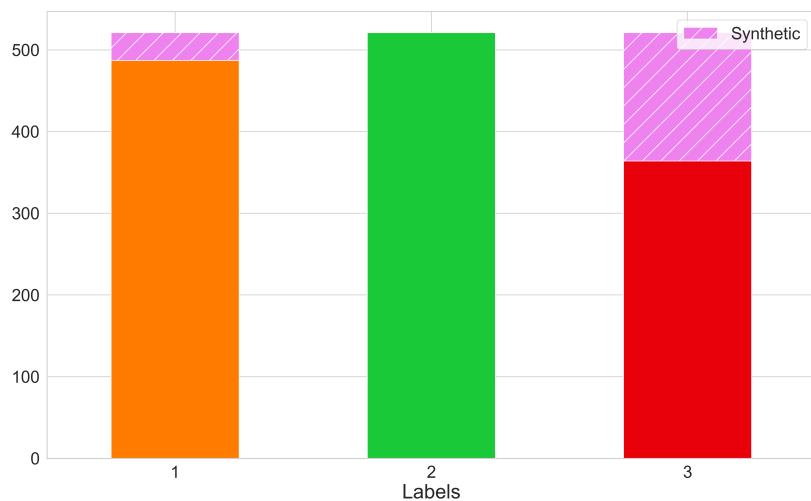


Figure 14. Number of samples per class label after applying the SMOTE algorithm.

478 After handling the class imbalance problem we can shift our attention to both the
 479 feature engineering as well as the feature selection process of this particular classification
 480 problem. In this scenario, the available context information we have is purely temporal
 481 (ordinal day of the week/year, month, season, etc.), and historical as well as forecasted
 482 meteorological data (air temperature, humidity, cloud opacity, etc.) and these will serve
 483 to act as the base-line number of features that our classifier will receive with which to
 484 assign a new sample into one of the previously generated clusters.

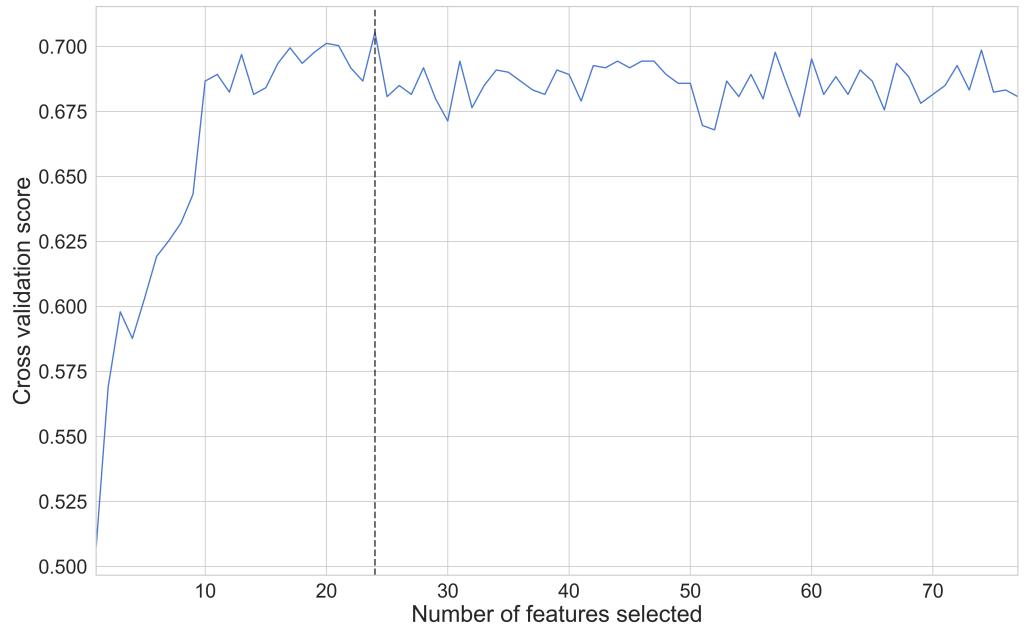


Figure 15. Assessing the number of important features through the use of the Recursive Feature Elimination and Cross-Validation (RFECV) algorithm. In this particular scenario, the optimal number of features was pruned down from a total of 77 to a mere 24.

