INDIVIDUAL SMART METER'S ENERGY CONSUMPTION FORECASTING FOR STRATEGIC DECISION MAKING

Student: María Belén Alberti

Advisor: Ramiro H. Gálvez

Master in Management Analytics

School of Business

Universidad Torcuato Di Tella

Argentina

2020

Abstract

This paper analyzes the benefits of high frequency data obtained from smart meters readings, specifically from individual smart meter household's energy consumption. The purpose is to learn the consumer's behavior as leverage to improve the business strategy, the consumer's experience and work towards a more efficient market. To tackle this, we performed exploratory data analysis techniques where we not only learned more about the customers, but we cleaned the data to perform load forecasting. For this last point we employed both statistical and machine learning techniques in order to help reach a consensus on the best option for this type of data. Results showed that customer characterization can be key for analyzing consumption behavior as well as a great strategy to improve forecasting. Also, the industry's standard for forecasting performed very poorly compared to other techniques. From an industry standpoint this study shows how the use of data form smart meters can greatly benefit both the industry and the consumer. Energy consumption and, therefore, generation is a key player for the world economy whilst also being a scarce resource that we should learn to better manage; big data together with the right analytics tools can be a great place to start.

PREDICCIÓN DEL CONSUMO DE ENERGÍA DE MEDIDORES INTELIGENTES INDIVIDUALES PARA LA TOMA DE DECISIONES ESTRATÉGICAS

Resumen

Este trabajo busca analizar los beneficios de los datos masivos que podemos obtener del uso de medidores inteligentes, en nuestro caso especifico, de medidores de electricidad en viviendas personales. El propósito es aprender el comportamiento del consumidor como partida para mejorar la estrategia de las empresas de energía, la experiencia del consumidor y finalmente, trabajar hacia un mercado de energía más eficiente. Para lograr esto, empleamos técnicas de exploración de datos donde no solo aprendimos más sobre los consumidores, sino que también limpiamos los mismos para predecir el consumo. Para esto último, empleamos técnicas tanto estadísticas como de aprendizaje automático con el fin de ayudar a alcanzar un consenso sobre el mejor modelo para este tipo de datos. Los resultados muestran que la caracterización de los consumidores puede ser clave para analizar el comportamiento a la hora de consumir, como así también para mejorar la estrategia de predicción. Asimismo, la estrategia estándar de la industria no estuvo a la altura de lo esperado en comparación con otras técnicas. Desde el punto de vista de la industria, este estudio muestra como el uso de los datos de medidores inteligentes puede beneficiar tanto a la industria misma como el consumidor. El consumo de energía, y por lo tanto la producción, es un jugador clave para la economía mundial y a la vez un recurso escaso que deberíamos aprender como administrar mejor; el manejo adecuado de los datos masivos puede ser un gran primer paso.

Table of Contents

1 Introduction

As defined in (Koponen & Rochas, 2008), meters have been called smart since the introduction of static meters that included one or more microprocessors. Also, a meter is called smart to imply that it includes significant data processing and storage for various purposes such as monitoring that the meter is installed correctly and working properly, or for data communication with the meter using secure and open standard protocols, and providing real time consumption data to various actors (distributor, retailer, end user). For the past decade smart meters have come to impact the energy industry in a notable way. For example, (EU, 2019) states that "*The regular provision of accurate billing information based on actual electricity consumption, facilitated by smart metering, is important for helping customers to control their electricity consumption and costs*", that "*Member States should encourage the modernisation of distribution networks, such as through the introduction of smart grids, which should be built in a way that encourages decentralised generation and energy efficiency*", and, finally, that "*Smart metering systems also enable distribution system operators to have better visibility of their networks, and as a consequence, to reduce their operation and maintenance costs and to pass those savings on to the consumers in the form of lower distribution tariffs*".

Smart meters are now part of the advanced metering infrastructure (AMI) and the idea behind them is that billing is no longer the main function for energy consumption recordings. This is because smart meters record energy on a high frequency, allowing for an immense amount of fine-grained electricity consumption data to be collected and things like consumer behaviour or lifestyles can be explored. Perhaps the most obvious employment of this technology is to enhance the efficiency and sustainability of the power grid and therefore improve the company's strategies on decision making. From the point of view of consumers, this technology allows for a more conscious consumption. Consumers can manage their consumptions aiming to become smart consumers like idealized in the European Green Deal (Ursula et al., 2020). Moreover, they are perhaps one of the most adopted Internet of Things (IoT) technology on households (Li et al., 2011).

There is a general commitment from governments to transfer outdated electricity meters to smart meters as a goal for updating the energy systems and work towards smart cities. As an illustrative example, a study from December 2019 (Tounquet & Alaton, 2019) on the deployment of smart meters in the EU found that:

- Close to 225 million smart meters for electricity and 51 million for gas will be rolled out in the EU by 2024. This represents a potential investment of €47 billion.
- By 2024, it is expected that almost 77% of European consumers will have a smart meter for electricity. About 44% will have one for gas.
- The cost of installing a smart meter in the EU is on average between €180 and €200.
- On average, smart meters provide savings of €230 for gas and €270 for electricity per metering point (distributed amongst consumers, suppliers, distribution system operators, etc.) as well as an average energy saving of at least 2% and as high as 10% based on data coming from pilot projects.

Furthermore, the implementation of such technologies opens the door to data analytics and, therefore, the possibility to make decisions in an analytical way like presented on (Wang et al., 2019). Given the energy consumption data for a sample of a population, all stages of analytics can be explored:

1. For descriptive analytics we have Load Analysis. Here, applications such as bad data detection, non-technical loss detection and load profiling can be done like seen on (Luo et al., 2018).

2. For predictive analysis, we have Load Forecasting. Here we can perform forecasting with or without individual meters as well as probabilistic forecasting. As performed in (Edwards et al., 2012).
3. For prescriptive analysis, we have Load Management. The applications for this last stage can be customer characterization, demand response program marketing and demand response implementation as shown in (Zhong & Tam, 2015).

One of the main challenges when it comes to energy is that it cannot be stored in large amounts. Therefore, it is of utmost importance that both supply and demand must be matched by system operators, this is why, in this industry, forecasting is essential (Gajowniczek & Zabkowski, 2014). Long term forecasting is needed for resource management and investment development. Mid-term forecasting can be used for planning power production and tariffs. Short term forecasting is mostly used for scheduling and analysing the distribution network as well as implementing dynamic pricing.

Time series forecasting has become popular with the rise of big data. To tackle such problems there are both traditional and complex models. For traditional modelling, Auto Regressive Integrated Moving Average (ARIMA) or Seasonal Auto Regressive Integrated Moving Average (SARIMA) are the most popular ones. In Auto Regressive models the forecast will correspond to a linear combination of past values, whereas in Moving Average models, the forecast will correspond to a linear combination of past errors. SARIMA simply adds the possibility for linear combination for past seasonal errors and values. For more complex models, machine learning has popularized methods such as Artificial Neural Networks and Support Vector Machines for time series analysis. They provide the possibility to easily include data beyond the variable we want to forecast in a very natural way, perhaps what makes them so interesting for the community.

The aim of this work is to evaluate alternative strategies for short term load forecasting. Concretely, in our analysis, we will test both traditional and modern machine learning methods. For traditional models we will use ARIMA since, as we mentioned above, is the industry standard for time series forecasting. For more complex methods, we will test Long Short-Term Memory Artificial Neural Networks (LSTM). We believe that LSTM models have the potential to perform very well while still being approachable from a business standpoint. In order to evaluate the aforementioned models, we will propose two benchmark models. The first one is the widely adopted averaging model and the second one is an average model fitted to our context where we take into account the timestamp information.

The rest of the draft is structured as follows: In Section 2 (Materials and Methods) we will analyse the raw data in the hopes of pinpointing any pattern or best alternatives for the future stages of analysis. Ultimately, predicting demand requires an understanding of consumer behaviour and this is the objective for this first stage. We will also present the models and methods that will be used for the predictive analysis. In Section 3 (Data Exploratory Analysis) we will further deepen our analysis of our data by joining the databases as well as clean the data. In Section 4 (Predictive analysis) we will tune the parameters and hyperparameters of our models and present the results. In Section 5 we will propose strategic decisions that can be taken regarding the energy business. The purpose is to minimize costs for the companies when providing energy and understand the customers behaviour. We will also draw final conclusions by presenting a summary of our work, limitations and future work. Finally, in Section 7 we have all the annexed plots and results from Section 4 and in Section 6 we have the corresponding bibliography.

2 Materials and Methods

2.1 Materials

In this Section we will present the data that will be used for this work. In Section 2.1.1 we will present details on the origin of the data for the energy consumption as well as the customer's characteristics. In Section 2.1.2 we will provide details on the origin of our proposed weather covariates for our analysis in Section 5. In Section 2.2.3 we will present the data used in a more detailed way.

2.1.1 Energy Consumption Data

In this work, we will focus on analysing smart meter data provided by UK Power Networks.[1] They have made some of their operational data open and available in the London Datastore.[2] The data contains information of energy consumption readings for a sample of 5566 London Households that took part in the UK Power Network led by Low Carbon London project between November 2011 and February 2014. The customers in the trial were recruited as a balanced sample representative of the Greater London population. In Section 2.2.3 we will deepen our analysis on the presented data.

Data on energy consumption is distributed across five datasets *households, half hourly consumption, daily consumption, weather daily and weather half hourly.* Below we provide details on each of them.

[1] ("Index @ www.ukpowernetworks.co.uk,").
[2] ("Smartmeter-Energy-Use-Data-in-London-Households @ data.london.gov.uk,")

- *Households:* This dataset contains all the information on the households: household ID, the type of tariff they were subject to,[3] the socio-economic status they belong and the block file where their consumption information can be found.
- *Half hourly consumption:* This dataset contains the block files with the half-hourly smart meter's measurement: their household ID, timestamp and the recorded energy in kWh/hh (kilowatt-hour per half hour).
- *Daily consumption:* This dataset has daily information on the energy consumption: household ID, date, median of the energy consumed that day as well as the mean, max, standard deviation, sum and minimum. It also has the amount of times the energy was recorded (if it was done every half hour, this row should be 48).

2.1.2 Weather Data

Electricity consumption depends on various factors. However, one seems to stand out above the rest: weather. For this reason, we have decided to add weather information to further deepen the analysis. Data comes from the Dark Sky API.[4]

The use of weather variables has two purposes, the first derives from the use of energy sources such as wind or solar, and the second comes from consumer behaviour itself. The increasing use of solar and wind power as a source of energy poses a challenge directly linked with weather since the output of such sources is dependent on weather. If there is no wind or no sun, there is no energy generated or stored, and these fluctuations in weather behaviour (increased by climate change factors) affect directly the energy supplied and therefore

3 Approximately 1100 customers were subjected to Dynamic Time of Use (dToU) energy prices throughout the 2013 calendar year period. The tariff prices were given a day ahead via the Smart Meter IHD (In Home Display) or text message to mobile phone. Customers were issued High, Low or normal price signals and the times of day these applied.

4 https://darksky.net/dev

consumed. Weather also affects the consumption/generation of energy since very cold or very hot weather shifts upwards the demand for energy. It is therefore very clear that weather has an impact con consumption and it would be interesting to enrich our models with such data. In Section 2.2.3 we will deepen our analysis on choosing weather data as our covariates.

Data on weather is distributed across two datasets, *weather daily* and *weather half hourly*. Below we provide details on each of them.

- *Weather daily:* This dataset contains daily weather variables such as maximum/minimum temperature, timestamps for maximum/minimum temperature, wind speed, pressure, humidity and so on.
- *Weather half hourly:* Same as *Weather Daily* but with a timestamp interval of 30 minutes.

2.1.3 Basic Data Exploration

In this Section we explore the data presented in Section 2.1.1 and 2.1.2 in detail. A more in detail analysis, where we will relate patterns observed across datasets, like merging datasets for a more complex and rounded analysis, will be presented in Section 3.

2.1.3.1 Household information

Table 1 shows how the dataset looks like if we print its first rows. As mentioned in the previous Section, it has the household ID: 'LCLid', the type of tariff that household was subject to: 'stdorToU', the socio-economic category, Acorn, sub-group it belongs to: 'Acorn', the Acorn group: 'Acorn_grouped', and finally the block file where we can find the energy consumption for that specific household. There are 5566 rows and, therefore, households.

Table 1: First rows of the Household information dataset

LCLid	stdorToU	Acorn	Acorn_grouped	file
MAC005492	ToU	ACORN-	ACORN-	block_0
MAC001074	ToU	ACORN-	ACORN-	block_0
MAC000002	Std	ACORN-A	Affluent	block_0
MAC003613	Std	ACORN-A	Affluent	block_0
MAC003597	Std	ACORN-A	Affluent	block_0

As for the Acorn information, this classification was developed by CACI Limited in London.("Acorn-Consumer-Classification-Caci @ www.gov.uk") It is a segmentation tool which categorises the UK population into demographic types. There are 6 categories, 18 groups and 62 types.[5] In our data set we have 5 present categories, 'Acorn_grouped', and 19 groups (there is an extra group present), 'Acorn'. For example, the household 'MAC000002' belongs to the group 'ACORN-A' and the category 'Affluent'. For CACI classification this falls into 'Exclusive enclaves, Metropolitan money and Large house luxury'.

For this dataset in particular, we have the following number of households belonging to each group and category as shown on *Table 2*:

Table 2: Number of households belonging to each CACI category and group

Group	Category	Count	Percentage of Total
ACORN-	ACORN-	2	0.036%
ACORN-B	Affluent	25	0.449%
ACORN-U	ACORN-U	49	0.880%
ACORN-I	Comfortable	51	0.916%
ACORN-O	Adversity	103	1.851%
ACORN-P	Adversity	110	1.976%
ACORN-J	Comfortable	112	2.012%
ACORN-M	Adversity	113	2.030%
ACORN-C	Affluent	151	2.713%
ACORN-N	Adversity	152	2.731%
ACORN-A	Affluent	157	2.821%
ACORN-K	Adversity	165	2.964%
ACORN-G	Comfortable	205	3.683%
ACORN-D	Affluent	292	5.246%
ACORN-L	Adversity	342	6.144%
ACORN-H	Comfortable	455	8.175%
ACORN-F	Comfortable	684	12.289%
ACORN-Q	Adversity	831	14.930%
ACORN-E	Affluent	1567	28.153%

For the purpose of this analysis we will only be using the 'Category' classification. Henceforth, we have the following household distribution shown on *Table 3*:

Table 3: Number of households belonging to each CACI category

Category	LCLid
ACORN-	2
ACORN-U	49
Adversity	1816
Affluent	2192
Comfortable	1507

As for the tariff information, presented in the data in the column 'stdorToU', as mentioned before, some households, during the period of 2013, were subject to dynamic pricing. Three types of tariffs were applied: high, low and normal. The remaining households who were not subject to dynamic pricing had a fixed tariff. Those who were subject to dynamic pricing are referenced as 'ToU' and those with fixed prices are referenced as 'Std'. There are 1123 households with ToU and 4443 with Std.

2.1.3.2 Half Hourly Dataset

Table 4 shows how the dataset looks like if we print the first rows. As mentioned in the previous Section, it has the household ID: 'LCLid', the timestamp 'tstp' showing the year, month, day, hour, minute and second of the reading and the consumption of energy for that timestamp 'energy(kWh/hh)' presented in Kilowatt-hour per half-hour. There are 167,817,021 rows and, therefore, 167,817,021 energy measurements.

Table 4: First rows of the Half Hourly dataset

LCLid	Tstp	energy(kWh/hh)
MAC000027	2011-12-07 11:30:00.0000000	0.185
MAC000027	2011-12-07 12:00:00.0000000	0.155
MAC000027	2011-12-07 12:30:00.0000000	0.147
MAC000027	2011-12-07 13:00:00.0000000	0.164
MAC000027	2011-12-07 13:30:00.0000000	0.187

2.1.3.3 Daily Dataset

Table 5 shows how the dataset looks like if we print the first rows. As mentioned in the previous Section, it has the household ID: 'LCLid', the date: 'day', the median, mean, maximum, standard deviation, minimum and sum of the energy consumed on the corresponding day. It also has the number of times there was a consumption reading on the corresponding day for every household: 'energy_count'. All energy measurements are in Kilowatt-hour per half-hour.

Table 5: First rows of the Daily dataset

LCLid	day	energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
MAC000131	2011-12-15	0.485	0.432	0.868	22	0.239	9.505	0.072
MAC000131	2011-12-16	0.142	0.296	1.116	48	0.281	14.216	0.031
MAC000131	2011-12-17	0.102	0.190	0.685	48	0.188	9.111	0.064
MAC000131	2011-12-18	0.114	0.219	0.676	48	0.203	10.511	0.065

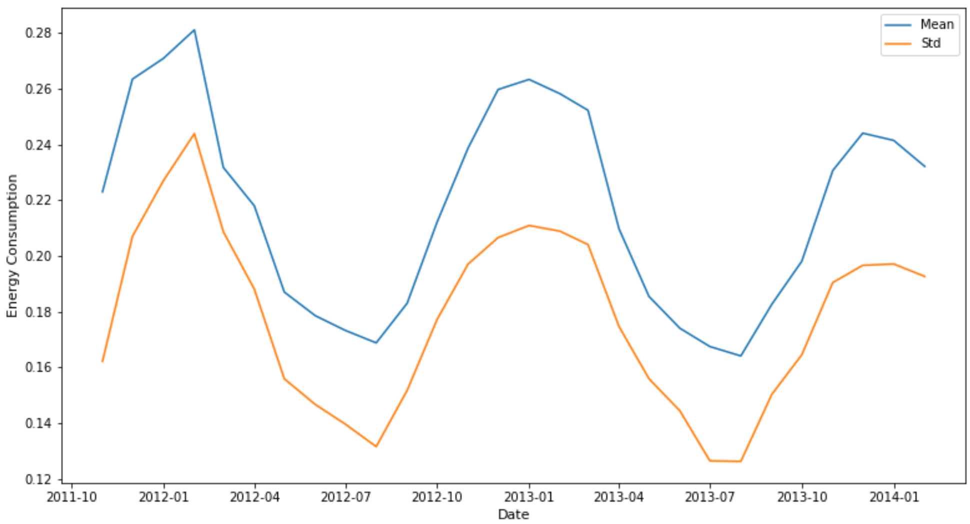
Table 6 shows the basic statistical summary:

Table 6: Basic statistical summary for each variable

	energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
mean	0.159	0.212	0.835	47.804	0.173	10.124	0.060
std	0.170	0.191	0.668	2.811	0.153	9.129	0.087
min	0	0	0	0	0	0	0
25%	0.067	0.098	0.346	48	0.069	4.682	0.020
50%	0.115	0.163	0.688	48	0.133	7.815	0.039
75%	0.191	0.262	1.128	48	0.229	12.569	0.071
max	6.971	6.928	10.761	48	4.025	332.556	6.524

This data set, having the consumption on a daily basis, came very handy since trying to visualize the half hourly consumptions is extremely hard. So, if any clear pattern were to be identified, using the daily aggregated data was the best option. However, for our predictive work, we will not use it.

Figure 1: Energy consumption for the period 2011-2014



As expected, it is already possible to visualize a pattern on consumption. *Figure 1* shows how there is a clear cycle by plotting the mean energy consumption as well as the standard deviation.

2.1.3.4 Weather Daily

Table 7 shows how the dataset looks like if we print the first rows. For simplicity we will only keep the maximum temperature (‘temperatureMax’) in Fahrenheit and its associated timestamp (‘temperatureMaxTime’), the pressure (‘pressure’) in millibars and wind speed (‘windSpeed’) in miles per hour. The idea of using this daily information is the same as with the daily energy consumption data: pattern identification.