485 Numerous methods exist to minimize the overall amount of features being passed to
 486 our classification model. In brief, we chose to make use of a Random Forest Classifier,
 487 the hyperparameters of which were tuned through a randomized search over a pre-
 488 determined distribution of values per hyperparameter. After assessing the optimal
 489 hyperparameters for our use-case, we passed the model as well as the complete set of
 490 features through a feature selection algorithm titled RFECV. RFECV works on the basis
 491 providing a cross-validated selection of the most important features when considering a

492 target label and pruning the less important features. Applying this algorithm reduces
 493 the overall number of features that our Random Forest Classifier utilizes from an initial
 494 77 down to a mere 24 (as shown in Figure 15) which is an overall reduction of ~ 68%.
 495
 496 After transforming the data set and pruning the less important features we, once
 497 again, train the model on the new, transformed data set utilizing 5-fold stratified cross-
 498 validation to assess whether our model is overfitting at any stage and so as to ensure an
 499 even distribution of class labels per validation set. We can conduct a quick inspection of
 500 the now fitted model by calculating the permutation feature importance on a per-feature
 501 basis to validate whether the final set of features are relevant when attempting to classify
 502 a new sample into the correct cluster. By definition, the permutation importance of a
 503 feature is the overall decrease in accuracy of our model when said feature's values are
 504 randomly shuffled and, by doing so, we break the relationship between the feature and
 505 the target label. By doing this we can assess how much our model depends on said
 506 feature, the results of which can be seen in Figure 16. It is important to note that, when
 507 calculating the permutation importance of strongly correlated features – the model will
 508 still have access to the shuffled feature through its correlated feature which will result in
 509 a lower importance value for *both* features when they might actually be important. To
 510 address this, it is possible to further prune the data set and remove subsets of features
 511 that present strong inter-correlation.

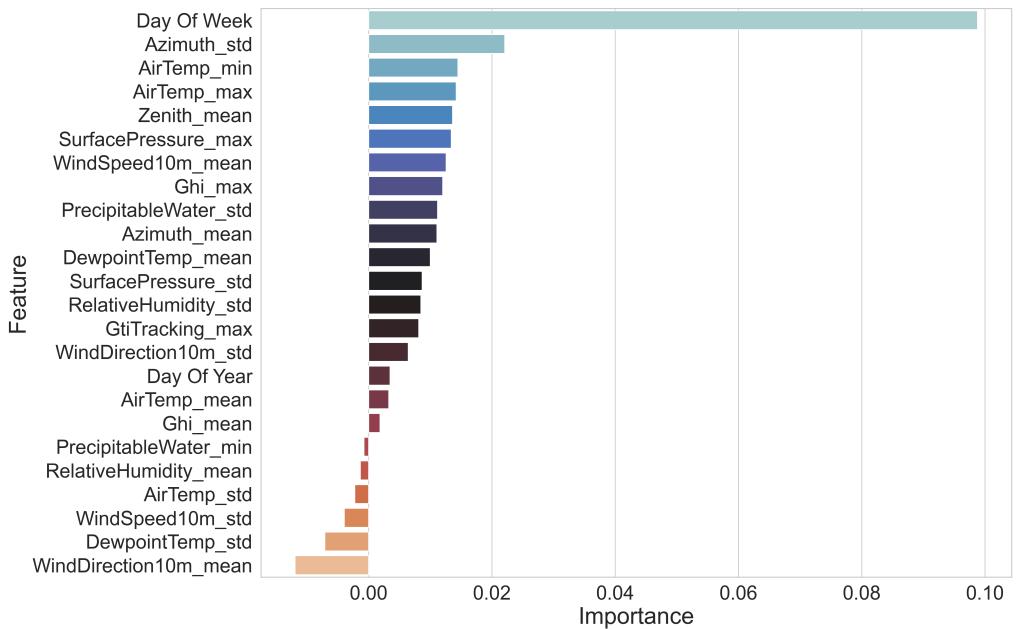


Figure 16. The permutation importance of each of the features chosen as part of our fitted Random Forest classifier.

512 The final model is then ready to accept new samples and assigns them a cluster based
 513 on the training procedure outlined through Section 5.4.1.

5.4.2. Stage 4.2 - CNN-LSTM Network

515 The focal point of our research lies in the implementation of a CNN-LSTM model
 516 in which the Convolutional Neural Network (CNN) component serves to learn the
 517 relative importance of each of the features (temporal as well as meteorological) that we
 518 pass to the network as input in what we can loosely call a *feature extraction* step. The
 519 extracted features are then passed to the LSTM portion of the network that learns the
 520 temporal relationship between past, or otherwise historical, values of said features with
 521 the present, or future, value(s) of the target variable where finally, an output prediction
 522 is made. The combination of both CNN and LSTM components allows the network to

523 learn spatio-temporal relationships between the features being passed as input and the
 524 target variable that we are attempting to forecast. In contrast to other architectures and
 525 forecasting models, this architecture is demonstrably more efficient [21] when tackling
 526 time series problems such as that of residential energy consumption forecasting. The
 527 sample network illustrated in Figure 17 can be expanded to forecast multiple time steps
 528 ahead with minor adjustments and is capable of understanding patterns at variable
 529 time resolutions. For the purposes of this example, we will be moving forward with
 530 the previously defined resolution of 15 minutes using a window of 24 historical values
 531 ($t - 24, t - 23, \dots, t$) to make a prediction one step into the future ($t + 1$) for both the
 532 previously established trend component as well as the raw, unadulterated data.

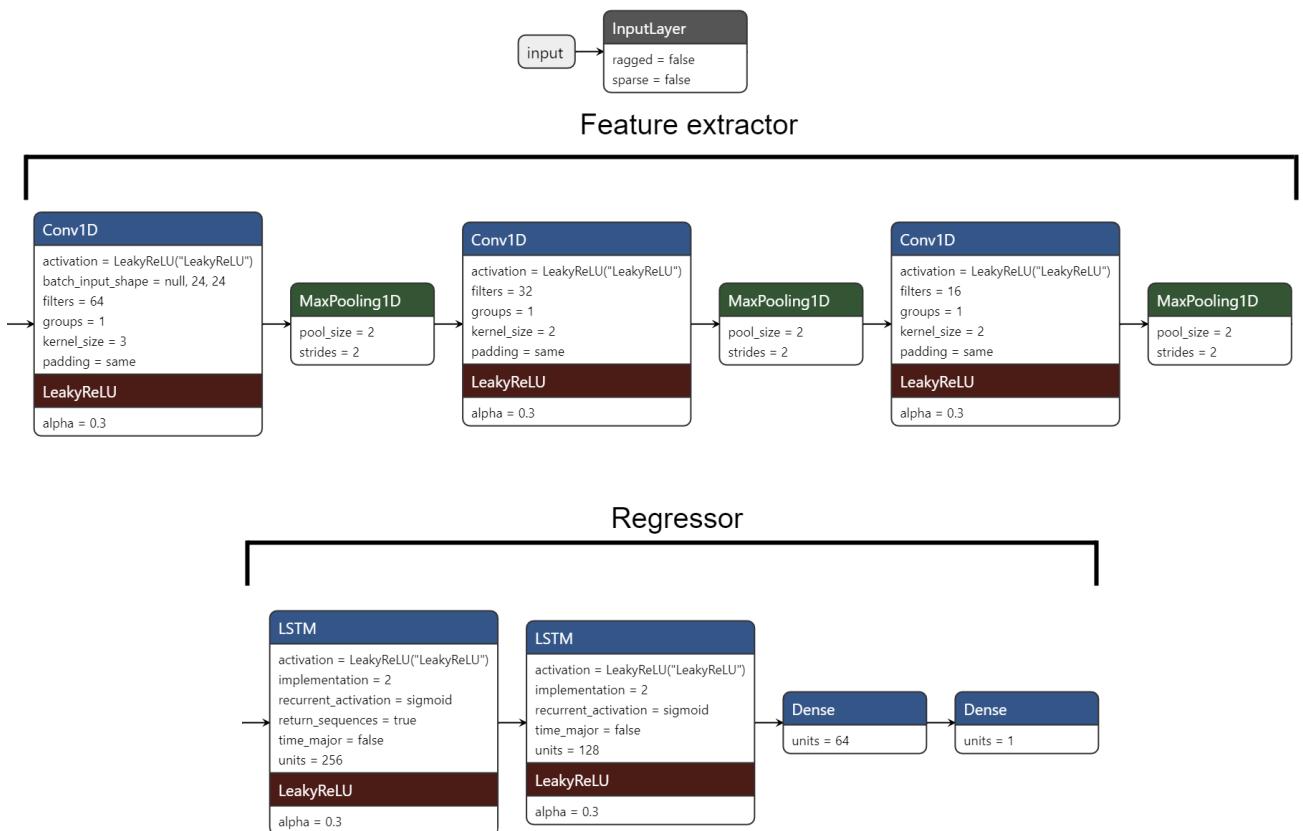


Figure 17. A simple, example CNN-LSTM network that makes one-step-ahead predictions.