Table 7: First rows of the Weather dataset

temperatureMax	temperatureMaxTime	pressure	windSpeed
11.96	2011-11-11 23:00:00	1016.08	3.88
8.59	2011-12-11 14:00:00	1007.71	3.94
10.33	2011-12-27 02:00:00	1032.76	3.54
8.07	2011-12-02 23:00:00	1012.12	3.00

Figure 2: Pressure behaviour for the period 2011 - 2014

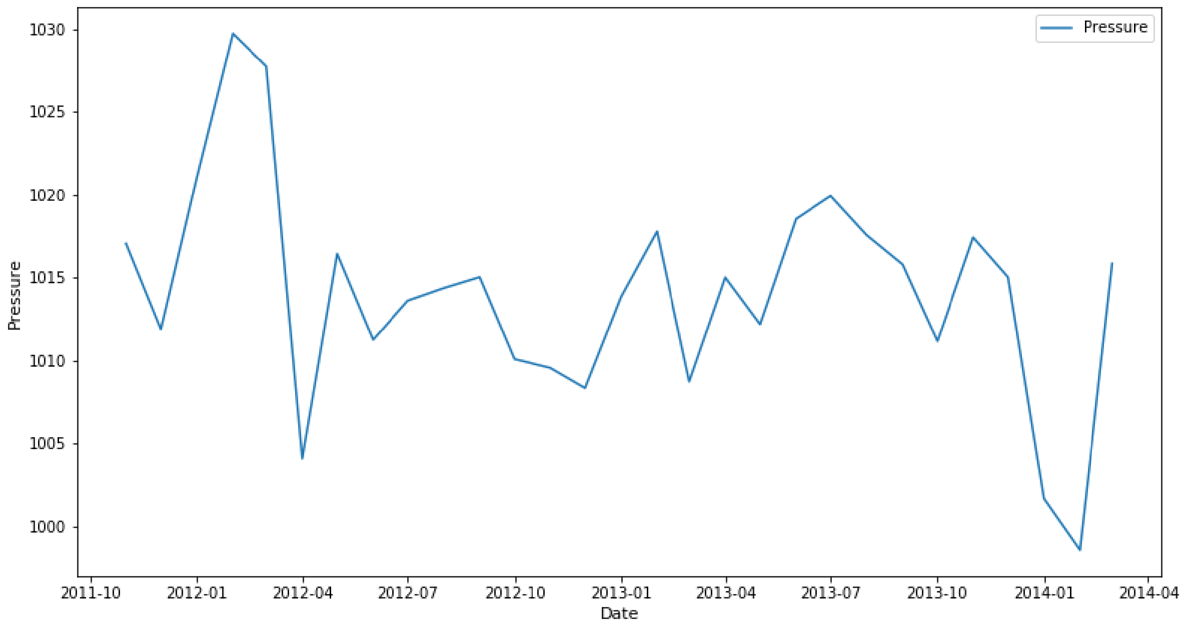
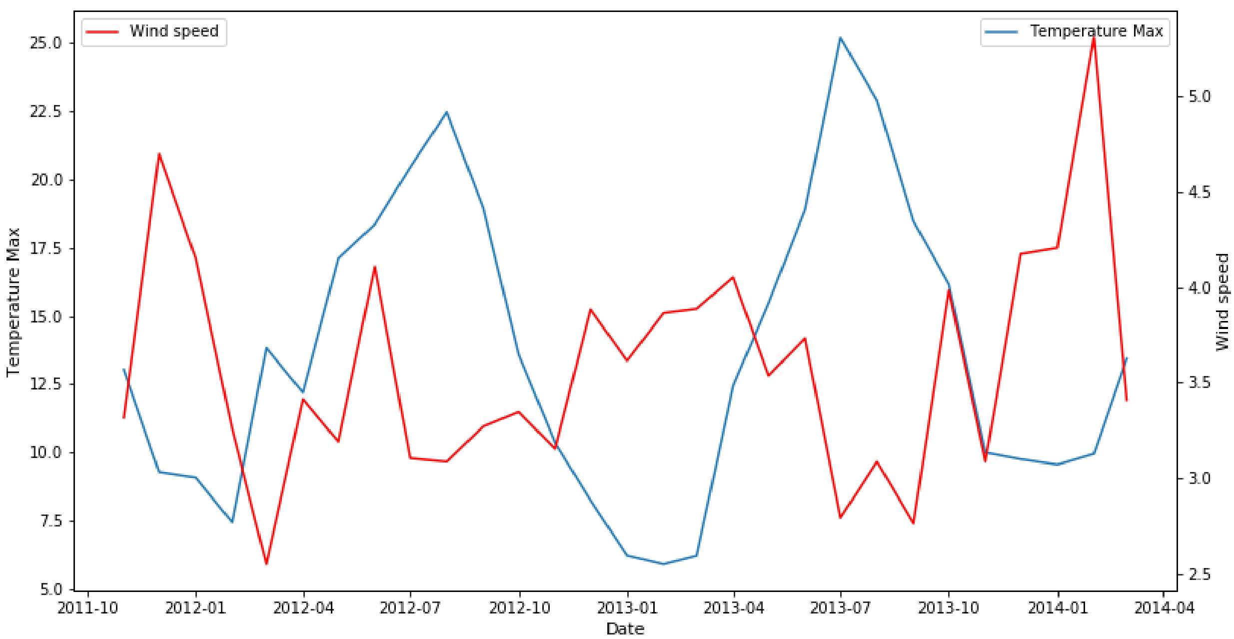


Figure 3: Wind and Temperature behaviour for the period 2011 - 2014



With *Figure 2* and *Figure 3* it is already possible to identify that energy consumption and weather variables have some correlation. The temperature and the wind speed seem to behave in an opposite way. Most importantly, they have a pattern that can be clearly related to the energy consumption. As for pressure, the behaviour is not clear and the year 2012 seems to be an outlier.

2.1.3.5 Weather Hourly

Table 8 shows what the dataset looks like if we print this dataset's first rows. Because there was no available data for weather information in a 30-minute interval, we used this information and duplicated the information for every timestamp, creating a database with information every half hour. That dataset will be used for the predictive analysis. Visibility is in miles, wind bearing in degrees, temperature and dew point in Fahrenheit, and pressure in millibars.

Table 8: First rows of the Weather Hourly dataset

timestamp	visibility	windBearing	temperature	dewPoint	pressure
01/01/2012 00:00	12.99	229	12.12	10.97	1008.10
01/01/2012 00:30	12.99	229	12.12	10.97	1008.10
01/01/2012 01:00	12.89	238	12.59	11.02	1007.88
01/01/2012 01:30	12.89	238	12.59	11.02	1007.88
01/01/2012 02:00	11.54	229	12.45	11.04	1007.95

2.2 Methods

In this Section we detail on the models, metrics and validation strategy used in our predictive analysis. The models are presented on Section 2.2.1 and the metrics on Section 2.2.2.

2.2.1 Models

We will perform three different models in an attempt to predict the consumption for each individual household. The first model is an ARIMA model and the second and third ones are LSTM. The difference between the two LSTM models is that the second model involves covariates.

In this Section we present in total five models: The *average consumption* model is presented in 2.2.1.1, *average consumption by timestamp* on Section 2.2.1.2, *ARIMA* on Section 2.2.1.3, *LSTM Univariate* on 2.2.1.4 and finally model *LSTM Multivariate* on 2.2.1.5.

2.2.1.1 Model 1: Average consumption

This model has the intention to serve as a benchmark. It is an extremely simple model where, after splitting the data into train and test groups, we average the consumption for the train group and take that average as our prediction. This is done for each household.

To place it formally, $\hat{y}$ is our prediction were s represents each individual household and the total amount of available timestamps in our dataset is given by t, …, $|T|$.

$$\widehat{y_s} = \frac{\sum_{i=1}^{|T|} y_{s_t}}{|T|}$$

2.2.1.2 Model 2: Average consumption by timestamp

The previous model may be a good benchmark but in order to push the more advanced models (ARIMA/LSTM) even further, we can come up with a smarter averaging model.

We will average the consumption on the train data for each day of the week and for each timestamp. For instance, we will group all the energy readings that were recorded on Mondays at each timestamp (48 each day) for the data contained on the train group and average them. For example, for Monday we calculate the average consumption at timestamp 00:00 AM (midnight), then for 00:30 AM (half past twelve) and so on. That will be repeated for each day of the week.

To place it formally, remembering that $\hat{y}$ is our prediction and s represents each household, we can put as an example the data for all Mondays at timestamp 00:00. So, for each household, s, we sum the information for all available data for Mondays at 00:00 and divide by the count of Mondays at 00:00 given by t, …, $|T|$:

$$\hat{y}_s = \frac{\sum_{i=1}^{|T|} y_{s_t}}{|T|}$$

We will have for each unique timestamp (48 per day) and for each day of the week the average of energy consumption.

2.2.1.3 Model 3: ARIMA(p,d,q)

ARIMA stands for Auto Regressive Integrated Moving Average (Seymour et al., 1997). The key aspects are:

- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and a number of lagged observations.
- **I**: *Integrated*. The use of differencing of raw observations in order to make the time series stationary.

- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

So, we have three hyper parameters to account for: p, d, q, where:

1) p accounts for the periods to lag and helps adjust the line that is being fitted to forecast the time series.
2) d accounts for the number of differencing transformations required by the time series to be stationary. This is because it is easier to predict when the mean and variance are constant over time.
3) q accounts for the lag of the error component, the error being a component of the time series that is not explained by trend or seasonality.

Being Y_t our variable of interest at the moment in time t, the proposed model has the following specification,

$$Y_t = \alpha + \beta_1 Y_{t-1} + \cdots + \beta_p Y_{t-p} + \varepsilon_t + \varphi_1 \varepsilon_{t-1} + \cdots + \varphi_q \varepsilon_{t-q}$$

where α, β_1, β_p, φ_1, φ_q are parameters within the model and ε_t is the error at t which is assumed as random.

Normally either the AR term or the MA term is used, both being used in rare occasions. In order to choose correctly between AR and MA, we use autocorrelation and partial autocorrelation plots. Autocorrelation refers to how correlated a time series is with its past values. The *auto correlation function* (ACF) plot shows the correlation between Y_t and Y_{t-k} for different values of k. The correlation coefficient is in the y-axis and the number of lags is shown in the x-axis. The *partial autocorrelation function* (PACF) plot measures the relation between Y_t and Y_{t-k} after removing the lag 1,2, 3, …, k -1. This plot is particularly useful since the ACF plot may show a correlation between Y_t and Y_{t-2} simply by being connected to Y_{t-1}.

Therefore, by using ACF plot we can choose between MA and AR models as follows:

- If there is a positive autocorrelation at lag 1, then we use AR model

- If there is a negative autocorrelation at lag 1, then we use the MA model.

There are two ways to determine if the time series is stationary: rolling statistics and the Augmented Dickey-Fuller Test. We will use the latter.

Taking all this into consideration, in this work, the steps that will be followed to construct an ARIMA model are:

1) Make the series stationary, if necessary, by differencing. To determine this, we will use the Dickey-Fuller test.
2) Study the pattern of autocorrelations and partial autocorrelations with the Autocorrelation function plot (ACF) and the Partial Autocorrelation Function plot (PACF) respectively.
3) Fit the model.

2.2.1.4 Model 4: Artificial Neural Network (LSTM) – Univariate Case

LSTM Neural Networks stands for Long Short-Term Memory, and so can act as long-term or short-term memory cells (Jason, 2018). The output is modulated by the state of the cells and this is important since we need the prediction of the neural network to depend on the historical context of inputs. Their most common domain is in natural language processing, speech recognition and image recognitions amongst other applications.

The idea behind LSTM is that its predictions are always tied by past inputs. However, as time passes, the next input will most probably not be tied to a very old input and LSTM manages this by learning when to remember and when to forget through their forget gate weights. The two hyper parameters that we will tune are the number of epochs and the number of neurons. One last important aspect for our LSTM model is that we will use the chosen metric to measure the model performance (RMSLE) as the loss function used to learn the model weights (more on this in Section 2.2.2).

2.2.1.5 Model 5: Artificial Neural Networks (LSTM) – Multivariate Case

This model is identical to Model 4 with the sole difference of having covariates with the hopes of improving the predictions. The covariates are the weather variables discussed in the 'Materials' Section. Besides that, everything else is equal relative to the LSTM univariate model.

2.2.2 Metrics

We use one metric to compare the different model performances, and that is RMSLE. RMSLE penalizes the underestimation of the actual value more severely than it does for overestimation. This last property is highly valuable for Load Forecasting as underestimation entails not being able to meet the demand of energy.

$$RMSLE = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(\log(p_i + 1) - \log\,(a_i + 1))^2\right)}$$

Following the formula stated above, p_i is the prediction for a given period *t* and a_i is the true value observed in period *t*. The sum is done for each value that is predicted and its true value given by *i*=1 up to n. In our case, these represent the timestamps.

Moreover, we also included RMSE as a control metric to see if the results would hold against RMSLE or if they changed. RMSE formula follows the standard deviation of the residuals, so the metric measures how spread out the residuals are.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - o_i)^2}{n}}$$

Following the formula stated above, $p_i$stands for the forecast – in our case for a given period *t,* and o_i stands for the observed value – again, for a given period *t.* The sum is done for each value that is predicted and its true value given by *i*=1 up to *n*. In our case, these represent the timestamps.

2.2.3 Validation

Neural nets have the complexity of hyperparameters that need to be calibrated. It is the most obvious distinction from statistical models such as ARIMA where parameters are decided with statistical tools. Machine learning models' parameters are calibrated by testing a range of different values and once the best combination is decided, the model can be executed. The way we test different values for our parameters is by proposing a validation scheme.

Once we have our dataset, we will divide it in three groups called *train*, *validation* and *test*, roughly in an 80%-10%-10% proportion. After this, we will decide upon a range of values for our hyperparameters and with our *train* data we will run our model and validate our result with our *validation* data. This will be done for each combination of our hyperparameter's candidate values by observing the lower RMLSE in the validation dataset. Once finished we will decide which is the best option and combine our *train* and *validation* data into a new *train* group. Once grouped, we will run our model with the optimum combination of parameters and test the results with our *test* data.

On Section 4.1.2 we will discuss our range of possible candidates for our LSTM models.

3 Data Exploratory Analysis

In this Section we will perform a deeper analysis of our data by joining the databases. In Section 3.1 we will join and explore the *household* data with *daily consumption.* In Section 3.2 we will join and explore the *household* data with *daily consumption* and *weather daily.* In Section 3.3 we will join and explore the *household* data and *daily consumption* data again but this time for analysing consumption behaviour regarding the tariffs paid. In Section 3.4 we will clean the data for missing values and finally in Section 3.5 we will present a summary of any conclusions on our data.

3.1 Household Information – Daily Consumption

Given the data presentation in Sections 2.1.3.1 – 2.1.3.5, we thought that as a first more fine-grained exploratory analysis it would be interesting to merge the household dataset with the daily consumption dataset. The idea behind this is the categorization of consumers into groups (acorn) and if this has any translation into their consumption behaviour.

The first step was to merge both databases using LCLid as the key. After that, we selected the columns: ‘LCLid’, ‘day’, ‘energy_mean’, ‘Acorn_grouped’ resulting in the following datasets (I’m only showing the first rows) as shown on *Table 9*:

Table 9: First rows of the join between Household Information and Daily Consumption dataset

LCLid	day	energy_mean	Acorn_grouped
MAC000131	2011-12-15	0.432	Affluent
MAC000131	2011-12-16	0.296	Affluent

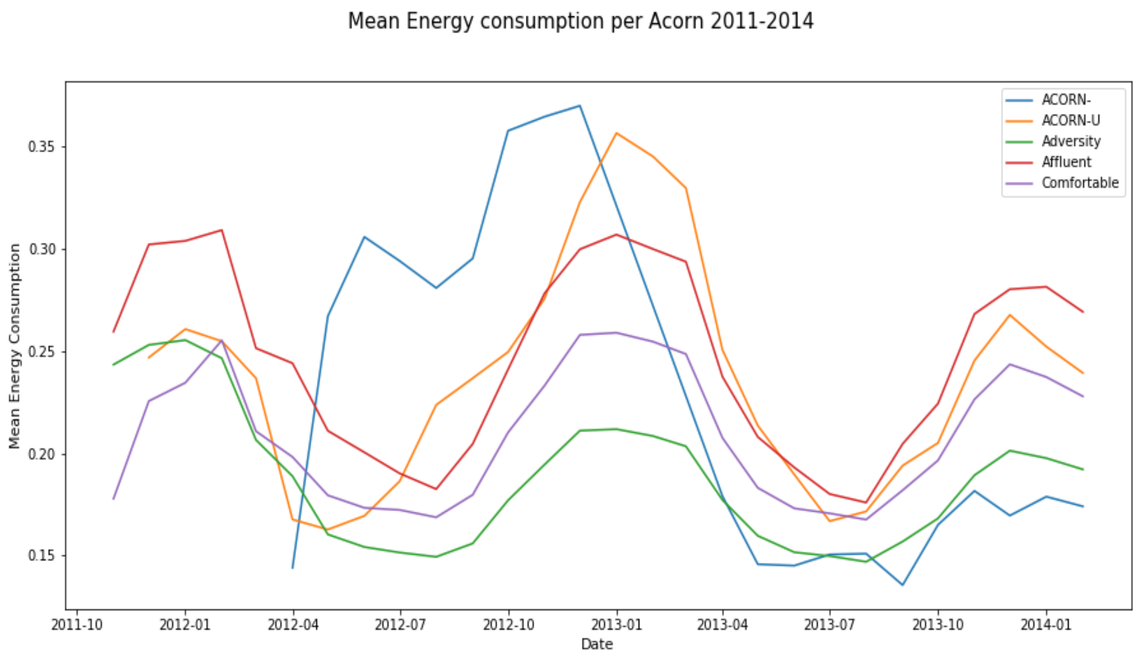
The first question that rose is how balanced the dataset regarding the Acorn categorization was. The result was an unbalanced dataset where the Affluent group dominates the rest of the groups as almost 40% of the total in contrast with ACORN- and ACORN-U who do not reach the 1% as shown on *Table 10*.

Table 10: Number of households belonging to each ACORN

Acorn_grouped	Count	% of total
ACORN-	2	0.04%
ACORN-U	49	0.88%
Adversity	1816	32.63%
Affluent	2192	39.38%
Comfortable	1507	27.08%

Next, the best way to see if there was a pattern in consumption between groups, was visually as shown on *Figure 4*:

Figure 4: Mean Energy Consumption per Acorn during the period 2011 - 2014



Effectively, there seems to be a difference in the mean consumption of energy per acorn. However, they all have the same consumption pattern. The only group that seems to be independent from the rest is 'ACORN-' but as of 2013 it seems to join the rest. After further analysis, we came to the realization that 'ACORN-' was missing the first three months of 2012 as well as the available data from 2011. Moreover, this group is only 0.02% of the total dataset. For this reason, we decided to exclude this acorn group for any further analysis. Regarding ACORN-U, even though the data is available for all months, since we have a very small sample (0.84%) we also decided to exclude this group from any further analysis.

To analyse if the consumption is statistically different between groups, we performed t-tests comparing each group's average consumption. The null hypothesis is that there is no difference in the mean of consumption between any of the two acorn groups. In total we did three t-test: Affluent/Adversity, Affluent/Comfortable, Adversity/Comfortable. The p-value of each t-test is shown on *Table 11*:

Table 11: Mean and p-value for each t-test

	Mean for each group	**p-value**
Affluent/Adversity	0.2400123 = 0.178	0
Affluent/Comfortable	0.2400123 = 0.209	0
Adversity/Comfortable	0.1779077 = 0.209	0

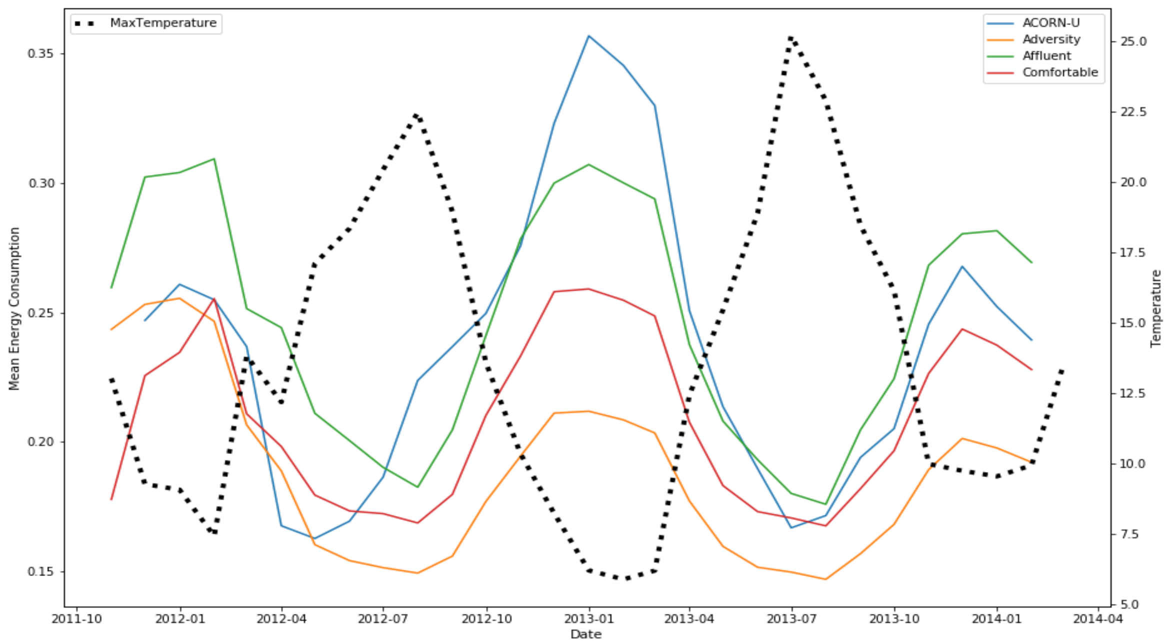
So, for every case, the null hypothesis is rejected at standard significance levels. For this reason, the groups will be analysed separately in any further analysis.

3.2 Consumption – Climate

Temperature is usually related to energy consumption levels, so as a next step, we wanted to compare the climate database with the dataset created for the 'Household Information – Daily Consumption' analysis. The idea here was to find evidence on the need of adding weather as covariates for the predictive analysis.

The result is better seen graphically on *Figure 5*: when the temperature rises, the energy consumption decreases and vice-versa. So, during summer the demand for energy decreases and during wintertime it increases.

Figure 5: Mean Energy Consumption per Acorn and Temperature for the period 2011 – 2014



To further see any correlation between them, we present a correlation map on *Table 12*. With this map, should we add any weather as covariates in any predictive model, we can be sure it will be an asserted decision.

Table 12: Correlation Map between covariates and the mean energy consumption

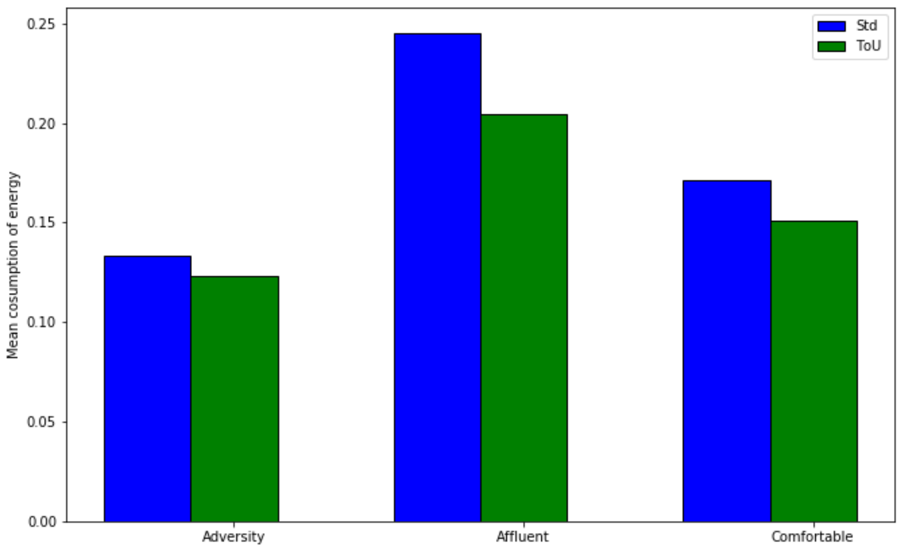
	visibility	windBearing	temperature	dewPoint	pressure	Apparent Temperature	WindSpeed	humidity	energy_mean
visibility	1.000	0.206	0.736	0.653	-0.107	0.724	-0.021	-0.625	-0.703
windBearing	0.206	1.000	0.087	0.183	0.104	0.108	-0.154	0.293	0.002
temperature	0.736	0.087	1.000	0.970	0.018	0.998	-0.404	-0.564	-0.945
dewPoint	0.653	0.183	0.970	1.000	-0.059	0.974	-0.384	-0.349	-0.894
pressure	-0.107	0.104	0.018	-0.059	1.000	0.018	-0.397	-0.275	0.149
Apparent Temperature	0.724	0.108	0.998	0.974	0.018	1.000	-0.406	-0.548	-0.947
windSpeed	-0.021	-0.154	-0.404	-0.384	-0.397	-0.406	1.000	0.239	0.310
humidity	-0.625	0.293	-0.564	-0.349	-0.275	-0.548	0.239	1.000	0.616
energy_mean	-0.703	0.002	-0.945	-0.894	0.149	-0.947	0.310	0.616	1.000

3.3 Tariff Analysis

Tariff discrimination seemed at first like a key component for the understating of consumer's behaviour. As mentioned before, during the year 2013 an experiment on dynamic pricing was done in random households, so it felt natural to see if there was any statistical evidence on a change in the consumption behaviour.

The first step was to graphically see if there was any obvious difference amongst the groups; there doesn't seem to be big differences as shown on *Figure 6.*

Figure 6: Mean Energy Consumption per Acorn during the year 2013



However, we performed a t-test to be sure that there is enough statistical evidence that the means between groups can be said to be equal for the year 2013 (year in which the difference in tariffs was made). So, we filtered the database for the year 2013 and then splitted between those with 'Std' tariff and those with 'ToU' tariff. To even further deepen the analysis, we splitted the result between Acorn groups. Consequently, we had the three Acorn groups for the 'Std' tariffs and three Acorn groups for the 'ToU' tariff.

We then performed the t-test where the null hypothesis was equal means between groups (i.e. Affluent (Std) and Affluent (ToU)), so in total we had three t-tests. The p-value of each t-test is shown on *Table 13*:

Table 13: Mean and p-value for each t-test

Group	Mean for each group	p-value
Adversity (Std) / Adversity (ToU)	0.133572 = 0.123	0
Affluent (Std) / Affluent (ToU)	0.245382 = 0.205	0
Comfortable (Std) / Comfortable (ToU)	0.171018 = 0.151	0

To conclude, with a confidence level of 1%, it is possible to say that there is enough statistical evidence to reject the null hypothesis. The difference in means between groups is statistically positive.

Same as for the difference in energy consumption, the groups will be analysed separately in any further analysis.

3.4 Data Cleaning

Data cleaning was only performed on the half hourly dataset since it will be used for the predictive analysis. Before performing any predictive analysis, it was necessary to see if the available data was clean. Most importantly, because the predictive analysis is done on an individual basis, it was important to know if every household had smart meter readings throughout the period of analysis.

The first step on seeing how clean the data was, was looking for any Null values. To our surprise we found 5560 null values on 5566 houses (only one null value per household) on random timestamps like shown on the following sample on *Table 14*:

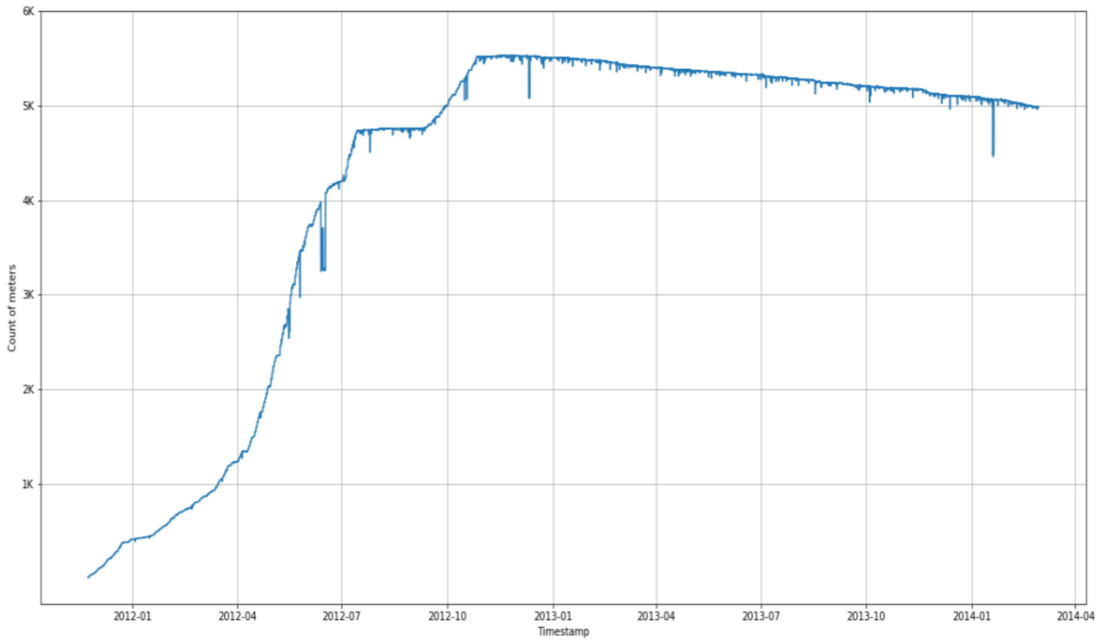
Table 14: Sample of households presenting Null values for Energy consumption data

LCLid	TimeStamp	energy(kWh/hh)
MAC000027	2012-12-18 15:13:41	Null
MAC000406	2012-12-18 15:15:33	Null
MAC000492	2012-12-18 15:15:43	Null
MAC000512	2012-12-18 15:15:45	Null
MAC000726	2012-12-18 15:16:42	Null

So, 99.99% of our available households have a random *Null* reading. These readings were deleted with no trouble at all since we are only interested in readings every 30' (every hour and half hour).

Next, *Figure 7* shows the count of unique meters per day throughout the month of available data. Here we realized that if the household had smart meter readings in January 2013, for instance, it does not imply they will have readings in February 2013 or any further month. This is because the household could 'leave' the smart meter program.

Figure 7: Count of meters for every timestamp available from November 2011 to February 2014



This brought a new question: how many households have complete data? The answer is none. *Table 15* shows the count of unique timestamps for each household and, should we have no missing values, we should have 39792 rows (the data provided starts on 23/11/2011 and finishes on 28/02/2014 and we have 48 reading per day). As shown on *Table 15*, no household has information for the whole period provided.

Table 15: First entries of the count of readings per household between 2011 and 2014 in descending order.

LCLid	Count
MAC000145	39725
MAC000147	39725
MAC000150	39720
MAC000152	39719
MAC000149	39718

If we consider 2012 and 2013 (the only complete years available), the result should be of 35088 unique values if complete and *Table 16* shows us that effectively we do have households with complete data for that period.

Table 16: First entries of the count of readings per household between 2012 and 2013 in descending order.

LCLid	Count
MAC004463	35088
MAC000019	35088
MAC000049	35088
MAC000131	35088

But, for how many households? *Table 17* shows only 12 households are complete.

Table 17: Number of houses with 35088 (or less) consumption readings between the years 2012 – 2013

Amount of data per household	Count
35088	12
35087	33
35086	57

This posed the challenge to decide if we were willing to fill in missing data. Reducing our data from 2011-2014 to 2012-2013 periods was not hard since only 5 houses were lost in that decision, but from the remaining 5561 reducing them to 12 households because of missing data was a huge leap.

Missing data is complicated to manage since we have to come up with the best solution to fill in those gaps without interfering with the true behaviour the consumer would have shown. It is important to highlight that this problem is not particular to our data, smart meters data is prone to missing values and both researchers and professionals are faced with this challenge. There are estimation methods based on statistics and machine learning, but we found that an estimation method that works very well for industrial wireless sensor networks is *Last Observation Carried Forward (LOCF).* (Zhou et al., 2018) show in their work how this method 'can acquire a high reconstruction accuracy for large time series data which changes stably'. So, for our case, we came to the conclusion that the best solution would be to follow LOCF, such that if the timestamp '1/1/2012 00:30' data was missing it was filled with the value from the previous timestamp '1/1/2012 00:00'.

Moreover, this opens the possibility to data leakage. We know in advance that during the predictive analysis we will be dividing our data into train, validation and test groups

(Section 2.2.3) and if we happen to fill in a value from validation with train data or a value from test with validation data we would be effectively having data leakage problems. So, in order to avoid this, missing values from each group (train, validation, test) will only be filled with data from the same group.

The next decision towards missing data was how many values were we willing to fill. And for this, we thought that since we wanted to understand consumer's behaviour on an individual scale and with high frequency data, adding in too much data would interfere with true behaviour. Hence, we did not want to add too many values that could prevent us from understanding individual household consumption beyond their aggregated behaviour.

Finally, we chose to keep those households with more than 35000 data points since over 99% of their original data is present. Again, we are interested in the consumer on an individual basis so having to keep fewer houses in order to have houses with as much real data as possible seems like the lesser evil. Having fewer houses was not going to affect the performance of the models but having more altered data might have.

After dividing each household into each group, we have the following distribution shown on *Table 18*:

Table 18: Number of households belonging to each group (ACORN – Tariff)

Group	Count	Percentage
Adversity - Std	116	31 %
Adversity – Tou	22	
Affluent – Std	94	40 %
Affluent – Tou	12	
Comfortable – Std	85	29 %
Comfortable – Tou	14	

If we would have kept all the households with missing values, *Table 19* would look like *Table 19:*

Table 19: Number of households belonging to each group (ACORN – Tariff)

Acorn_grouped	Count	% of total
Adversity	1816	32.93%
Affluent	2192	39.75%
Comfortable	1507	27.33%

Taking this into consideration, our sample of 343 households (with over 35000 data points) resulted in an almost perfect proportionate subsample. In our sample Adversity represents 31% when it used to be 32.93%, Affluent 40% when it used to be 39.75% and Comfortable 29% when it used to be 27.33%.

After cleaning the data, *Figure 8* shows the daily consumption for January, February and March 2012 for a complete household, ID: 'MAC004463', but comparing the three months, we are not able to see any clear pattern between months.

However, there seems to be a cyclical behaviour on smaller windows, such as daily, shown by the small peaks and valleys that repeat themselves.

To further investigate this, we have a daily graph, *Figure 9*, for the month of January 2012 for the same household. Again, no clear pattern is present. This entails that perhaps more complex models such as Artificial Neural Networks may prove to be better suited as they easily incorporate covariates that may help explain erratic behaviour.

Figure 8: Household MAC004463's Energy Consumption in KWh/HH for January, February and March 2012

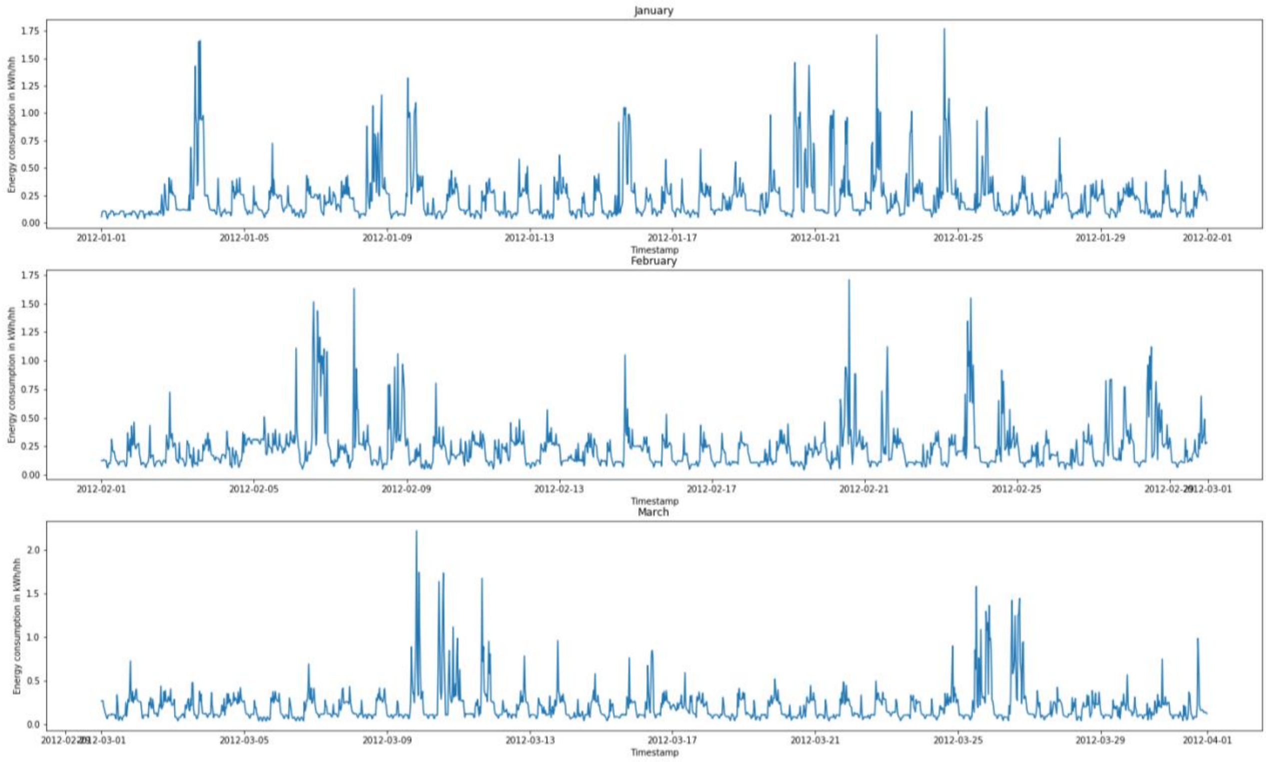
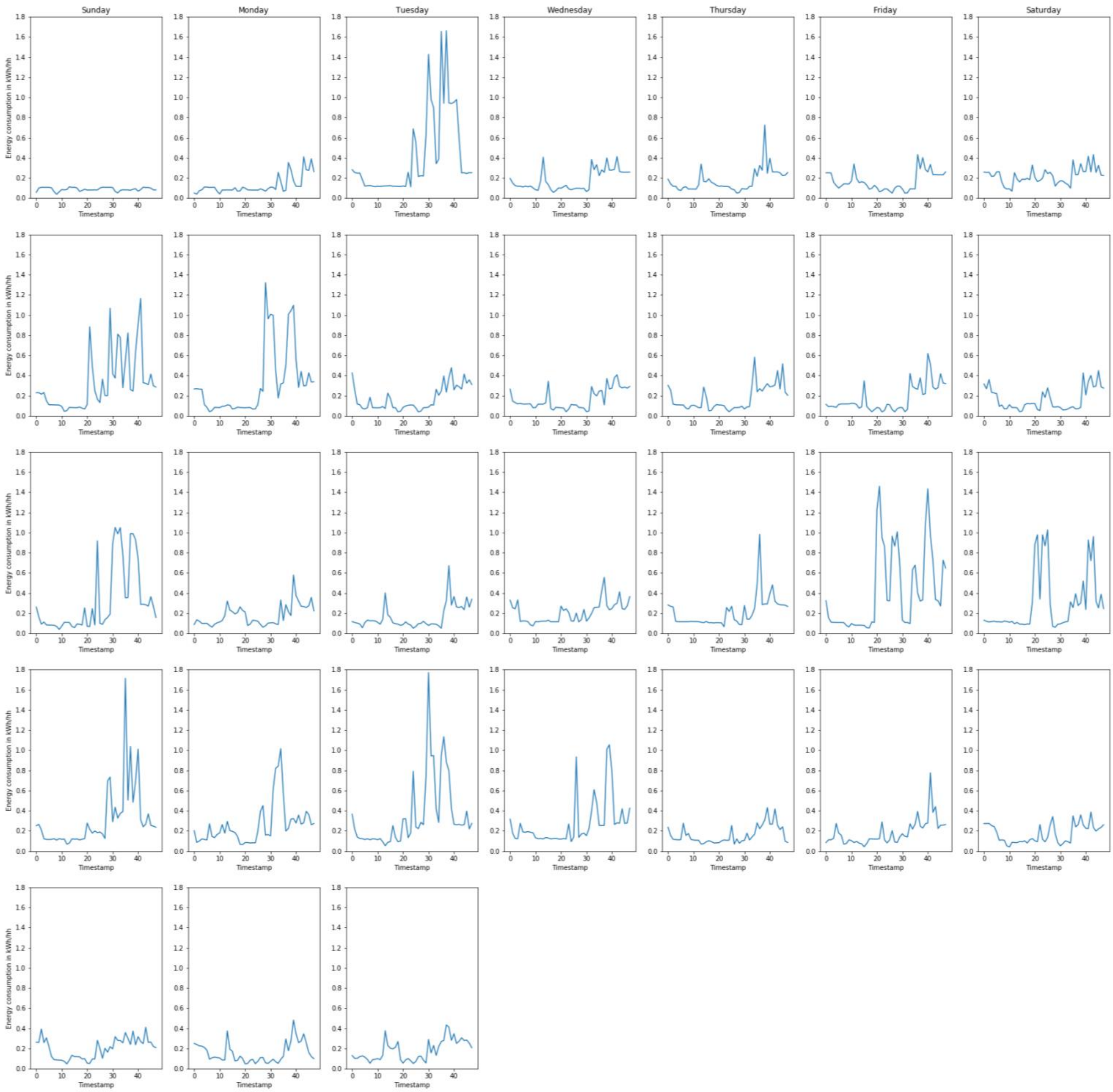


Figure 9: Household MAC004463's Energy Consumption in KWh/HH for January 2012



3.5 Summary

To sum up this Section and keeping in mind we will be analysing the individual consumption of households, after the descriptive analysis it seemed appropriate to only keep data for the year 2012 and 2013. These are the only complete years where we can appreciate the cyclic consumption of the households.

Moreover, we will only be analysing households belonging to the ACORN 'Affluent', 'Adversity' and 'Comfortable' since 'ACORN-' and 'ACORN-U' only represent less than 1% of the data. Also, because of statistical evidence on a difference in consumption between them, we will analyse them separately. The same applies for the difference between households subject to dynamic or fixed tariffs (more on how this differentiation will be done on the next Section).

Furthermore, as seen on the correlation map of weather and energy, it is possible to use them as covariates in order to explain the consumption of energy.

Finally, we had to decide on the best strategy to deal with missing values we had across almost all households. We settled on keeping those with over 99% of their data and therefore kept 343 households.

4 Predictive Analysis

In the previous section we had to decide on the best strategy to deal with missing values we had across almost all households. We settled on keeping those with over 99% of their data and therefore kept 343 households. Moving on to the predictive analysis, given that decision, we will have 343 results for each model.

Let us recall that because of statistical evidence on the difference in consumption amongst the ACORN classifications and type of tariff, we will group the households by ACORN group and type of tariff applied. Moreover, we will tune, if necessary, each group independently. Therefore, we have six different groups of households: those belonging to the ACORN Affluent and had simple tariff, Affluent with dynamic tariff, Adversity with simple tariff, Adversity with dynamic tariff, Comfortable with simple tariff and Comfortable with dynamic tariff. For all cases, the length of each household data is 35088 observations: 366 days for the year 2012 with 48 readings per day and 365 days for the year 2013 with 48 readings per day. For all models the consumption data is simply the energy consumption in KWh/hh (Kilowatt-hour / half-hour) for every half hour and for the LSTM multivariate case, we include the weather variables. These weather variables are visibility, wind bearing, temperature, dew point, and pressure; the information frequency is the same as that of consumption so for every half hour there is a reading.