533 To train our network, we will be utilizing Adam [27]: an adaptive learning rate optimi-
 534 zation algorithm that was designed specifically for training deep neural networks. In
 535 contrast to the ever-familiar Stochastic Gradient Descent, Adam leverages the power
 536 of adaptive learning rate methods and momentum to allocate individual learning rates
 537 for each parameter of the network being trained. For further explanations as to the
 538 workings of this algorithm we refer the reader to the initial paper by Kingma and Ba
 539 [27]. Additionally, when training our network(s), we will be making use of a variety
 540 of techniques to improve generalization and prevent overfitting to the training data
 541 set(s). The first of these techniques is the notion of *early stopping*. Early stopping is a
 542 form of regularization that monitors the validation loss (or generalization error) and
 543 aborts the training when the monitored values either begin to degrade or do not shift for
 544 an arbitrarily set number of epochs. The second technique we will be using works on
 545 the notion of employing a variable learning rate that, in theory, facilitates convergence
 546 of our weight update rule and prevents learning from stagnating thus allowing us to
 547 break through plateaus and avoid settling at local minima. For the purposes of our
 548 experiments and procuring the results showcased in Chapter 6, we will be implementing
 549 a network on a per-cluster basis for both the raw data as well as the trend component of

550 each of our data sets. The networks implemented will serve to provide one-step-ahead
 551 forecasts as well as one-shot 12-step-ahead (3 hour) forecasts as proof of concept.

552 6. Results and Discussion

553 Following the brief example in Section 5, we extended the implementation to house
 554 12 of the REFIT data set. The subsequent sections serve to demonstrate the efficacy of
 555 both the classification step as well as the forecasting step of our method.

556 6.1. Cluster Label Classification

557 The first results to be presented are those of the classification step of our method –
 558 refer to Table 3.

| Data Set | No. of Clusters | Accuracy |
|------------------|-----------------|----------|
| UCID | 3 | 76% |
| REFIT - House 12 | 3 | 66% |

Table 3: Result of training, optimizing and evaluating a random forest classifier on the cluster labels obtained for each of the UCID as well as the REFIT data sets.

559 Being able to correctly assign new samples into the correct cluster is imperative so as
 560 to insure the highest likelihood of achieving consistently reliable forecasting accuracy.
 561 Given that we have an equal number of 3 clusters per data set and that we are working
 562 with a (synthetic) uniform distribution of samples over the different clusters, the scores
 563 outlined in Table 3 are fairly good (a random predictor would achieve an accuracy of
 564 33.3%). The disparity in the results between the 2 data sets could predominantly be
 565 linked to the following 2 reasons:

- 566 1. The UCID data set contained a much larger number of samples (days).
 567 2. The distribution of the samples over the different days of the week as well as the
 568 months is much more uniform in the UCID data set.

569 Figures 18a and 18b allow us to clearly visualize both the correct as well as the incorrect
 570 predictions made by our model. Interestingly, given that both the clusters formed for
 571 each of the UCID as well as the REFIT data set were quite similar in terms of the overall
 572 patterns that were captured, the fitted model per data set seems to be making mistakes,
 573 or otherwise incorrect predictions, of a similar magnitude, with cluster 2 containing the
 574 largest number of incorrect predictions for each of the data sets and cluster 1 containing
 575 the largest number of correct predictions.