The implementation for all models is in Python and the libraries used across all models are *Pandas*, *NumPy*, *glob*, *matplotlib*, and *sklearn*. *Pandas* and *Numpy* libraries were used to use and manipulate the data in the pandas dataframe format, and most specifically to take advantage of *Panda's* support of datetime data. Every household consumption data was on a separate *comma separated value* (csv) file and in order to read them all together in an efficient way, the *glob* library was used. Finally, *matplotlib* was used for all visualizations and *sklearn* for the metrics. Specifically, for the ARIMA model implementation the *statsmodel* library was used and for LSTM model the *keras* and *math* library - both for univariate and multivariate models.

The input for the models varied from model to model, where for Model 1, 2 and 3 the input was simply the dataframe corresponding to each household divided in train-test accordingly. For model 4 and 5 (LSTM Univariate/Multivariate) even though the raw input was also in a dataframe format corresponding to each household - divided in train-validation-test accordingly, the input was further reshaped to be in a 3D format corresponding to samples, timesteps, and features necessary for neural network models to work.

Finally, for the predictive analysis process we could have incorporated feature engineering transformation, more specifically, scaling. However, we decided to opt out because the values of our data are very small. For instance, if we recall section 2.1.3.3 (Daily Dataset), we show on table 6 the basic statistical summary for the daily energy consumption dataset. In that dataset we have the mean energy consumption and, given the statistical summary, the mean is equal to 0.21173. Because of this characteristic of our data, we did not see necessary to apply a scaling method.

4.1 Setting Parameters and Hyperparameters

As mentioned in the Models Sections, ARIMA and LSTM have parameters and hyperparameters that need to be found and tuned. This Section is dedicated to that. We will present one case for each group and the rest can be found Annexed.

4.1.1 ARIMA

Since we are forecasting each individual household's time series, there is no clear strategy on how to construct the model since constructing an ARIMA for each household seemed very unpractical.

Because of how we divided the households, one clear option was to construct an ARIMA for each group: Affluent-Std, Affluent-Tou, Adversity-Std and so on. However, we still have various houses in each group, so deciding on which model to use for each group was not trivial. The reached compromise was to randomly choose 5% of the households for each group and construct an ARIMA model for each one (following the mentioned steps). After that, decide, depending on the results, on which ARIMA to use for each group. We will therefore choose randomly 5% of the households and perform the Dickey Fuller test. Then, we will present the ACF and PACF plots. Based on the autocorrelation plots and the Dickey-Fuller test, we will decide on the best ARIMA model for each group.

4.1.1.1 Conclusion for ARIMA parameters

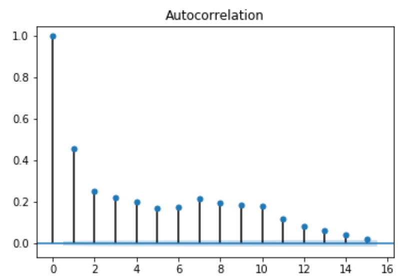
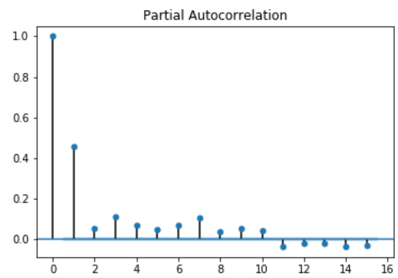
For the group Adversity-Std we had 94 households, and so the 5% to analyse were 5 houses, for Adversity-Tou we had 12 households, and so the 5% to analyse was 1 house. For Affluent-Std we had 116 households, and so the 5% to analyse were 6 houses, for Affluent-Tou we had 22 households, and so the 5% to analyse was 1 house. Finally, for Comfortable-Std we had 85 households, and so the 5% to analyse were 4 houses and for Comfortable-Tou we had 12 households, and so the 5% to analyse was 1 house.

All households analysed came out as stationary after the Dickey-Fuller Test. They also had positive autocorrelation at lag 1 and a slow decay on ACF. So, the hyper parameters chosen are $d = 0$ and $q = 0$.

For the hyper parameter p, 100% of households had two spikes on the PACF plot and therefore p will be equal to 2. Taking all this into account, we decided to choose an ARIMA (2,0,0) for all groups.

As an example, we show the results for the first analysed household for Adversity-Std group, the rest can be found on the annex.

1) Household MAC000019

<table>
<tr><th colspan="2">Dickey – Fuller Test</th></tr>
<tr><td colspan="2">• ADF Statistic: -8.740055
• p-value: 0.000000
• Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

4.1.2 LSTM

Just like for ARIMA, we chose 5% of households for each group. In order to tune the hyperparameters, we did cross validation on the amount of LSTM neurons. Recall Section 2.2.3 for our validation scheme.

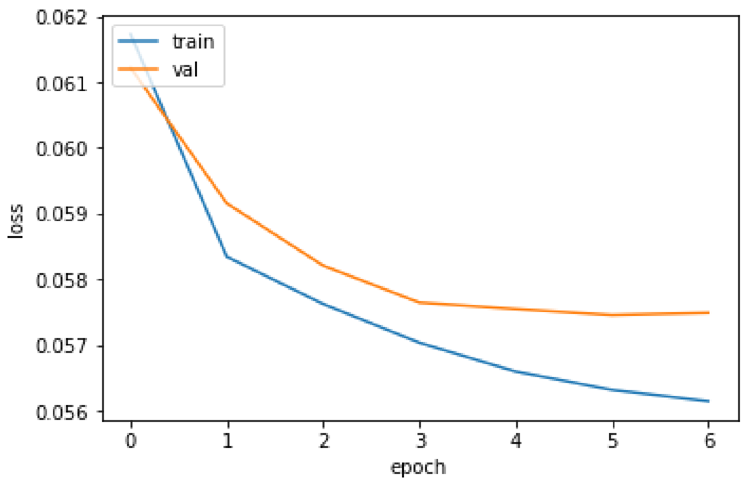
For the batch size, we decided upon 70 since it is a relatively small number yet not big enough that would harm the model's performance. Batch size is the number of samples that are shown to the network before a weight update.

For the number of neurons, we decided upon a range of 50, 100, 150 and 200 neurons since 100 is a standard number to start with and we felt that reducing all the way to 50 and increasing up to 200 was a good range for our model to validate. We trained each group with the different cases of neurons and saved the average RMSLE for each. The model with the lowest *validation* average RMSLE for each group will be the one used for the final results.

For the number of epochs, the decision was not clear and so we implemented what is called an 'early stopping' to halt the training. This hyperparameter is very special because if you pass too many epochs it can lead to overfitting, whereas if you pass too few, it may result in an underfitted model. By applying early stopping we specified an arbitrarily large number of epochs (in our case it was 15, since more than 15 would entail a long running time) and it will automatically stop training once the model performance stops improving. This led to a positive secondary effect, where our models finish running faster since they do not train excessively unless necessary. This conclusion applies for both LSTM Multivariate and LSTM Univariate. The following extract (*Extract 1*) from our results, when running LSTM Univariate for the Adversity-Std group, shows how the epochs stopped at the seventh iteration, how each epoch took approximately 3 seconds to finish and finally a graph showing the loss value at each iteration for training and validation sets – we chose at random one household form each group to see these results, besides the one shown below, the rest are on the annex section (*Extract 2-12*).

Extract 1: Household MAC000127 from Adversity-Std group with LSTM Univariate Model

```
Epoch 1/15
32252/32252 [==============================] - 3s 101us/step - loss: 0.0617 - rmsle_loss: 0.0617 -
val_loss: 0.0612 - val_rmsle_loss: 0.0608
Epoch 2/15
32252/32252 [==============================] - 3s 84us/step - loss: 0.0583 - rmsle_loss: 0.0584 -
val_loss: 0.0592 - val_rmsle_loss: 0.0588
Epoch 3/15
32252/32252 [==============================] - 3s 98us/step - loss: 0.0576 - rmsle_loss: 0.0576 -
val_loss: 0.0582 - val_rmsle_loss: 0.0579
Epoch 4/15
32252/32252 [==============================] - 3s 84us/step - loss: 0.0570 - rmsle_loss: 0.0570 -
val_loss: 0.0576 - val_rmsle_loss: 0.0573
Epoch 5/15
32252/32252 [==============================] - 3s 82us/step - loss: 0.0566 - rmsle_loss: 0.0566 -
val_loss: 0.0575 - val_rmsle_loss: 0.0572
Epoch 6/15
32252/32252 [==============================] - 3s 81us/step - loss: 0.0563 - rmsle_loss: 0.0563 -
val_loss: 0.0575 - val_rmsle_loss: 0.0571
Epoch 7/15
32252/32252 [==============================] - 3s 81us/step - loss: 0.0561 - rmsle_loss: 0.0562 -
val_loss: 0.0575 - val_rmsle_loss: 0.0571
Epoch 00007: early stopping
```



The results are present on *Table 20* for LSTM Univariate and on *Table 21* for LSTM Multivariate. The minimum for each group is in bold.

Table 20: Average RMSLE for each group for the Univariate model

	Number of Neurons			
Group	50	100	150	200
adv_std	0.09200	0.09040	0.09040	**0.08900**
adv_tou	0.08900	0.08500	0.08300	**0.08100**
aff_std	0.18933	0.18850	**0.18833**	0.18867
aff_tou	0.06800	0.06700	**0.06600**	0.06600
com_std	**0.10200**	0.10250	0.10225	0.10200
com_tou	0.12600	0.12600	0.12400	**0.12300**

Table 21: Average RMSLE for each group for the Multivariate model

	Number of Neurons			
Group	50	100	150	200
adv_std	0.10860	**0.10800**	0.10800	0.10860
adv_tou	0.08400	**0.08300**	0.08400	0.08500
aff_std	0.13333	**0.13233**	0.13267	0.13267
aff_tou	**0.07600**	0.09200	0.09100	0.09000
com_std	**0.05800**	0.05825	0.05900	0.06000
com_tou	**0.08700**	0.08800	0.08800	0.08800

4.2 Results

In this Section, we present our results. As mentioned in the Materials and Methods Section, we used RSMLE as our metric for comparison.

Before presenting those results, we thought it would be interesting to see how the model's predictions behaved from model to model and form group to group. For this, we constructed a graph showing the true value for the first 48 hours of our *test* group and compared these values with the predicted ones for each of the five models. We only did this for a lapse of 48 hours because it was the best time window to visualize clearly the difference without the graph getting too crowded. Because our predictions are also on individual household, we chose a random house per group to represent. For the Adversity-Std group we have household number 22 (*Figure 10*), for Adversity-Tou we have household number 31 (*Figure 11*), for Affluent-Std we chose household number 18 (*Figure 12*), for Affluent-Tou household number 15 (*Figure 13*), for Comfortable-Std household number 20 (*Figure 14*) and, finally, for Comfortable-Tou we have household number 44 (*Figure 15*). As our first peek into our results, we can see that our basic average model is flat compared to our true values and will probably perform poorly. ARIMA surprisingly does not perform much better than our basic model and this did come as a surprise. The fight will therefore be between LSTM models and we came to realise that our true benchmark will be our weekly average model. As for LSTM models, even though they are both very good, LSTM Univariate seems too good and therefore raises concern about overfitting; it also seems to underestimate the values together with the basic weekly average and, as we mentioned in our metric section (2.2.2), this is not desirable (the black line labelled 'Original' is representative of our true values).

Figure 10: Actual values versus predicted values for household 22 from Adversity-Std

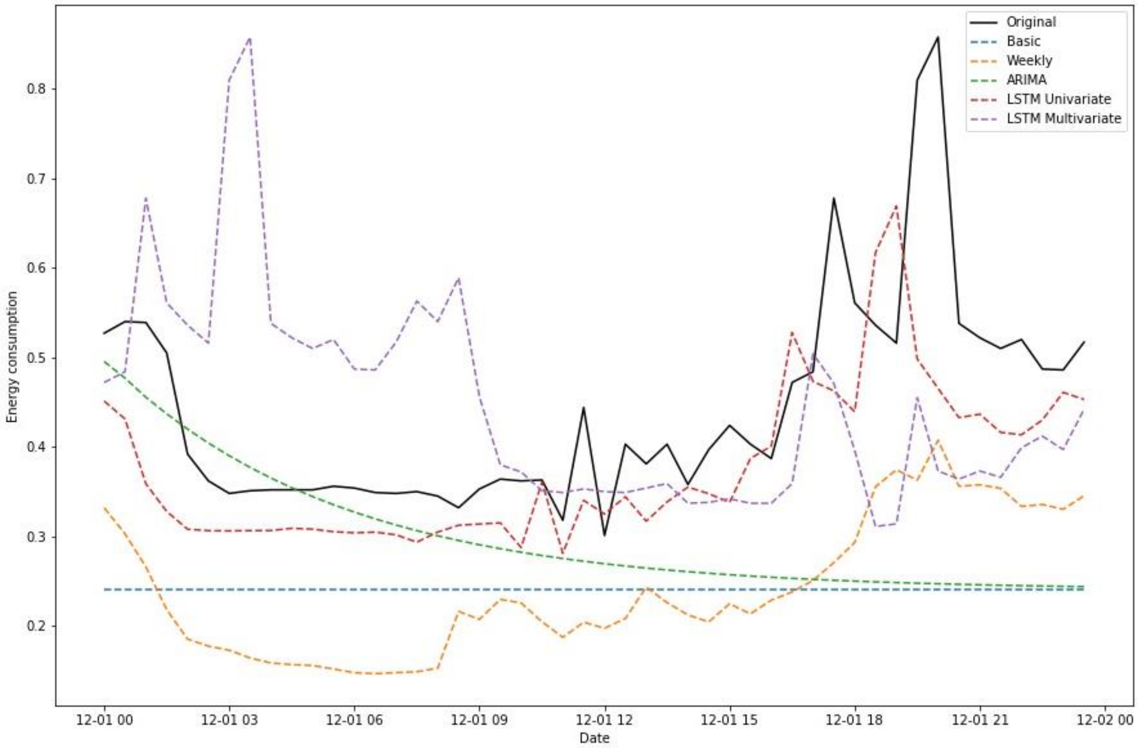


Figure 11: Actual values versus predicted values for household 31 from Adversity-Tou

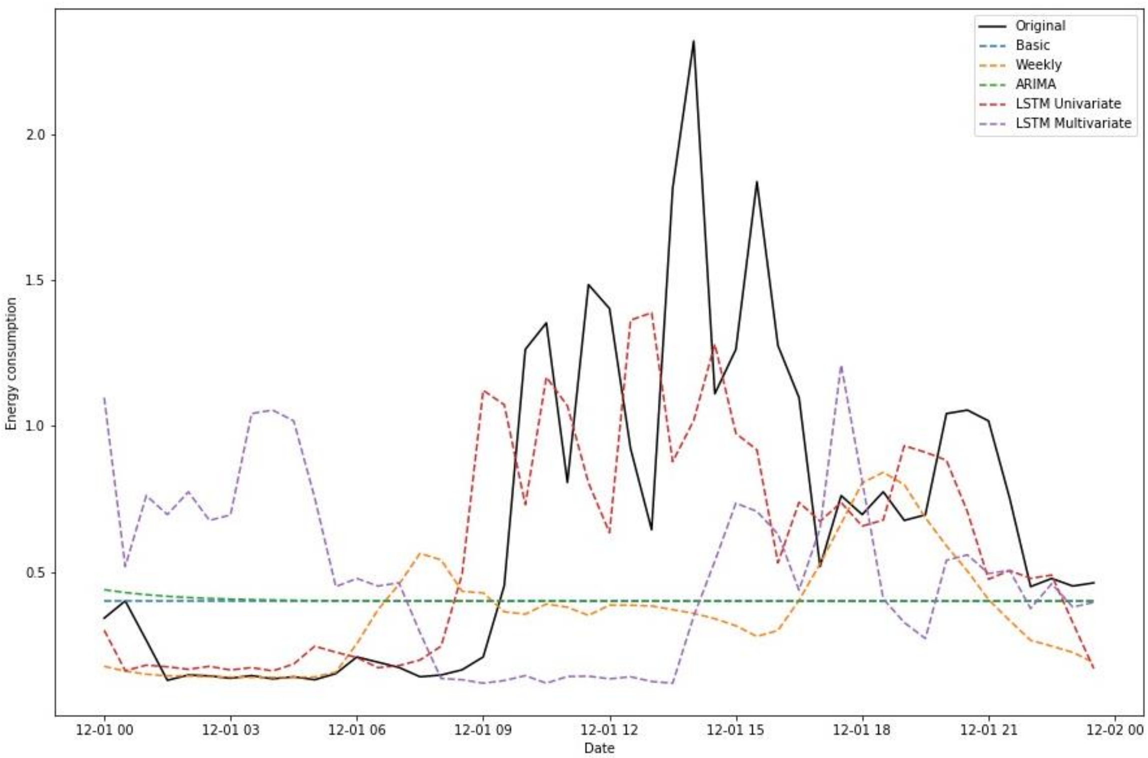


Figure 12: Actual values versus predicted values for household 18 from Affluent-Std

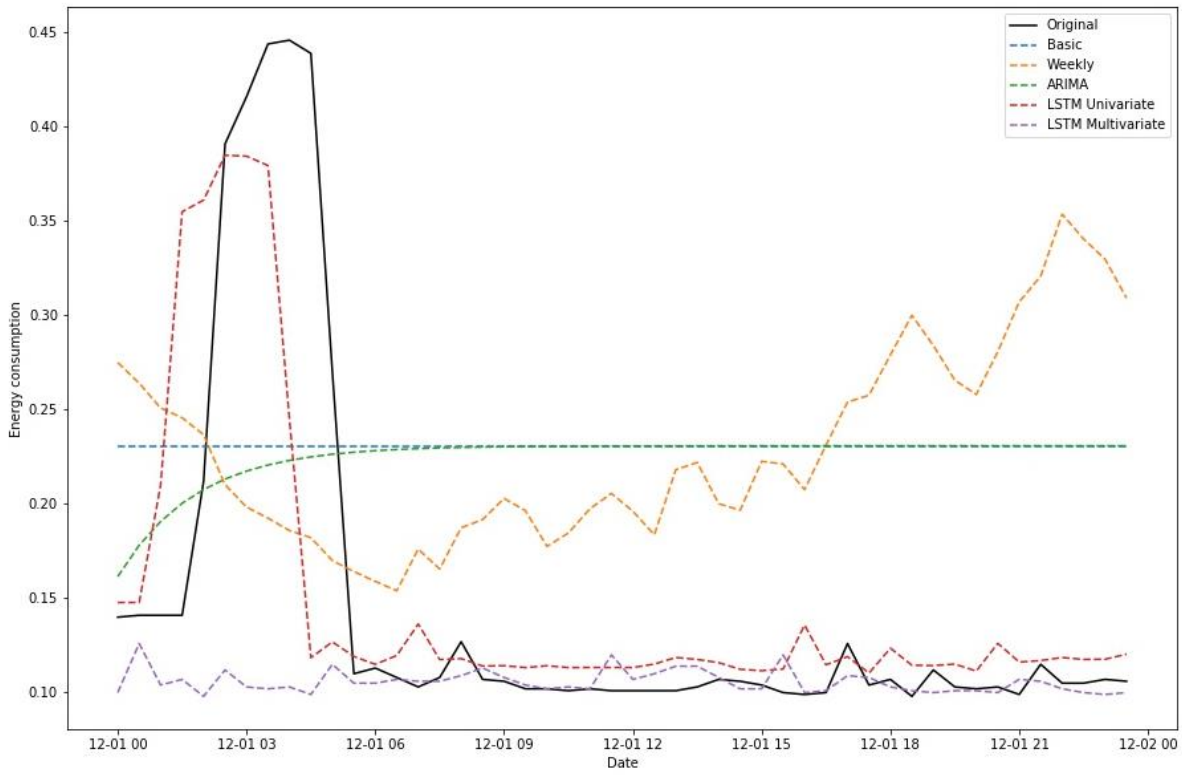


Figure 13: Actual values versus predicted values for household 15 from Affluent-Tou

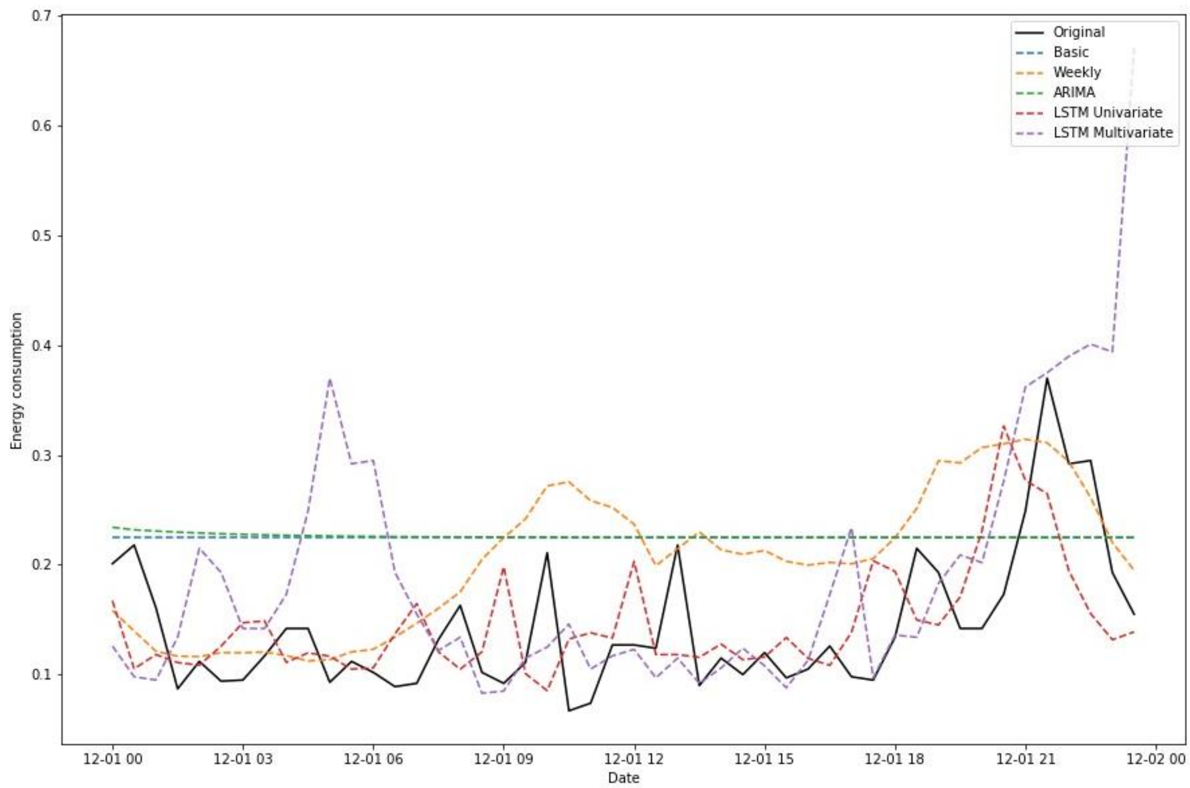


Figure 14: Actual values versus predicted values for household 20 from Comfortable-Std

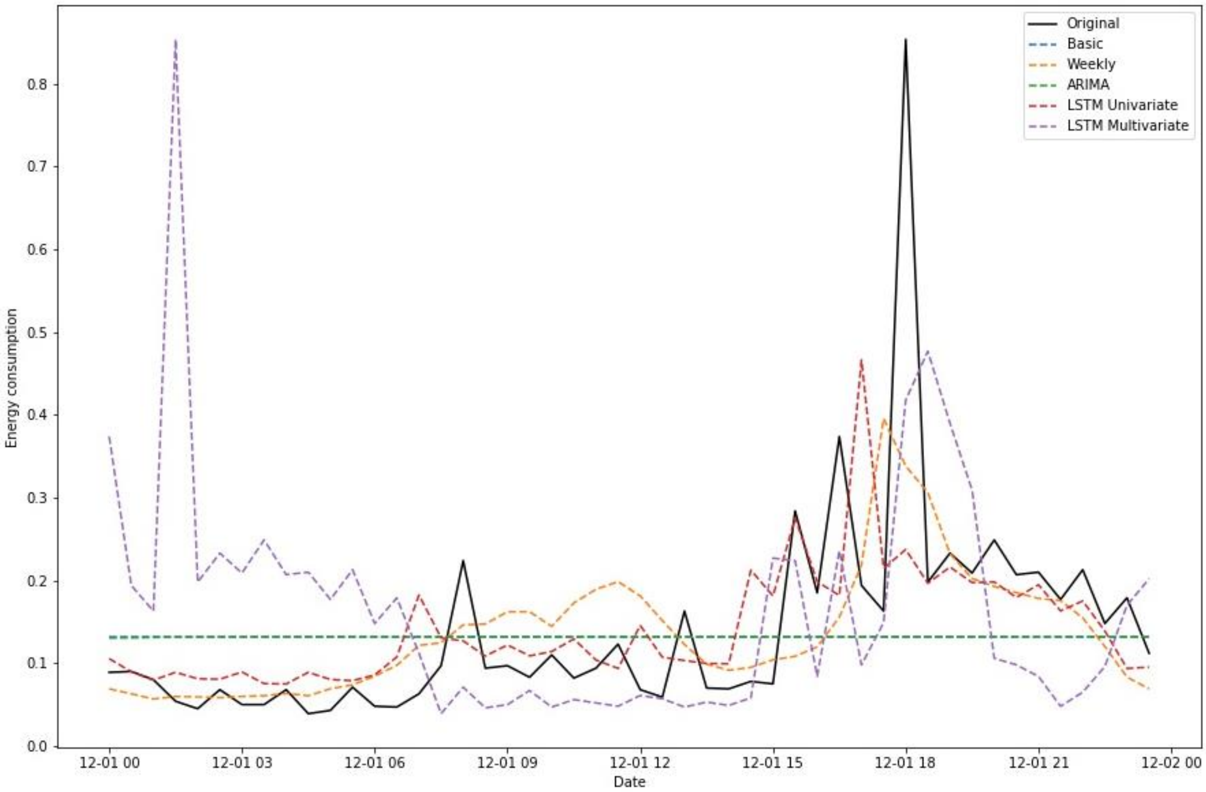


Figure 15: Actual values versus predicted values for household 44 from Comfortable-Tou

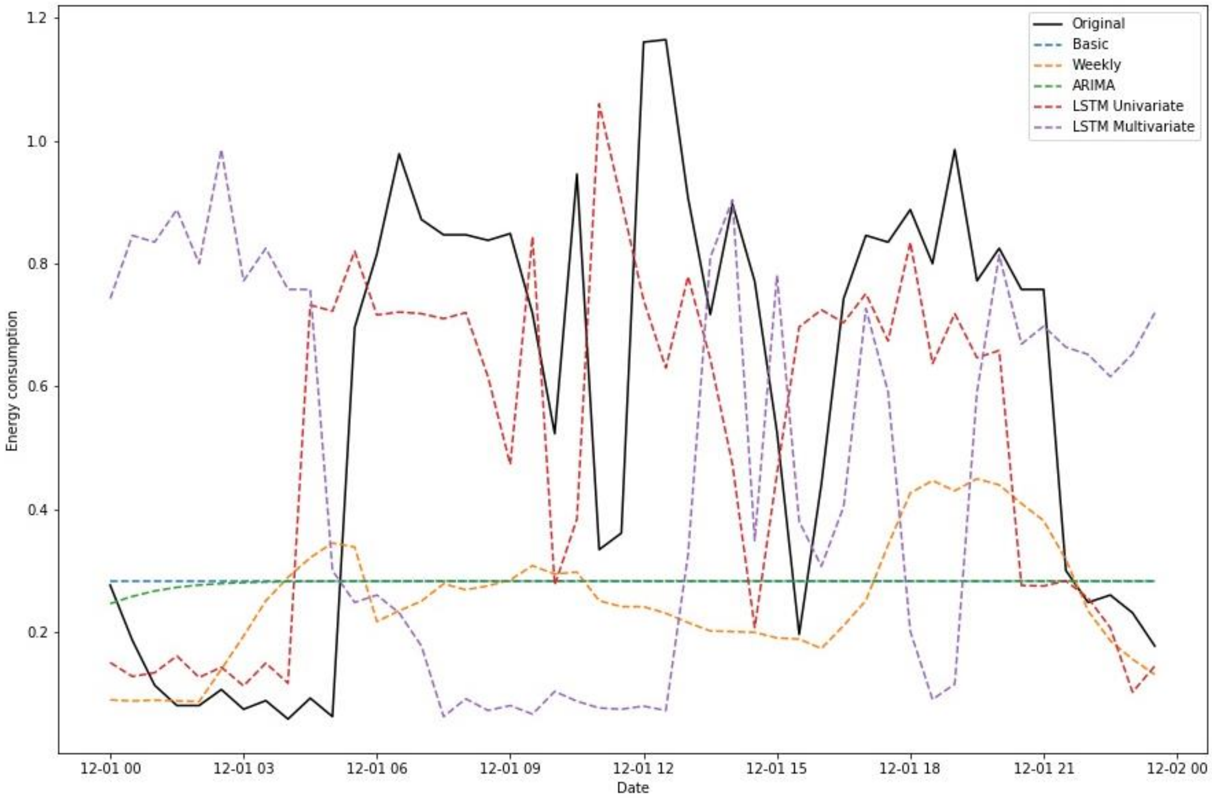
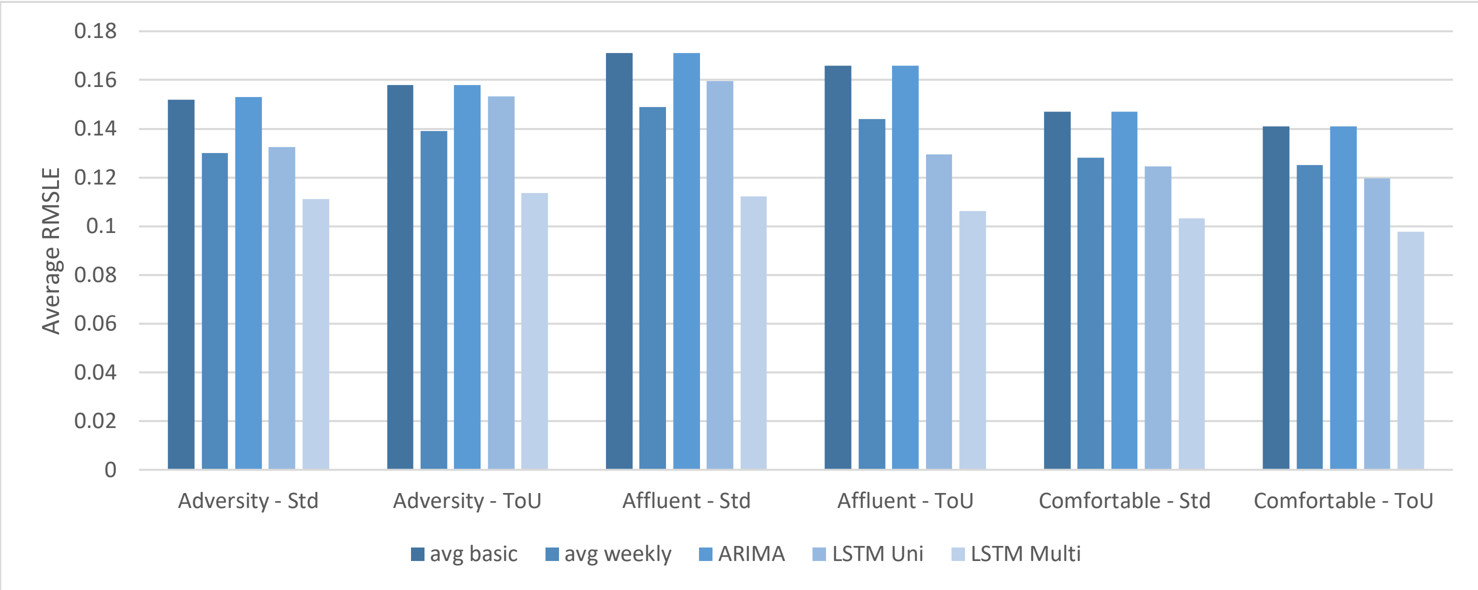


Figure *16* shows in a clear way how the different models performed for each group ACORN-Tariff according to RMSLE outcomes. Each column represents the average RMSLE for each model and each group of columns stands for each ACORN-Tariff grouping.

Figure 16: Average RMSLE result for each model and for each Acorn-Tariff group

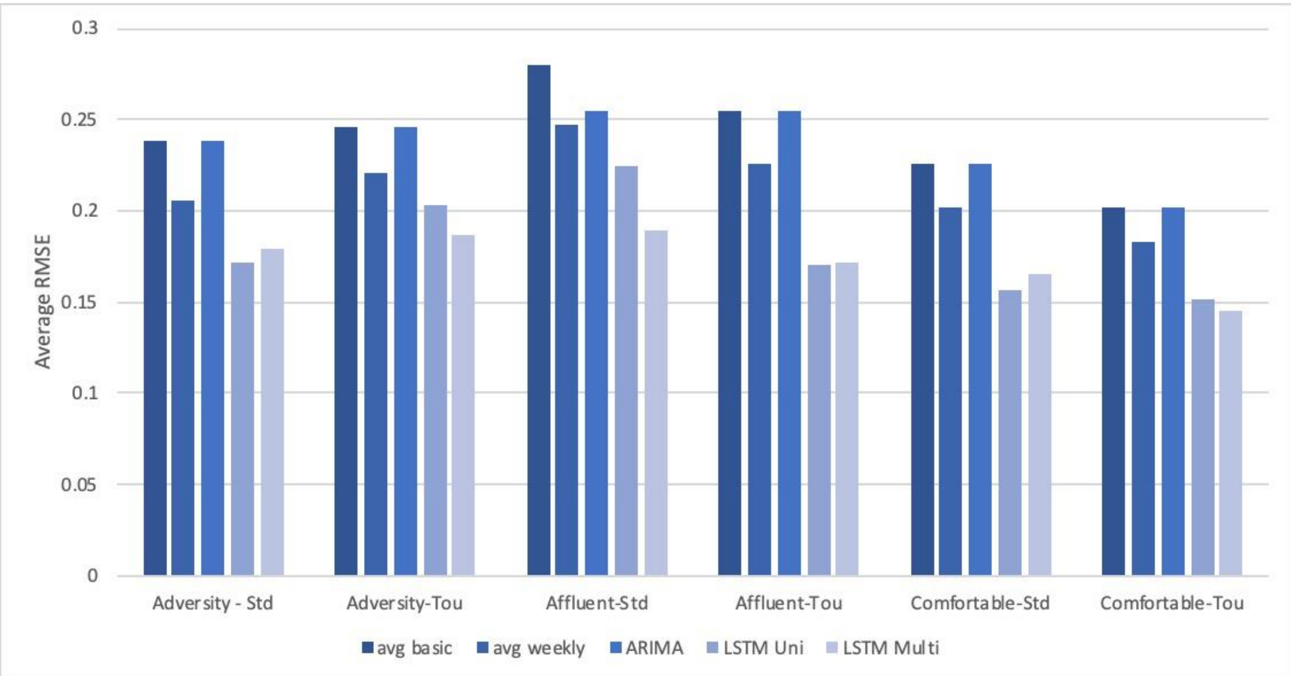


If we start by comparing our three models (no benchmarks), it is clear how LSTM, with the presence of covariates, outperforms the other two. Moreover, LSTM Univariate is still better than ARIMA for all cases. When comparing with our benchmark models, surprisingly, ARIMA performed on par with the benchmark Average Basic model for all cases as well. This came as a shock since ARIMA is the industry's standard for time series forecasting. Furthermore, and even more surprisingly, the benchmark Average weekly outperformed ARIMA in all cases. LSTM univariate was also somewhat disappointing since its performance lies somewhere between ARIMA and Average Weekly model.

To conclude, the best two models were both LSTM models and the Average Weekly model. However, LSTM Multivariate did the best performance throughout all groups.

As mentioned in section 2.2.2, where we proposed RMSLE as a metric, we also proposed RMSE as a counterpart to see how results would hold. In Figure 17 we have Figure 16's analogue where we show for all groups and models the average RMSE results. Surprisingly enough, results held the same compared to RMSLE for ARIMA and the average basic model (model 1 and 3) as they were still the worst performers. Model 2, the weekly average, did perform worst, or viewed form another standpoint, LSTM models performed much better – specially LSTM Univariate. LSTM Multivariate model still remained the best model. Even though these results are more in line with what we would have expected, if we recall the presented figures in the results section 4.2, LSTM Univariate did tend to underestimate the values. This for our particular context is not good because that would mean potentially not meeting the supply of energy for our customers. RMSLE, by penalizing the underestimation of the actual value more severely than it does for overestimation, gives LSTM Multivariate the advantage, and this is what we seek as a good behaviour from our predictive model. More detailed results for RMSE can be found on the annex.

Figure 17: Average RMSE result for each model and for each Acorn-Tariff group



5 Conclusion

5.1 Prescriptive Analysis

As we have seen throughout this work, data description can help energy companies better understand their customers. Customer segmentation like ACORN and the statistical evidence on their difference on consumption (and therefore behaviour) implies the possibility of differentiating their prices or even the services they are offered. As a direct consequence of this, revenues can increase, and the customer experience improved.

The advantages of predictive analysis like Load Forecasting is very clear. It enables the utility company to plan well by having an understanding of the future consumption, and the better the forecast, the better the planning. Planning may include decisions like investment on infrastructure or grid optimization. The obvious consequence of this is the possibility of better managing the risk involved in investments and the probability of shortage or surplus of energy. This last point entails the possibility of a better management of costs that are normally difficult to foresee.

From the standpoint of this work, where we treat the households on an individual scope, it may help improve the relationship with customers. For example, things like scheduling maintenance of the power systems can be done ensuring a minimum impact on the consumer by knowing when the energy consumption will be at its lowest.

The group that derives the greatest benefits from individual load forecasting is most probably the customers by offering them the possibility to access their information on consumption. They can plan ahead and manage their own consumption and therefore economics.

Furthermore, it was important to see that dynamic tariff does seem to statistically affect customer behaviour. This can have a great impact on consumption and planning for the companies. When companies visualize timestamps where consumption may lead to exceed their capacity of production, they can easily increase the energy prices. As soon as demand settles, prices can lower again.

5.2 Summary of our work

To conclude, the proposal for this work was to give data obtained from individual smart meters a clear business application.

The use of historical data on energy consumption in aggregated form has been used in the past and proved to be very beneficial. Decisions on strategy for investment and demand-supply equilibrium can be greatly improved by this. However, with the introduction of smart meters and fine grain data, we open the possibility of finding useful insights tailored to individual customers or segments.

When analysing individual households, we found that no household is the same. This meant that trying to adjust one model to fit them all would be challenging. By analyzing characteristics that could be common to some households like the ACORN group or type of tariffs imposed, we found that, if we grouped them accordingly, the difference within groups was lower but between groups was still statistically significant. Doing this allowed us to propose a predictive analysis that was robust while the parameters and hyperparameters remain constant within groups.

We proposed five different approaches on Load Forecasting, four models without the use of covariates and one model that included weather information.

The first two models, Average Basic and Average Weekly were presented as benchmarks for the remaining three: ARIMA, LSTM Univariate and LSTM Multivariate. The expected result was a clear dominance of LSTM Multivariate over all the others. This entails a very important conclusion on the use of covariates. It made clear that the use of other variables other than consumption itself can greatly improve predictions and the understanding of customer behaviour as a whole.

We chose RMSLE as our metric for comparing the models because it penalizes the underestimation of the actual value more severely than it does for overestimation. This last property is highly valuable for the energy companies as it helps avoid shortages.

The results were both surprising and reassuring. On the one hand, LSTM Multivariate was a clear winner amongst all options. On the other hand, ARIMA, the industry standard, performed poorly, on par with LSTM Univariate compared with our benchmark Average Weekly. Overall LSTM Multivariate proved to be very good with an average RSMLE of 0.108.

Having seen that households are unique but can still be grouped using common characteristics without losing their differentiation can be of advantage for targeted strategies such as investments or dynamic pricing.

Moreover, by visualizing their consumption behaviour, anomalies such as faulty smart meters (recall the Null entries we had to delete that were present on random timestamps) can be easily detected.

Lastly, consumers can become ‘smart’ by making use of their consumption data and consume in a much more conscious way, reducing their energy consumption if it is not necessary or changing their consumption behaviour with the adoption of dynamic pricing. This last point can help flatten the curve of demand-supply and avoid peaks of consumption on specific timestamps that could lead to insufficient supply and, therefore, revenue loss.

All in all, smart meters and high frequency data correctly analysed can yield enormous benefits both for the consumer and the companies providing the energy.

5.3 Limitations and Future work

For future work, given that we were expecting the ARIMA model to perform better, we would try and improve it. Here we tuned the ARIMA hyper parameters ourselves with PACF and ACF plots as well as the Dickey-Fuller test. However, it would be interesting to use an automated version that could be applied in a real business scenario. AutoARIMA was tried but the time it took to run seemed impractical for business applications.

Moreover, given the outstanding results that yielded from adding covariates to our models, it would be interesting to pursue more models that can contemplate covariates in a natural and easy way like Long Short-Term Memory Artificial Neural Networks do.