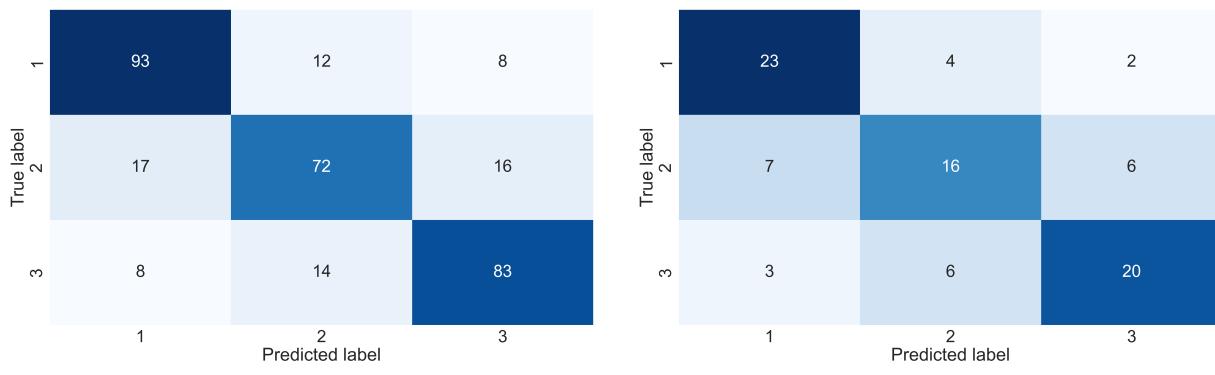


Figure 18. Confusion matrices for both the REFIT data set as well as the UCID data set.

576 6.2. Forecasting Accuracy

577 When compared to the current state-of-the-art in modern literature, particularly
578 with regards to data available on hand pertaining to the UCID data set, our method yields
579 superior forecasting accuracy at variable resolution. Table 4 presents a performance
580 comparison of common models discussed in the literature and our method at forecasting
581 one step into the future. We note that, at the time of writing, no published results
582 attempting to forecast energy consumption on the REFIT data set could be found and
583 thus, barre attempting to recreate the results ourselves, we have had to omit them from
584 Table 4 for the time being.

| Data Set | Method | MAE (kW) | RMSE (kW) | MAPE |
|-----------------|---------------|-----------------|------------------|-------------|
| UCID | LSTM [21] | 0.62 | 0.86 | 51.45% |
| | CNN-LSTM [21] | 0.34 | 0.61 | 34.84% |
| | Proposed | 0.14 | 0.19 | 21.62% |
| REFIT | LSTM | N/A | N/A | N/A |
| | CNN-LSTM | N/A | N/A | N/A |
| | Proposed | 0.11 | 0.17 | 25.77% |

Table 4: Performance comparison of different methods on each of UCID as well as the REFIT data set. Note that these results were obtained for one-step-ahead prediction at a resolution of 15 minutes over the raw data sets.

585 Another component that is frequently (attempted to be) forecasted in the literature is the
586 trend component obtained as part of a time-series decomposition step. We attempted
587 to tackle this problem ourselves and applied the method to both the smoothed trend
588 component of the UCID data set as well as house 12 of the REFIT data set, the results
589 of which can be seen in Table 5. We note that the results here are considerably good,
590 achieving a MAPE of $\sim 4\%$ for both data sets when forecasting a single time step into
591 the future.

| Data Set | MAE (kW) | RMSE (kW) | MAPE |
|-----------------|-----------------|------------------|-------------|
| UCID | 0.02 | 0.02 | 2.58% |
| REFIT | 0.02 | 0.02 | 4.32% |

Table 5: Performance metrics obtained when applying our method on the trend component of each of the UCID as well as the REFIT data sets to obtain a one-step-ahead prediction.

592 Finally, we attempted to extend our model by scaling up the number of predictions from
593 a singular step (15 minutes into the future in this scenario) to a total of 12 sequential steps
594 (leading to a grand total of 3 hours being forecasted given the previously mentioned
595 step size of 15 minutes), the results of which can be seen in Table 6.

| Data Set | Method | MAE (kW) | RMSE (kW) | MAPE |
|-----------------|---------------|-----------------|------------------|-------------|
| UCID | Raw | 0.37 | 0.59 | 38.23% |
| | Trend | 0.02 | 0.02 | 3.15% |
| REFIT | Raw | 0.17 | 0.31 | 39.75% |
| | Trend | 0.02 | 0.02 | 4.75% |

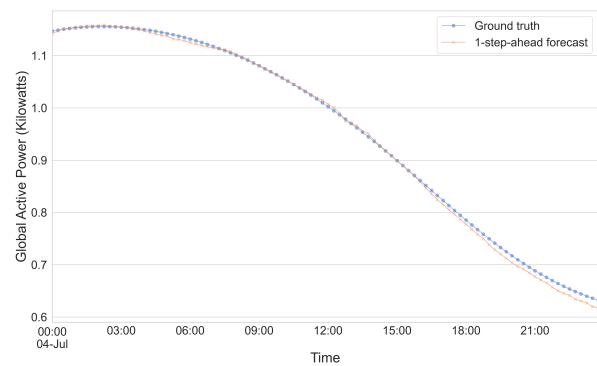
Table 6: Performance metrics obtained when applying our method on both the raw data as well as trend component of each of the UCID as well as the REFIT data sets to obtain twelve-step-ahead predictions.