6 Bibliography

- Athanasopoulos, George; Hyndman, R. J. (2018). *Forecasting: Principles and Practice*.
- Chollet, F. (2018). Deep Learning with Phyton. In *Manning*. http://faculty.neu.edu.cn/yury/AAI/Textbook/Deep Learning with Python.pdf
- Edwards, R. E., New, J., & Parker, L. E. (2012). Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, *49*, 591–603. https://doi.org/10.1016/j.enbuild.2012.03.010
- EU. (2019). Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU. *Official Journal of the European Union*, *125*(L: 158). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2019:158:FULL&from=EN
- European Commission. (2014). *Cost-benefit analysis and state of play of smart metering deployment in the EU-27 Accompanying the document Report from the Commission Benchmarking smart metering deployment in the EU-27 with a focus on electricity*. 1–76. http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1403084595595&uri=SWD:2014:189:FIN
- Gajowniczek, K., & Zabkowski, T. (2014). Short term electricity forecasting using individual smart meter data. *Procedia Computer Science*, *35*(C), 589–597. https://doi.org/10.1016/j.procs.2014.08.140
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning - with Applications in R | Gareth James | Springer*. https://www.springer.com/gp/book/9781461471370%0Ahttp://www.springer.com/us/book/9781461471370

- Jason, B. (2018). *Deep Learning for Time Series Forecasting Predict the Future with MLPs, CNNs and LSTMs in Python*. 575.
- Koponen, P., & Rochas, C. (2008). *Definition of Smart Metering and Applications and Identification of Benefits Definition of Smart Metering and Applications and Identification of Benefits Deliverable D3 of the European Smart Metering Alliance ESMA , State of art . editor Pekka Koponen , V. January*.
- Li, X., Lu, R., Liang, X., Shen, X., Chen, J., & Lin, X. (2011). Smart community: An internet of things application. *IEEE Communications Magazine*, *49*(11), 68–75. https://doi.org/10.1109/MCOM.2011.6069711
- Luo, J., Hong, T., & Yue, M. (2018). Real-time anomaly detection for very short-term load forecasting. *Journal of Modern Power Systems and Clean Energy*, *6*(2), 235–243. https://doi.org/10.1007/s40565-017-0351-7
- Seymour, L., Brockwell, P. J., & Davis, R. A. (1997). Introduction to Time Series and Forecasting. In *Journal of the American Statistical Association* (Vol. 92, Issue 440). https://doi.org/10.2307/2965440
- Stanley, R. E., & Taraza, D. (2001). Bearing characteristic parameters to estimate the optimum counterweight mass of a 6-cylinder in-line engine. *American Society of Mechanical Engineers, Internal Combustion Engine Division (Publication) ICE*, *36*(3), 123–135. https://doi.org/10.1115/ices2001-140
- Tounquet, F., & Alaton, C. (2019). Benchmarking smart metering deployment in EU. In *European Commission* (Issue December). https://doi.org/10.2833/492070
- Ursula, P., Frans, E. V., Economy, C., & Plan, A. (2020). *The_European_Green_Deal_sets_out_how_to_make_Europe_the_first_climate-neutral_continent_by_2050__boosting_the_economy__improving_people_s_health_and_quality_of_life__caring_fo*. *March*, 11–12.

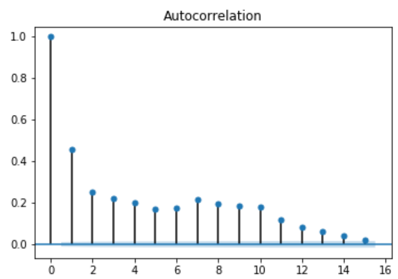
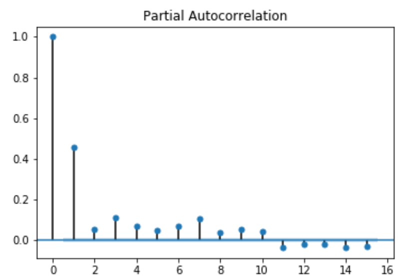
- Wang, Y., Chen, Q., Hong, T., & Kang, C. (2019). Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid*, *10*(3), 3125–3148. https://doi.org/10.1109/TSG.2018.2818167
- Zhong, S., & Tam, K. S. (2015). Hierarchical Classification of Load Profiles Based on Their Characteristic Attributes in Frequency Domain. *IEEE Transactions on Power Systems*, *30*(5), 2434–2441. https://doi.org/10.1109/TPWRS.2014.2362492
- Zhou, H., Yu, K. M., Lee, M. G., & Han, C. C. (2018). The Application of Last Observation Carried Forward Method for Missing Data Estimation in the Context of Industrial Wireless Sensor Networks. *Proceedings of the 2018 IEEE 7th Asia-Pacific Conference on Antennas and Propagation, APCAP 2018*, 130–131. https://doi.org/10.1109/APCAP.2018.8538147
- *acorn-consumer-classification-caci @ www.gov.uk*. (n.d.). https://www.gov.uk/government/publications/quality-assurance-of-administrative-data-in-the-uk-house-price-index/acorn-consumer-classification-caci
- *smartmeter-energy-use-data-in-london-households @ data.london.gov.uk*. (n.d.). https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households
- *index @ www.ukpowernetworks.co.uk*. (n.d.). https://www.ukpowernetworks.co.uk/

7 Annex

ARIMA model estimation for each group corresponding to Section ***Predictive Analysis / Setting Parameters and Hyperparameters.***

- Adversity – Std

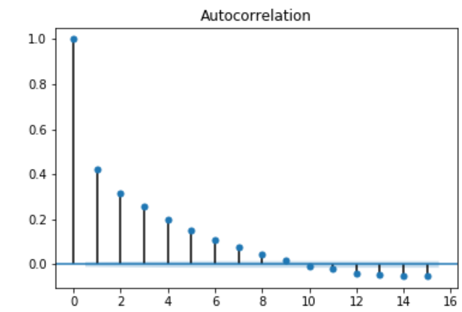
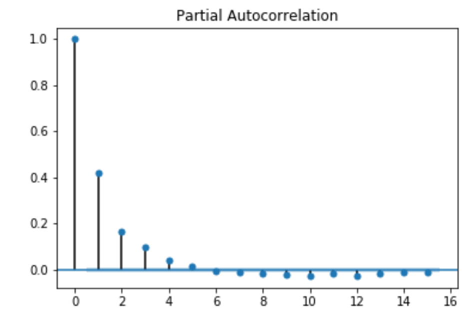
2) Household MAC000019

Dickey – Fuller Test	
• ADF Statistic: -8.740055 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

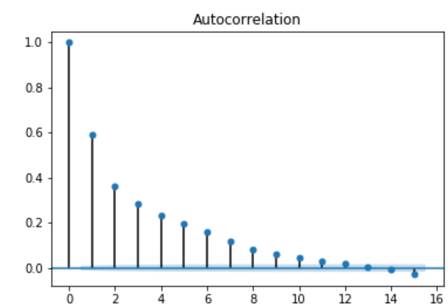
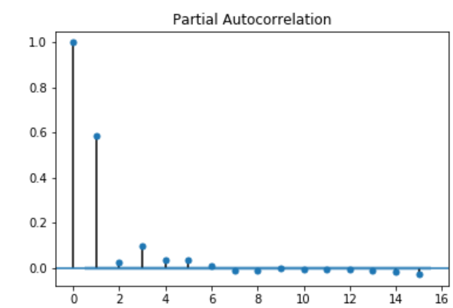
3) Household MAC000115

Dickey-Fuller Test	
• ADF Statistic: -13.548696 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

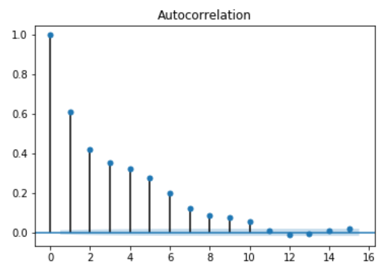
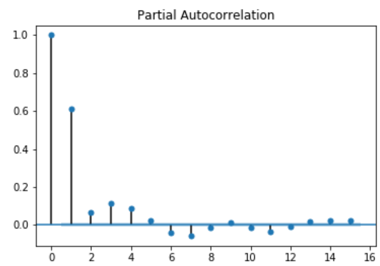
4) Household MAC000268

Dickey-Fuller Test	
• ADF Statistic: -17.782722 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

5) Household MAC004533

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">• ADF Statistic: -11.269950
• p-value: 0.000000
• Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

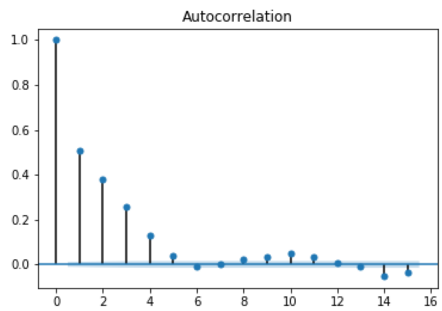
The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

6) Household MAC004866

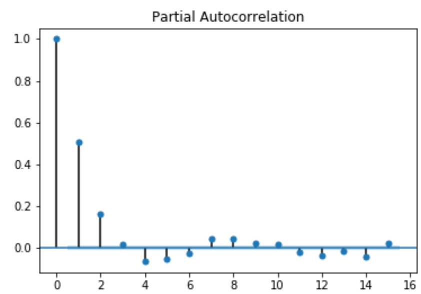
<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">

- ADF Statistic: -4.965481
- p-value: 0.000026
- Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567

</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>



</td><td>



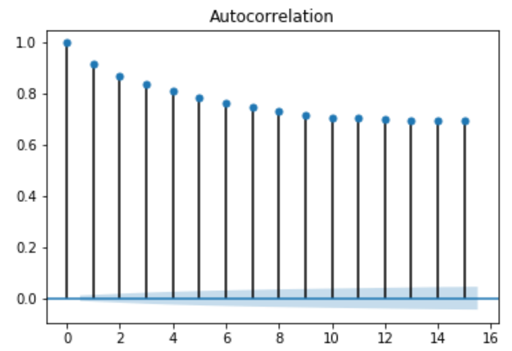
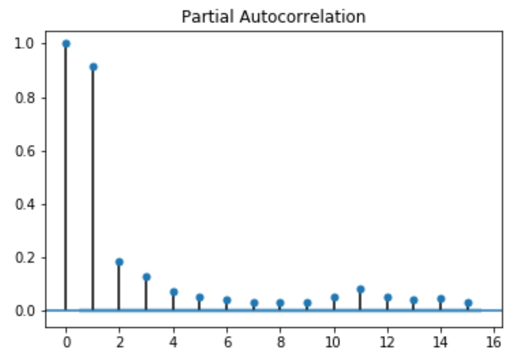
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

- Adversity – Tou

1) MAC000195

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">• ADF Statistic: -6.751407
• p-value: 0.000000
• Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

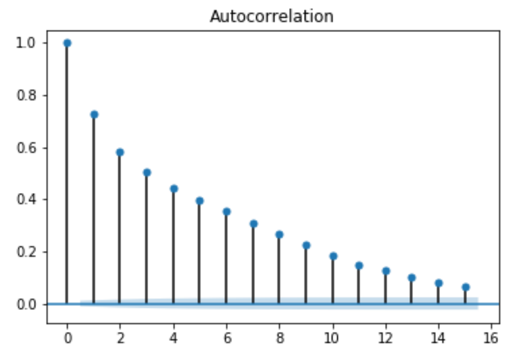
- Affluent – Std

1) MAC000030

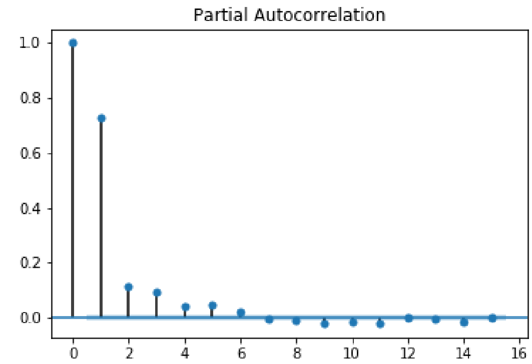
<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">

- ADF Statistic: -13.350816
- p-value: 0.000000
- Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567

</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>



</td><td>



</td></tr>
</table>

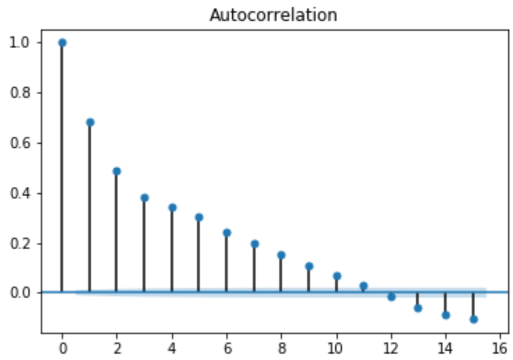
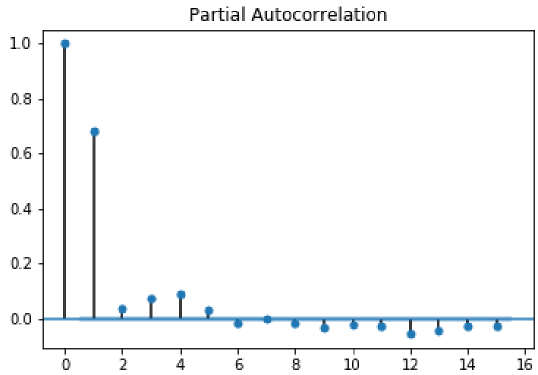
The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

1) MAC000110

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">

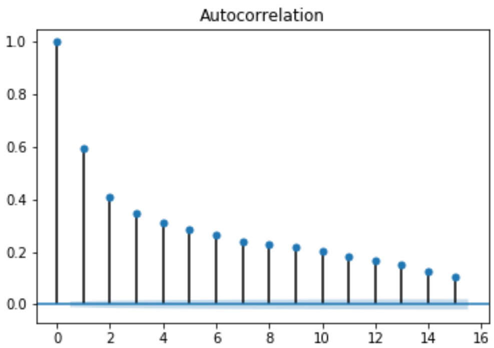
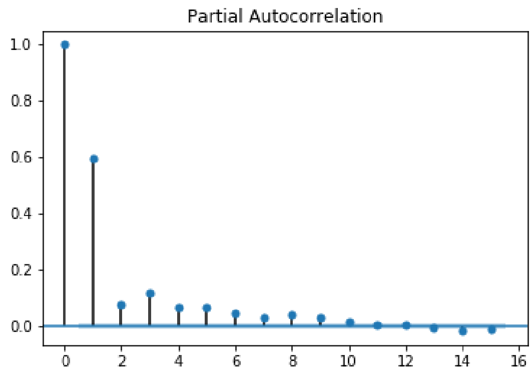
- ADF Statistic: -14.540008
- p-value: 0.000000
- Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567

</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

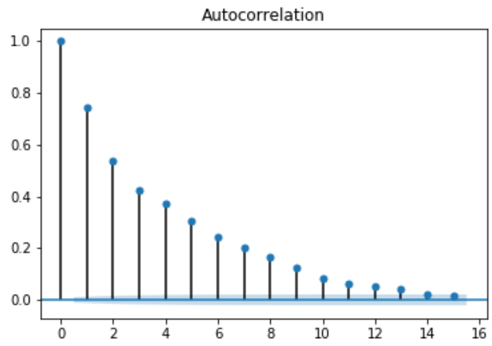
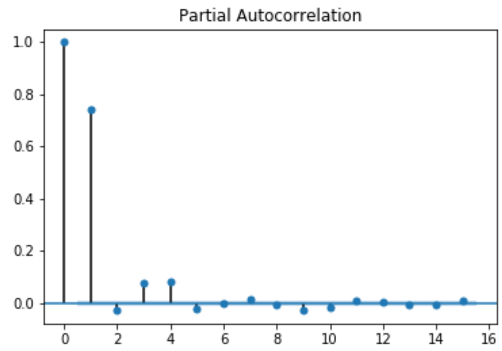
2) MAC000242

Dickey-Fuller Test	
• ADF Statistic: -9.425087 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

3) MAC004519

Dickey-Fuller Test	
• ADF Statistic: -12.912283 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

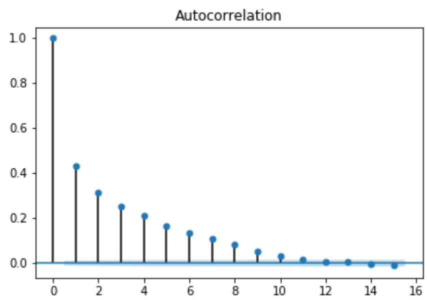
The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

4) MAC004555

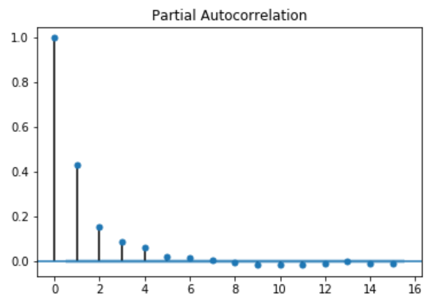
<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">

- ADF Statistic: -12.163935
- p-value: 0.000000
- Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567

</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>



</td><td>

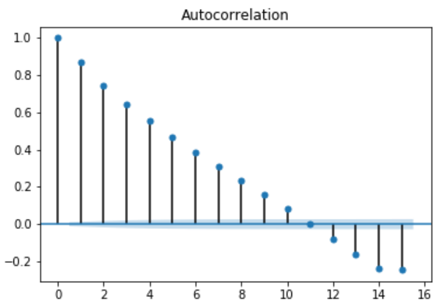
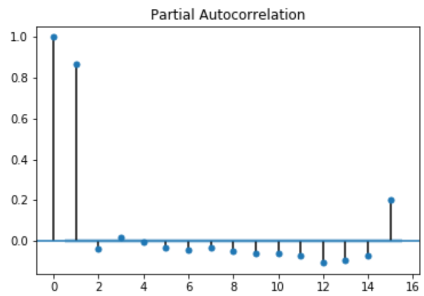


</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

5) MAC004863

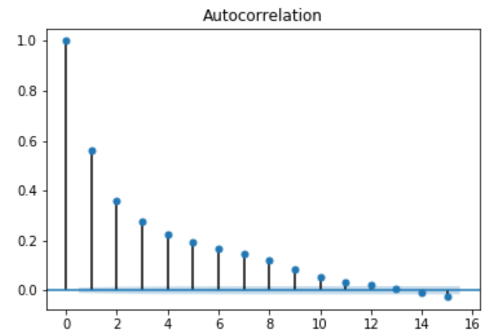
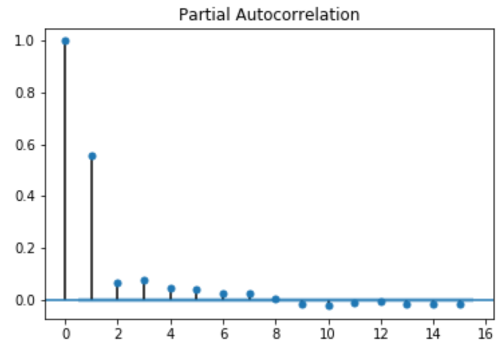
Dickey-Fuller Test	
• ADF Statistic: -8.126708 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

- Affluent – Tou

1) MAC000109

Dickey-Fuller Test	
• ADF Statistic: -15.162764 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

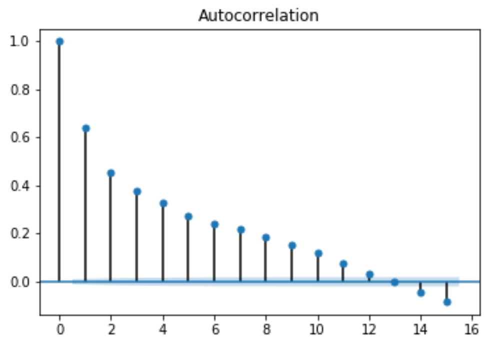
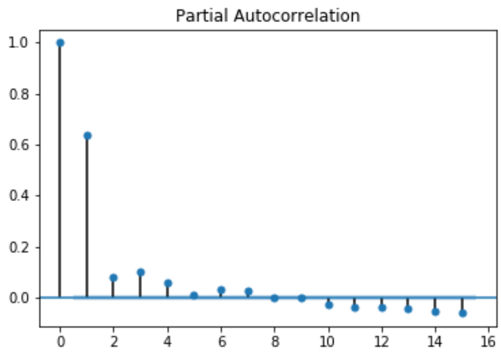
The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

- Comfortable - Std

1) MAC000059

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">

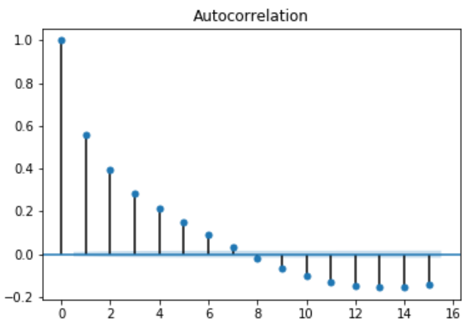
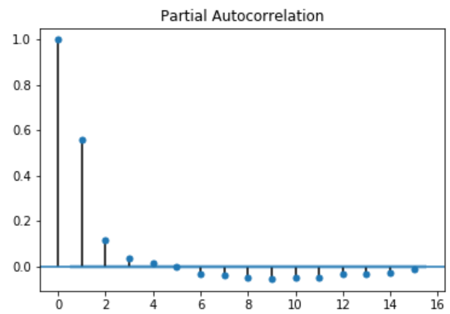
- ADF Statistic: -14.910114
- p-value: 0.000000
- Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567

</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

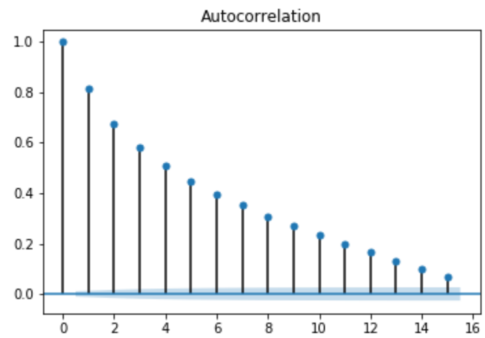
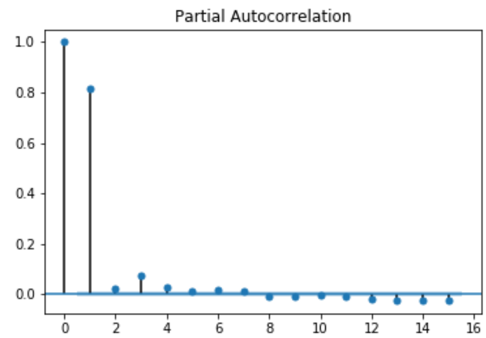
1) MAC000151

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">• ADF Statistic: -13.087662
• p-value: 0.000000
• Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the parameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

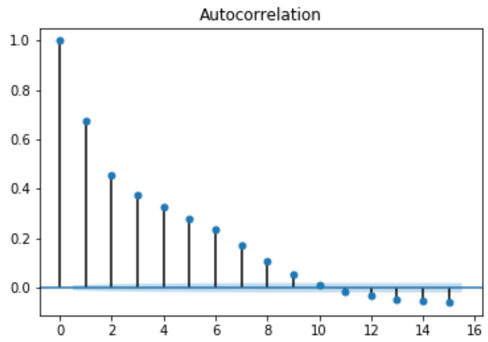
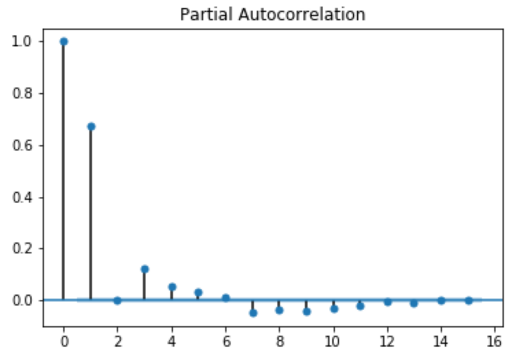
2) MAC000217

<table>
<tr><th colspan="2">Dickey-Fuller Test</th></tr>
<tr><td colspan="2">• ADF Statistic: -9.618152
• p-value: 0.000000
• Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567</td></tr>
<tr><th>ACF</th><th>PACF</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

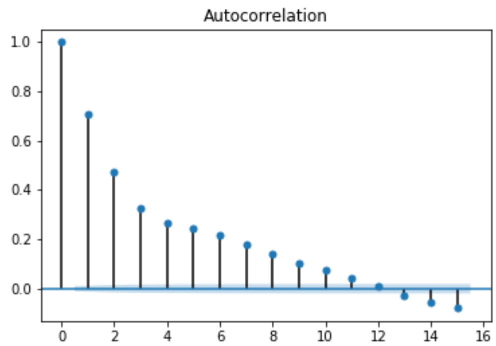
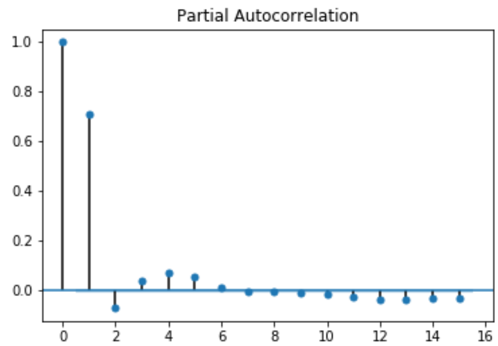
3) MAC004545

Dickey-Fuller Test	
• ADF Statistic: -18.522891 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA(2,0,0).