596 Oddly enough, for both the UCID data set as well as house 12 of the REFIT data set,
 597 we achieved marginal improvements with regards to MAPE scores when attempting to
 598 build twelve-step-ahead forecasts on their respective trend components. On the other
 599 hand, MAPE scores for the raw data for each of our data sets fell somewhat substantially,
 600 with an overall loss of about $\sim 10\%$, when moving from one-step-ahead forecasts to
 601 twelve-step-ahead forecasts which is more in line with what one could expect in this
 602 scenario.

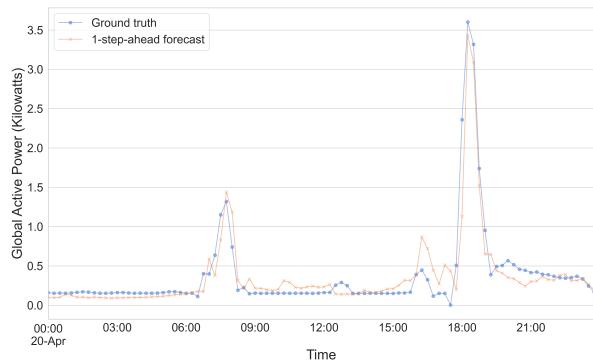
603 To further showcase, or otherwise visualize, the capabilities of our model we present
 604 Figures 19a, 19b, 20a and 20b that serve to illustrate one-step-ahead forecasts generated
 605 for a subset of each of the UCID data set as well as house 12 of the REFIT data set.
 606 These figures illustrate predictions made by each of our individual models on both the
 607 raw data as well as the trend component for each of the REFIT data set as well as the
 608 UCID data set over a period of 1 day out of our test set. A cursory glance at Figures 19b
 609 and 20b shows us that our model seems to excel at making one step predictions on the
 610 trend component of the UCID data set while the predictions being made for the trend
 611 component of the REFIT data set seem to be slightly less accurate. When considering
 612 the raw data for each of the data sets (Figures 19a and 20a) the differences are less
 613 pronounced and it seems that the model is capable of making accurate predictions one
 614 step into the future. This might vary in between days though and, given that the days
 615 chosen for these illustrations were completely random, it might be the case that the
 616 model performs better (or worse).



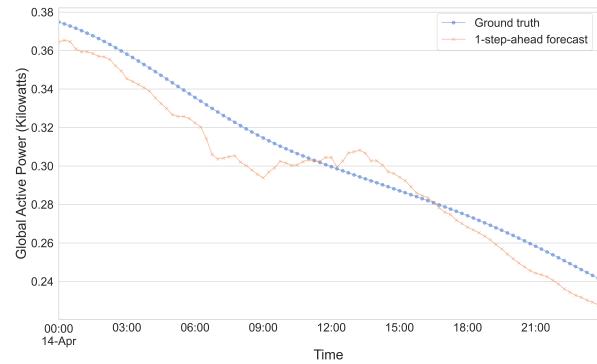
(a) UCID - Raw data.



(b) UCID - Trend.

Figure 19. Showcasing the capabilities of our method in making one-step-ahead predictions on the UCID data set.

(a) REFIT - Raw data.



(b) REFIT - Trend.

Figure 20. Showcasing the capabilities of our method in making one-step-ahead predictions on the REFIT data set.

618 N.B. we note that the results obtained as part of Section 6.2 are the averaged results obtained
619 from training, optimizing and assessing multiple models, one for each of the respective clusters
620 obtained as part of stage 2 of our method. Furthermore, all results were obtained at a resampled
621 resolution of 15 minutes per time-step; however, similar results have been observed for variable
622 time resolutions (1 minute, 1 hour etc.)

623 7. Conclusion and Future Work

624 In this study, we have shown that the application of a clustering step that utilizes
625 dimensionality reduction techniques such as t-SNE and hierarchical, density-based
626 clustering in the form of HDBSCAN leads to significant improvements in forecasting
627 accuracy when taking individual households into consideration. While this technique is
628 certainly more complex, in particular with regards to the number of steps and moving
629 parts associated with the entire pipeline, we maintain that the benefits in terms of
630 improved forecasting accuracy outweigh the overall increase with regards to the time
631 and effort it would take to train and set up such a model. The practicality of the model
632 lies in the availability of the data that it requires to function – primarily with respect to
633 historical energy consumption data for the individual households in question (which
634 is becoming easier and easier to obtain thanks to the prevalence of smart meters) and
635 meteorological data that can easily be obtained from numerous sources. Furthermore, it
636 is highly likely that, given enough historical data, the need to further train the model(s)
637 after the initial setup is rather low, further compounding the efficacy of our method.

638
639 Another one of the benefits of our method that we previously discussed is that no
640 prior knowledge of the number of clusters is required. As there is no guarantee that
641 any 2 individual households contain a similar number of *repeating* patterns we avoid
642 running into the problem of overly generalizing a single working solution that may or
643 may not work given said change in energy consumption patterns and instead present
644 a solution that could potentially extend to a much larger scale. A potential issue with
645 this implementation however, is that an individual household *may* contain a large
646 number of repeating consumption patterns which could possibly lead to an overall
647 decline in what can already be considered sub-par performance from our classifier.
648 That said, there is definitely room for improvement that could accommodate these
649 potential risks, specifically with regards to the feature engineering step – for example,
650 improvements in classifier accuracy could be seen through the utilization of a more
651 efficient classifier. Alternatively, the current lack of contextual information that serves
652 to explain the emergence of the clusters as part of the clustering step could likely be
653 the reason for obtaining sub-par accuracy scores as, in its current iteration, the premise
654 of our clustering step was to group together days that exhibited the highest similarity
655 purely in terms of their energy consumption patterns and, given that this information is
656 not readily available to us when considering a new day, we are left reaching for straws
657 when attempting to explain when any individual household is likely to observe energy
658 consumption patterns that fall within any of the obtained clusters. Evidently, temporal
659 and meteorological information is not enough to explain the emergence of said clusters
660 and other information (perhaps patterns in terms of cluster labels leading up to the
661 new sample) could serve to improve classifier accuracy. This is definitely an area of this
662 study that could be looked into as part of future research. Additionally, regardless of
663 the fact that the performance of our forecasting model is the highlight of this paper, it
664 is interesting to note that a byproduct of our method is the potential to extract insights
665 into variables that have an effect on the daily energy consumption patterns of unique
666 households. A cursory glance at applying our method to a portion of the data at hand,
667 as an example of the insights that we can obtain, shows us that some households have
668 frequently occurring patterns that tend to deviate among the different days of the week
669 while other households have an even bigger separation across months of the year or
670 even among meteorological factors such as the temperature or chance of rain.

- 671
672 To conclude, we note that, as a result of pre-clustering our data, and then training
673 separate models on a per-cluster basis we achieved an improvement in overall forecasting
674 accuracy with superior MAPE scores (21.62% in contrast to 34.84%) when considering
675 the current state-of-the-art (LSTM networks, clustering based on K-means, etc.).
- 676 **Author Contributions:** K. Al-Saudi is the main author, who implemented the software, performed
677 validation, analysis, writing, having completed this project as his Master Thesis. Both V. Degeler
678 and M. Medema are involved as supervisors of K. Al-Saudi, and in the methodology analysis and
679 validation, with V. Degeler being additionally involved in project conceptualization. All authors
680 have read and agreed to the published version of the manuscript.
- 681 **Data Availability Statement:** The data that is used in this paper is publicly available [9,10]. In
682 the interest of open science, the code is also made publicly available [here](#). Any use must include a
683 citation to this paper.
- 684 **Conflicts of Interest:** The authors declare no conflict of interest.