- Comfortable – Tou

Dickey-Fuller Test	
• ADF Statistic: -18.493115 • p-value: 0.000000 • Critical Values: 1%: -3.431 | 5%: -2.862 | 10%: -2.567	
ACF	**PACF**
	

The Dickey-Fuller test has a p-value of zero, and therefore with a confidence level of 1% we can say that the time series is stationary and so the hyperparameter d is equal to 0.

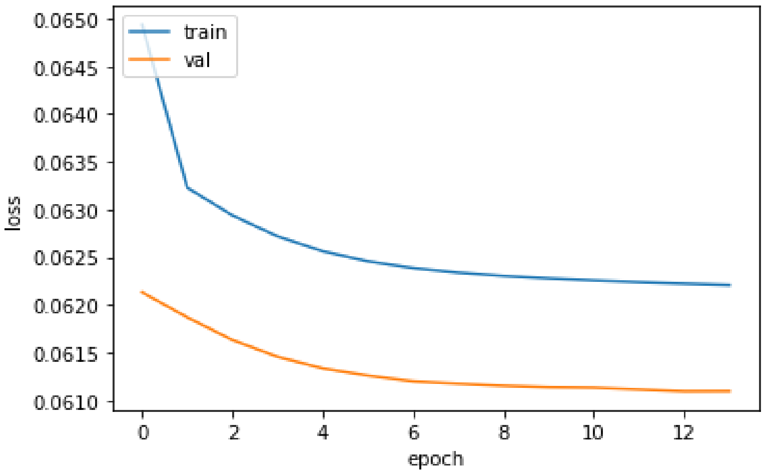
The ACF plot has a positive autocorrelation at lag 1 and a slow decay on ACF. We are therefore sure we need an AR model. PACF has two spikes and so we can conclude that this model is ARIMA (2,0,0).

Extractions from LSTM Univariate and LSTM Multivariate models. There are two from each group (Adversity-Std, Adversity-Tou, …) and therefore 12 in total – 6 for LSTM Univariate and 6 for LSTM Multivariate.

Extract 2: Household MAC000107 from Adversity-Tou group with LSTM Univariate Model

```
Epoch 1/15
32252/32252 [==============================] - 4s 126us/step - loss: 0.0649 - rmsle_loss: 0.0649 - val_loss: 0.0621 - val_rmsle_loss: 0.0606
Epoch 2/15
32252/32252 [==============================] - 3s 102us/step - loss: 0.0632 - rmsle_loss: 0.0632 - val_loss: 0.0619 - val_rmsle_loss: 0.0603
Epoch 3/15
32252/32252 [==============================] - 3s 103us/step - loss: 0.0629 - rmsle_loss: 0.0629 - val_loss: 0.0616 - val_rmsle_loss: 0.0601
Epoch 4/15
32252/32252 [==============================] - 3s 104us/step - loss: 0.0627 - rmsle_loss: 0.0627 - val_loss: 0.0615 - val_rmsle_loss: 0.0599
Epoch 5/15
32252/32252 [==============================] - 3s 105us/step - loss: 0.0626 - rmsle_loss: 0.0626 - val_loss: 0.0613 - val_rmsle_loss: 0.0598
Epoch 6/15
32252/32252 [==============================] - 3s 99us/step - loss: 0.0625 - rmsle_loss: 0.0625 - val_loss: 0.0613 - val_rmsle_loss: 0.0597
Epoch 7/15
32252/32252 [==============================] - 3s 94us/step - loss: 0.0624 - rmsle_loss: 0.0624 - val_loss: 0.0612 - val_rmsle_loss: 0.0596
Epoch 8/15
32252/32252 [==============================] - 3s 97us/step - loss: 0.0623 - rmsle_loss: 0.0623 - val_loss: 0.0612 - val_rmsle_loss: 0.0596
Epoch 9/15
32252/32252 [==============================] - 3s 94us/step - loss: 0.0623 - rmsle_loss: 0.0623 - val_loss: 0.0612 - val_rmsle_loss: 0.0596
```

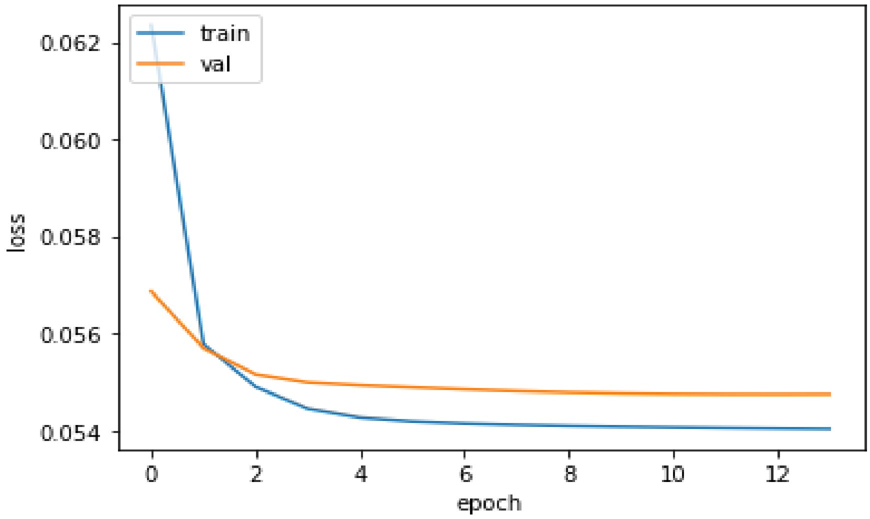
```
Epoch 10/15
32252/32252 [==============================] - 3s 96us/step - loss: 0.0623 - rmsle_loss: 0.0623 - val_loss: 0.0611 - val_rmsle_loss: 0.0596
Epoch 11/15
32252/32252 [==============================] - 3s 94us/step - loss: 0.0623 - rmsle_loss: 0.0623 - val_loss: 0.0611 - val_rmsle_loss: 0.0596
Epoch 12/15
32252/32252 [==============================] - 3s 92us/step - loss: 0.0622 - rmsle_loss: 0.0622 - val_loss: 0.0611 - val_rmsle_loss: 0.0596
Epoch 13/15
32252/32252 [==============================] - 3s 93us/step - loss: 0.0622 - rmsle_loss: 0.0622 - val_loss: 0.0611 - val_rmsle_loss: 0.0595
Epoch 14/15
32252/32252 [==============================] - 3s 94us/step - loss: 0.0622 - rmsle_loss: 0.0622 - val_loss: 0.0611 - val_rmsle_loss: 0.0595
Epoch 00014: early stopping
```



Extract 3: Household MAC000253 from Affluent-Std group with LSTM Univariate Model

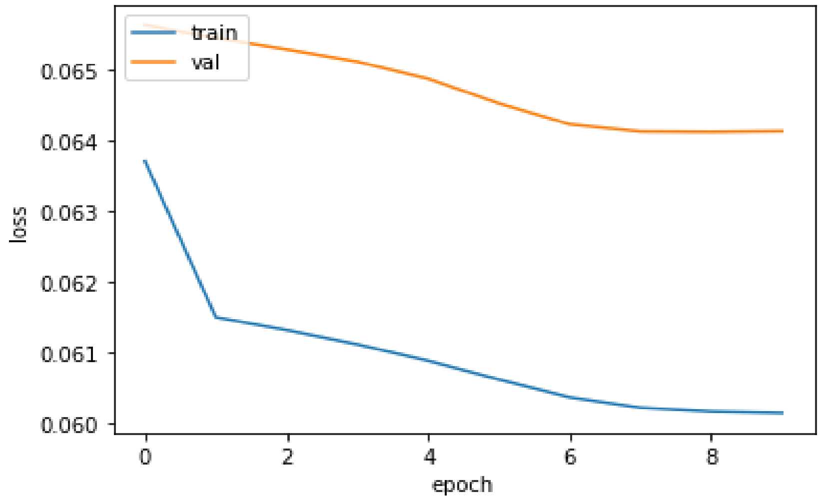
```
Epoch 1/15
32252/32252 [==============================] - 3s 104us/step - loss: 0.0623 - rmsle_loss: 0.0624 - val_loss: 0.0569 - val_rmsle_loss: 0.0562
Epoch 2/15
32252/32252 [==============================] - 2s 77us/step - loss: 0.0558 - rmsle_loss: 0.0558 - val_loss: 0.0557 - val_rmsle_loss: 0.0549
Epoch 3/15
32252/32252 [==============================] - 3s 84us/step - loss: 0.0549 - rmsle_loss: 0.0549 - val_loss: 0.0552 - val_rmsle_loss: 0.0544
Epoch 4/15
32252/32252 [==============================] - 3s 80us/step - loss: 0.0544 - rmsle_loss: 0.0545 - val_loss: 0.0550 - val_rmsle_loss: 0.0542
Epoch 5/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0543 - rmsle_loss: 0.0543 - val_loss: 0.0549 - val_rmsle_loss: 0.0542
Epoch 6/15
32252/32252 [==============================] - 3s 93us/step - loss: 0.0542 - rmsle_loss: 0.0542 - val_loss: 0.0549 - val_rmsle_loss: 0.0541
Epoch 7/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0541 - rmsle_loss: 0.0542 - val_loss: 0.0548 - val_rmsle_loss: 0.0541
Epoch 8/15
32252/32252 [==============================] - 3s 78us/step - loss: 0.0541 - rmsle_loss: 0.0541 - val_loss: 0.0548 - val_rmsle_loss: 0.0540
Epoch 9/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0541 - rmsle_loss: 0.0541 - val_loss: 0.0548 - val_rmsle_loss: 0.0540
Epoch 10/15
32252/32252 [==============================] - 3s 82us/step - loss: 0.0541 - rmsle_loss: 0.0541 - val_loss: 0.0548 - val_rmsle_loss: 0.0540
Epoch 11/15
32252/32252 [==============================] - 2s 76us/step - loss: 0.0541 - rmsle_loss: 0.0541 - val_loss: 0.0548 - val_rmsle_loss: 0.0540
Epoch 12/15
32252/32252 [==============================] - 3s 78us/step - loss: 0.0541 - rmsle_loss: 0.0541 - val_loss: 0.0547 - val_rmsle_loss: 0.0540
Epoch 13/15
```

```
32252/32252 [==============================] - 3s 80us/step - loss: 0.0540 - rmsle_loss: 0.0541 - val_loss: 0.0547 - val_rmsle_loss: 0.0540
Epoch 14/15
32252/32252 [==============================] - 3s 86us/step - loss: 0.0540 - rmsle_loss: 0.0540 - val_loss: 0.0547 - val_rmsle_loss: 0.0540
Epoch 00014: early stopping
```



Extract 4: Household MAC000173 from Affluent-Tou group with LSTM Univariate Model

```
Epoch 1/15
32252/32252 [==============================] - 4s 112us/step - loss: 0.0637 - rmsle_loss: 0.0637 - val_loss: 0.0656 - val_rmsle_loss: 0.0657
Epoch 2/15
32252/32252 [==============================] - 3s 85us/step - loss: 0.0615 - rmsle_loss: 0.0615 - val_loss: 0.0655 - val_rmsle_loss: 0.0655
Epoch 3/15
32252/32252 [==============================] - 3s 84us/step - loss: 0.0613 - rmsle_loss: 0.0613 - val_loss: 0.0653 - val_rmsle_loss: 0.0654
Epoch 4/15
32252/32252 [==============================] - 3s 88us/step - loss: 0.0611 - rmsle_loss: 0.0611 - val_loss: 0.0651 - val_rmsle_loss: 0.0652
Epoch 5/15
32252/32252 [==============================] - 3s 87us/step - loss: 0.0609 - rmsle_loss: 0.0609 - val_loss: 0.0649 - val_rmsle_loss: 0.0650
Epoch 6/15
32252/32252 [==============================] - 3s 84us/step - loss: 0.0606 - rmsle_loss: 0.0606 - val_loss: 0.0645 - val_rmsle_loss: 0.0646
Epoch 7/15
32252/32252 [==============================] - 3s 85us/step - loss: 0.0604 - rmsle_loss: 0.0604 - val_loss: 0.0642 - val_rmsle_loss: 0.0643
Epoch 8/15
32252/32252 [==============================] - 3s 87us/step - loss: 0.0602 - rmsle_loss: 0.0602 - val_loss: 0.0641 - val_rmsle_loss: 0.0642
Epoch 9/15
32252/32252 [==============================] - 3s 89us/step - loss: 0.0602 - rmsle_loss: 0.0602 - val_loss: 0.0641 - val_rmsle_loss: 0.0642
Epoch 10/15
32252/32252 [==============================] - 3s 89us/step - loss: 0.0601 - rmsle_loss: 0.0601 - val_loss: 0.0641 - val_rmsle_loss: 0.0642
Epoch 00010: early stopping
```



train
val
0.065
0.064
0.063
0.062
0.061
0.060
loss
0
2
4
6
8
epoch

Extract 5: Household MAC004478 from Comfortable-Std group with LSTM Univariate Model

```
Epoch 1/15
32252/32252 [==============================] - 3s 81us/step - loss: 0.0782 - rmsle_loss: 0.0782 - val_loss: 0.0717 -
val_rmsle_loss: 0.0704
Epoch 2/15
32252/32252 [==============================] - 1s 39us/step - loss: 0.0684 - rmsle_loss: 0.0684 - val_loss: 0.0689 -
val_rmsle_loss: 0.0676
Epoch 3/15
32252/32252 [==============================] - 1s 39us/step - loss: 0.0665 - rmsle_loss: 0.0665 - val_loss: 0.0666 -
val_rmsle_loss: 0.0653
Epoch 4/15
32252/32252 [==============================] - 1s 39us/step - loss: 0.0654 - rmsle_loss: 0.0654 - val_loss: 0.0656 -
val_rmsle_loss: 0.0644
Epoch 5/15
32252/32252 [==============================] - 1s 41us/step - loss: 0.0650 - rmsle_loss: 0.0651 - val_loss: 0.0651 -
val_rmsle_loss: 0.0639
Epoch 6/15
32252/32252 [==============================] - 2s 55us/step - loss: 0.0648 - rmsle_loss: 0.0648 - val_loss: 0.0648 -
val_rmsle_loss: 0.0636
Epoch 7/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0646 - rmsle_loss: 0.0646 - val_loss: 0.0644 -
val_rmsle_loss: 0.0632
Epoch 8/15
32252/32252 [==============================] - 3s 91us/step - loss: 0.0644 - rmsle_loss: 0.0644 - val_loss: 0.0642 -
val_rmsle_loss: 0.0630
Epoch 9/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0643 - rmsle_loss: 0.0643 - val_loss: 0.0639 -
val_rmsle_loss: 0.0627
Epoch 10/15
32252/32252 [==============================] - 2s 59us/step - loss: 0.0642 - rmsle_loss: 0.0642 - val_loss: 0.0637 -
val_rmsle_loss: 0.0626
Epoch 11/15
32252/32252 [==============================] - 3s 79us/step - loss: 0.0641 - rmsle_loss: 0.0641 - val_loss: 0.0636 -
val_rmsle_loss: 0.0624
Epoch 12/15
32252/32252 [==============================] - 2s 67us/step - loss: 0.0640 - rmsle_loss: 0.0641 - val_loss: 0.0635 -
val_rmsle_loss: 0.0623
Epoch 13/15
```

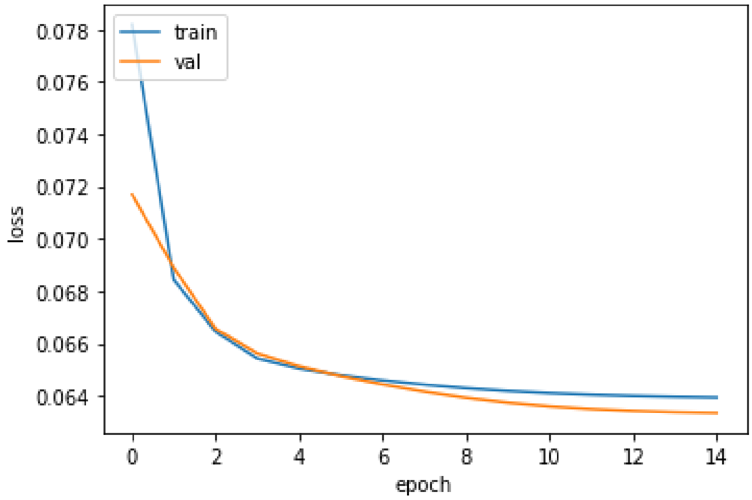
32252/32252 [==============================] - 2s 61us/step - loss: 0.0640 - rmsle_loss: 0.0640 - val_loss: 0.0634 - val_rmsle_loss: 0.0622

Epoch 14/15

32252/32252 [==============================] - 2s 75us/step - loss: 0.0640 - rmsle_loss: 0.0640 - val_loss: 0.0634 - val_rmsle_loss: 0.0622

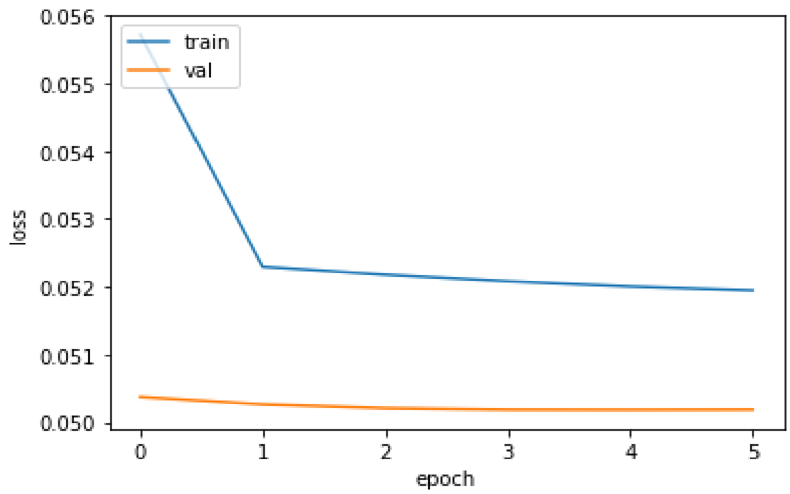
Epoch 15/15

32252/32252 [==============================] - 1s 41us/step - loss: 0.0639 - rmsle_loss: 0.0640 - val_loss: 0.0633 - val_rmsle_loss: 0.0622



Extract 6: Household MAC004485 from Comfortable-Tou group with LSTM Univariate Model