References

1. Wei, Y.; Zhang, X.; Shi, Y.; Xia, L.; Pan, S.; Wu, J.; Han, M.; Zhao, X. A Review of Data-Driven Approaches for Prediction and Classification of Building Energy Consumption. *Renewable and Sustainable Energy Reviews* **2018**, *82*, 1027–1047. doi:<https://doi.org/10.1016/j.rser.2017.09.108>.
2. Chen, C.; Das, B.; Cook, D. Energy Prediction in Smart Environments. Workshops Proceedings of the 6th International Conference on Intelligent Environments, Kuala Lumpur, Malaysia, 2010, pp. 148–157. doi:10.3233/978-1-60750-638-6-148.
3. Yildiz, B.; Bilbao, J.I.; Dore, J.; Sproul, A. Household Electricity Load Forecasting Using Historical Smart Meter Data with Clustering and Classification Techniques. *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)* **2018**. doi:10.1109/ISGT-ASIA.2018.8467837.
4. Hsiao, Y.H. Household Electricity Demand Forecast Based on Context Information and User Daily Schedule Analysis From Meter Data. *IEEE Transactions on Industrial Informatics* **2015**, *11*, 33–43. doi:10.1109/TII.2014.2363584.
5. Raza, M.Q.; Khosravi, A. A Review on Artificial Intelligence Based Load Demand Forecasting Techniques for Smart Grid and Buildings. *Renewable and Sustainable Energy Reviews* **2015**, *50*, 1352–1372. doi:<https://doi.org/10.1016/j.rser.2015.04.065>.
6. Al-Saudi, K. The Effectiveness of Different Forecasting Models on Multiple Disparate Datasets.
7. Foucquier, A.; Robert, S.; Suard, F.; Stéphan, L.; Jay, A. State of the Art in Building Modelling and Energy Performances Prediction: A Review. *Renewable and Sustainable Energy Reviews* **2013**, *23*, 272–288. doi:<https://doi.org/10.1016/j.rser.2013.03.004>.
8. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid* **2019**, *10*, 841–851. doi:10.1109/TSG.2017.2753802.
9. Murray, D.; Stankovic, L.; Stankovic, V. An Electrical Load Measurements Dataset Of United Kingdom Households From A Two-year Longitudinal Study. *Scientific Data* **2017**, *4*. doi:10.1038/sdata.2016.122.
10. UCI Machine Learning Repository: Individual Household Electric Power Consumption Data Set. Available online at: <https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption>.
11. Global Solar Irradiance Data and PV System Power Output Data, 2019. Available online at: <https://solcast.com/>.
12. Fallah, S.N.; Deo, R.C.; Shojafar, M.; Conti, M.; Shamshirband, S. Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions. *Energies* **2018**, *11*.
13. Backer, E.; Jain, A.K. A Clustering Performance Measure Based on Fuzzy Set Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1981**, *PAMI-3*, 66–75. doi:10.1109/TPAMI.1981.4767051.
14. Stephen, B.; Tang, X.; Harvey, P.R.; Galloway, S.; Jennett, K.I. Incorporating Practice Theory in Sub-Profile Models for Short Term Aggregated Residential Load Forecasting. *IEEE Transactions on Smart Grid* **2017**, *8*, 1591–1598. doi:10.1109/TSG.2015.2493205.
15. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD. AAAI Press, 1996, KDD'96.
16. Yildiz, B.; Bilbao, J.; Dore, J.; Sproul, A. Recent Advances in the Analysis of Residential Electricity Consumption and Applications of Smart Meter Data. *Applied Energy* **2017**, *208*, 402–427. doi:<https://doi.org/10.1016/j.apenergy.2017.10.014>.
17. Kohonen, T. The Self-Organizing Map. *Proceedings of the IEEE* **1990**, *78*, 1464–1480.
18. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*; Springer, 2017.
19. Fumo, N.; Rafe Biswas, M. Regression Analysis for Prediction of Residential Energy Consumption. *Renewable and Sustainable Energy Reviews* **2015**, *47*, 332–343. doi:<https://doi.org/10.1016/j.rser.2015.03.035>.
20. Shi, H.; Xu, M.; Li, R. Deep Learning for Household Load Forecasting – A Novel Pooling Deep RNN. *IEEE Transactions on Smart Grids* **2018**, *9*, 5271–5280. doi:10.1109/TSG.2017.2686012.
21. Kim, T.Y.; Cho, S.B. Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks. *Energy* **2019**, *182*, 72–81. doi:<https://doi.org/10.1016/j.energy.2019.05.230>.

22. O'Malley, T.; Bursztein, E.; Long, J.; Chollet, F.; Jin, H.; Invernizzi, L.; Others. Keras Tuner. <https://github.com/keras-team/keras-tuner>, 2019.
23. Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **2008**, *9*, 2579–2605.
24. Leland McInnes, John Healy and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, 2020, [[arXiv:stat.ML/1802.03426](https://arxiv.org/abs/stat/ML/1802.03426)].
25. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* **1964**, *36*, 1627–1639. doi:10.1021/ac60214a047.
26. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357. doi:10.1613/jair.953.
27. Kingma, D.; Ba, J. Adam: A Method For Stochastic Optimization. *ICLR* **2015**.