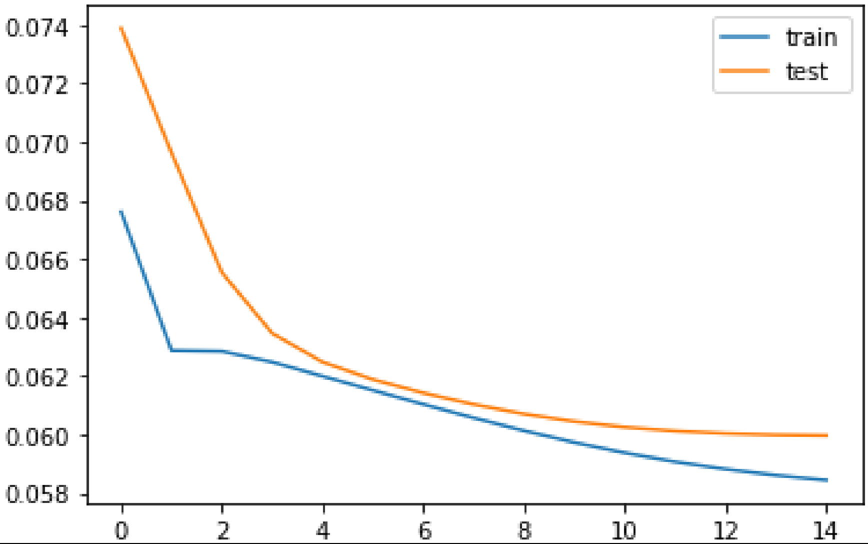
```
Epoch 1/15
32252/32252 [==============================] - 6s 184us/step - loss: 0.0557 - rmsle_loss: 0.0557 - val_loss: 0.0504 - val_rmsle_loss: 0.0499
Epoch 2/15
32252/32252 [==============================] - 5s 146us/step - loss: 0.0523 - rmsle_loss: 0.0523 - val_loss: 0.0503 - val_rmsle_loss: 0.0498
Epoch 3/15
32252/32252 [==============================] - 4s 132us/step - loss: 0.0522 - rmsle_loss: 0.0522 - val_loss: 0.0502 - val_rmsle_loss: 0.0498
Epoch 4/15
32252/32252 [==============================] - 4s 117us/step - loss: 0.0521 - rmsle_loss: 0.0521 - val_loss: 0.0502 - val_rmsle_loss: 0.0497
Epoch 5/15
32252/32252 [==============================] - 4s 121us/step - loss: 0.0520 - rmsle_loss: 0.0520 - val_loss: 0.0502 - val_rmsle_loss: 0.0497
Epoch 6/15
32252/32252 [==============================] - 4s 116us/step - loss: 0.0519 - rmsle_loss: 0.0519 - val_loss: 0.0502 - val_rmsle_loss: 0.0497
Epoch 00006: early stopping
```



Extract 7: Household from Adversity-Std group with LSTM Multivariate Model

```
Epoch 1/15
32256/32256 [==============================] - 2s 64us/step - loss: 0.0676 - rmsle_loss: 0.0676 - val_loss: 0.0739 - val_rmsle_loss: 0.0771
Epoch 2/15
32256/32256 [==============================] - 2s 56us/step - loss: 0.0629 - rmsle_loss: 0.0629 - val_loss: 0.0696 - val_rmsle_loss: 0.0729
Epoch 3/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0628 - rmsle_loss: 0.0629 - val_loss: 0.0655 - val_rmsle_loss: 0.0692
Epoch 4/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0625 - rmsle_loss: 0.0625 - val_loss: 0.0635 - val_rmsle_loss: 0.0674
Epoch 5/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0620 - rmsle_loss: 0.0620 - val_loss: 0.0625 - val_rmsle_loss: 0.0665
Epoch 6/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0615 - rmsle_loss: 0.0615 - val_loss: 0.0619 - val_rmsle_loss: 0.0660
Epoch 7/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0610 - rmsle_loss: 0.0610 - val_loss: 0.0614 - val_rmsle_loss: 0.0656
Epoch 8/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0606 - rmsle_loss: 0.0606 - val_loss: 0.0610 - val_rmsle_loss: 0.0653
Epoch 9/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0601 - rmsle_loss: 0.0601 - val_loss: 0.0607 - val_rmsle_loss: 0.0650
Epoch 10/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0597 - rmsle_loss: 0.0597 - val_loss: 0.0604 - val_rmsle_loss: 0.0648
Epoch 11/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0594 - rmsle_loss: 0.0594 - val_loss: 0.0602 - val_rmsle_loss: 0.0646
Epoch 12/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0591 - rmsle_loss: 0.0591 - val_loss: 0.0601 - val_rmsle_loss: 0.0645
Epoch 13/15
```

```
32256/32256 [==============================] - 1s 45us/step - loss: 0.0588 - rmsle_loss: 0.0588 - val_loss: 0.0600 - val_rmsle_loss: 0.0645
Epoch 14/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0586 - rmsle_loss: 0.0586 - val_loss: 0.0600 - val_rmsle_loss: 0.0645
Epoch 15/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0584 - rmsle_loss: 0.0585 - val_loss: 0.0600 - val_rmsle_loss: 0.0645
```



Extract 8: Household from Adversity-Tou group with LSTM Multivariate Model

```
Epoch 1/15
32256/32256 [==============================] - 2s 61us/step - loss: 0.0462 - rmsle_loss: 0.0462 - val_loss: 0.0489 - val_rmsle_loss: 0.0479
Epoch 2/15
32256/32256 [==============================] - 1s 44us/step - loss: 0.0450 - rmsle_loss: 0.0450 - val_loss: 0.0478 - val_rmsle_loss: 0.0467
Epoch 3/15
32256/32256 [==============================] - 1s 43us/step - loss: 0.0446 - rmsle_loss: 0.0446 - val_loss: 0.0469 - val_rmsle_loss: 0.0458
Epoch 4/15
32256/32256 [==============================] - 1s 43us/step - loss: 0.0443 - rmsle_loss: 0.0443 - val_loss: 0.0463 - val_rmsle_loss: 0.0452
Epoch 5/15
32256/32256 [==============================] - 1s 43us/step - loss: 0.0440 - rmsle_loss: 0.0440 - val_loss: 0.0459 - val_rmsle_loss: 0.0447
Epoch 6/15
32256/32256 [==============================] - 1s 44us/step - loss: 0.0437 - rmsle_loss: 0.0437 - val_loss: 0.0455 - val_rmsle_loss: 0.0443
Epoch 7/15
32256/32256 [==============================] - 3s 85us/step - loss: 0.0434 - rmsle_loss: 0.0434 - val_loss: 0.0451 - val_rmsle_loss: 0.0439
Epoch 8/15
32256/32256 [==============================] - 2s 70us/step - loss: 0.0431 - rmsle_loss: 0.0431 - val_loss: 0.0448 - val_rmsle_loss: 0.0435
Epoch 9/15
32256/32256 [==============================] - 2s 67us/step - loss: 0.0427 - rmsle_loss: 0.0428 - val_loss: 0.0445 - val_rmsle_loss: 0.0432
Epoch 10/15
32256/32256 [==============================] - 2s 48us/step - loss: 0.0425 - rmsle_loss: 0.0425 - val_loss: 0.0443 - val_rmsle_loss: 0.0431
Epoch 11/15
32256/32256 [==============================] - 2s 48us/step - loss: 0.0423 - rmsle_loss: 0.0423 - val_loss: 0.0442 - val_rmsle_loss: 0.0430
Epoch 12/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0422 - rmsle_loss: 0.0422 - val_loss: 0.0442 - val_rmsle_loss: 0.0429
Epoch 13/15
```

32256/32256 [==============================] - 2s 51us/step - loss: 0.0421 - rmsle_loss: 0.0421 - val_loss: 0.0441 - val_rmsle_loss: 0.0429

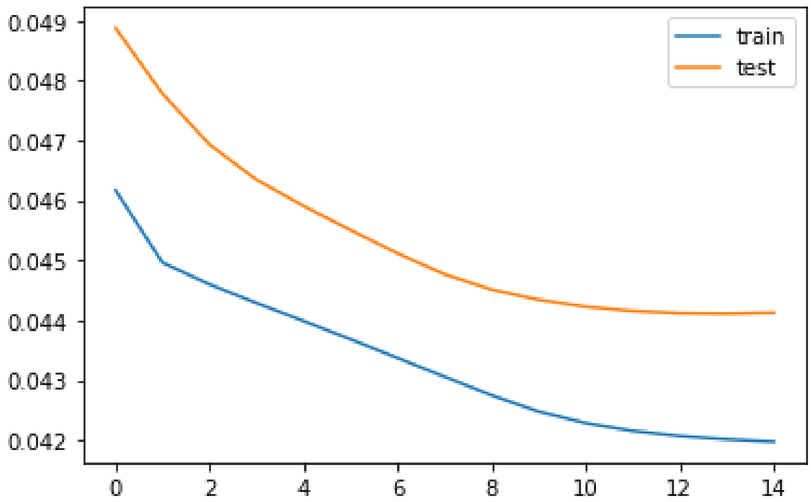
Epoch 14/15

32256/32256 [==============================] - 2s 67us/step - loss: 0.0420 - rmsle_loss: 0.0420 - val_loss: 0.0441 - val_rmsle_loss: 0.0429

Epoch 15/15

32256/32256 [==============================] - 2s 51us/step - loss: 0.0420 - rmsle_loss: 0.0420 - val_loss: 0.0441 - val_rmsle_loss: 0.0429

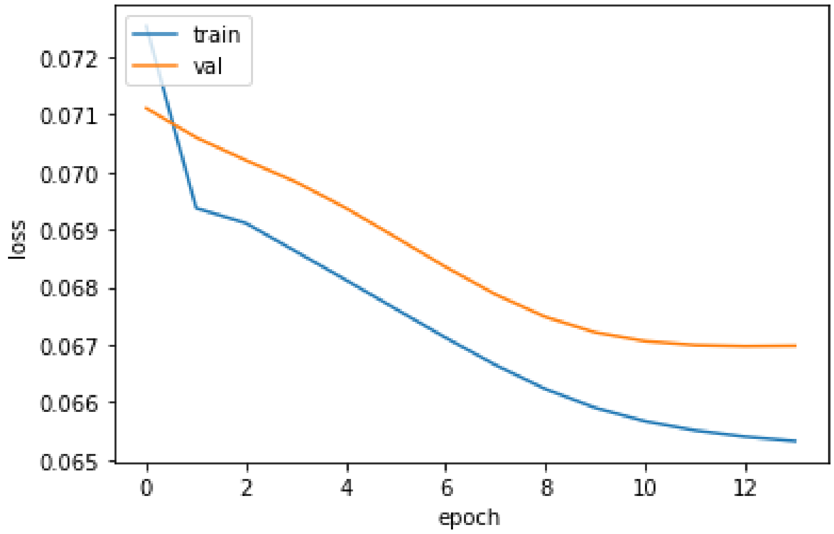
Epoch 00015: early stopping



Extract 9: Household from Affluent-Std group with LSTM Multivariate Model

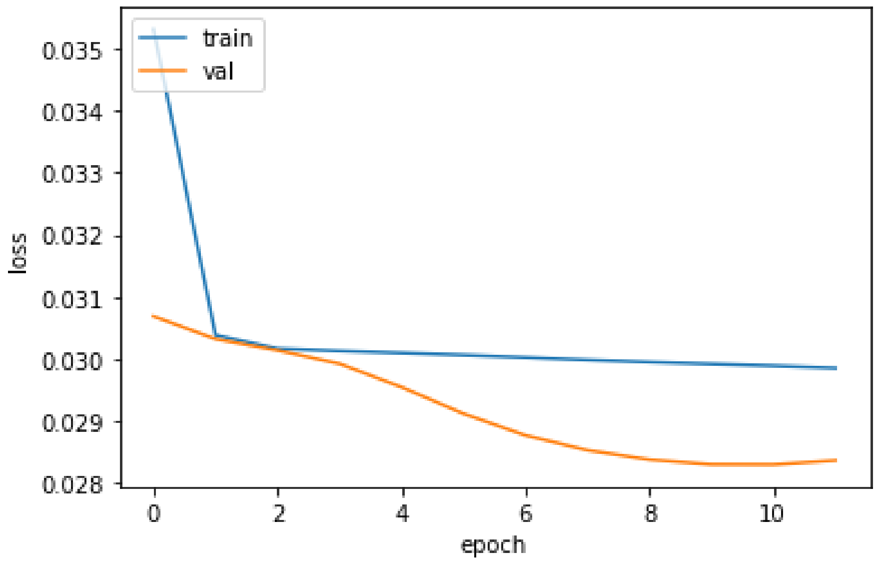
```
Epoch 1/15
32256/32256 [==============================] - 2s 65us/step - loss: 0.0725 - rmsle_loss: 0.0725 - val_loss: 0.0711 - val_rmsle_loss: 0.0719
Epoch 2/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0694 - rmsle_loss: 0.0694 - val_loss: 0.0706 - val_rmsle_loss: 0.0714
Epoch 3/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0691 - rmsle_loss: 0.0691 - val_loss: 0.0702 - val_rmsle_loss: 0.0709
Epoch 4/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0686 - rmsle_loss: 0.0686 - val_loss: 0.0698 - val_rmsle_loss: 0.0704
Epoch 5/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0681 - rmsle_loss: 0.0681 - val_loss: 0.0694 - val_rmsle_loss: 0.0699
Epoch 6/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0676 - rmsle_loss: 0.0676 - val_loss: 0.0689 - val_rmsle_loss: 0.0694
Epoch 7/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0671 - rmsle_loss: 0.0671 - val_loss: 0.0683 - val_rmsle_loss: 0.0688
Epoch 8/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0666 - rmsle_loss: 0.0667 - val_loss: 0.0679 - val_rmsle_loss: 0.0683
Epoch 9/15
32256/32256 [==============================] - 2s 48us/step - loss: 0.0662 - rmsle_loss: 0.0662 - val_loss: 0.0675 - val_rmsle_loss: 0.0678
Epoch 10/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0659 - rmsle_loss: 0.0659 - val_loss: 0.0672 - val_rmsle_loss: 0.0675
Epoch 11/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0657 - rmsle_loss: 0.0657 - val_loss: 0.0671 - val_rmsle_loss: 0.0673
Epoch 12/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0655 - rmsle_loss: 0.0655 - val_loss: 0.0670 - val_rmsle_loss: 0.0672
Epoch 13/15
```

```
32256/32256 [==============================] - 2s 47us/step - loss: 0.0654 - rmsle_loss: 0.0654 - val_loss: 0.0670 - val_rmsle_loss: 0.0672
Epoch 14/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0653 - rmsle_loss: 0.0653 - val_loss: 0.0670 - val_rmsle_loss: 0.0672
Epoch 00014: early stopping
```



Extract 10: Household from Affluent-Tou group with LSTM Multivariate Model

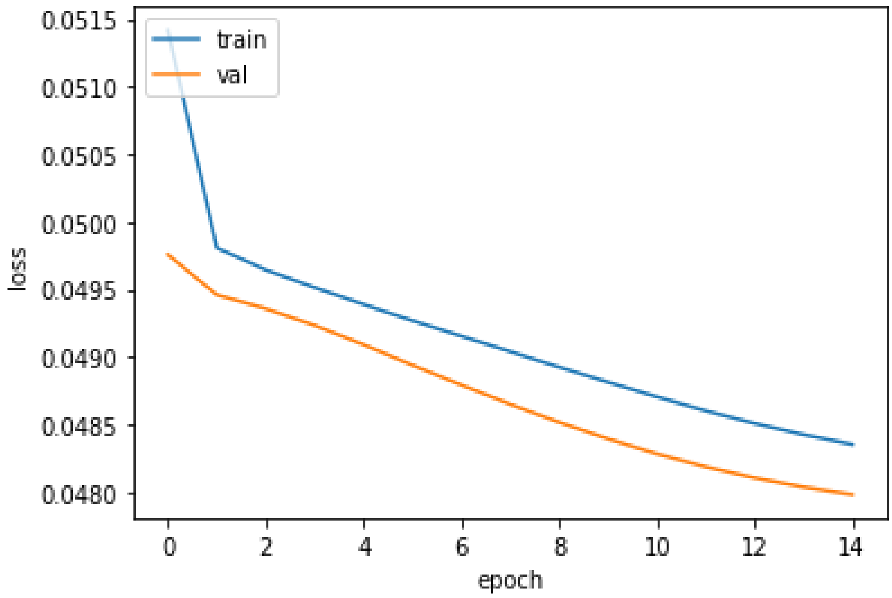
```
Epoch 1/15
32256/32256 [==============================] - 2s 53us/step - loss: 0.0353 - rmsle_loss: 0.0353 - val_loss: 0.0307 - val_rmsle_loss: 0.0313
Epoch 2/15
32256/32256 [==============================] - 1s 37us/step - loss: 0.0304 - rmsle_loss: 0.0304 - val_loss: 0.0303 - val_rmsle_loss: 0.0309
Epoch 3/15
32256/32256 [==============================] - 1s 37us/step - loss: 0.0302 - rmsle_loss: 0.0302 - val_loss: 0.0301 - val_rmsle_loss: 0.0308
Epoch 4/15
32256/32256 [==============================] - 1s 38us/step - loss: 0.0301 - rmsle_loss: 0.0301 - val_loss: 0.0299 - val_rmsle_loss: 0.0305
Epoch 5/15
32256/32256 [==============================] - 2s 49us/step - loss: 0.0301 - rmsle_loss: 0.0301 - val_loss: 0.0295 - val_rmsle_loss: 0.0302
Epoch 6/15
32256/32256 [==============================] - 2s 50us/step - loss: 0.0301 - rmsle_loss: 0.0301 - val_loss: 0.0291 - val_rmsle_loss: 0.0297
Epoch 7/15
32256/32256 [==============================] - 2s 50us/step - loss: 0.0300 - rmsle_loss: 0.0300 - val_loss: 0.0288 - val_rmsle_loss: 0.0294
Epoch 8/15
32256/32256 [==============================] - 1s 39us/step - loss: 0.0300 - rmsle_loss: 0.0300 - val_loss: 0.0285 - val_rmsle_loss: 0.0291
Epoch 9/15
32256/32256 [==============================] - 1s 36us/step - loss: 0.0300 - rmsle_loss: 0.0299 - val_loss: 0.0284 - val_rmsle_loss: 0.0290
Epoch 10/15
32256/32256 [==============================] - 1s 46us/step - loss: 0.0299 - rmsle_loss: 0.0299 - val_loss: 0.0283 - val_rmsle_loss: 0.0289
Epoch 11/15
32256/32256 [==============================] - 1s 45us/step - loss: 0.0299 - rmsle_loss: 0.0299 - val_loss: 0.0283 - val_rmsle_loss: 0.0289
Epoch 12/15
32256/32256 [==============================] - 1s 37us/step - loss: 0.0299 - rmsle_loss: 0.0299 - val_loss: 0.0284 - val_rmsle_loss: 0.0289
Epoch 00012: early stopping
```


train
val
0.035
0.034
0.033
0.032
0.031
0.030
0.029
0.028
loss
0
2
4
6
8
10
epoch

Extract 11: Household from Comfortable-Std group with LSTM Multivariate Model

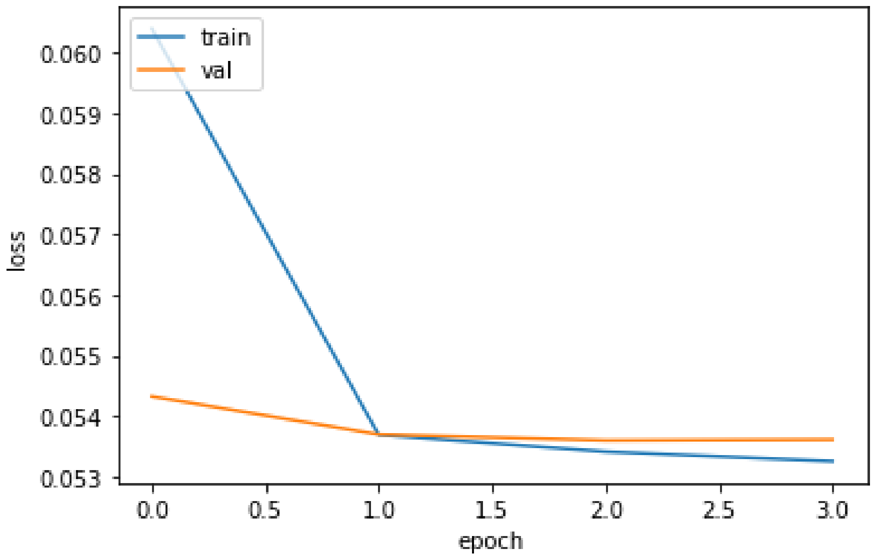
```
Epoch 1/15
32256/32256 [==============================] - 2s 52us/step - loss: 0.0514 - rmsle_loss: 0.0514 - val_loss: 0.0498 - val_rmsle_loss: 0.0515
Epoch 2/15
32256/32256 [==============================] - 1s 33us/step - loss: 0.0498 - rmsle_loss: 0.0498 - val_loss: 0.0495 - val_rmsle_loss: 0.0511
Epoch 3/15
32256/32256 [==============================] - 1s 33us/step - loss: 0.0496 - rmsle_loss: 0.0496 - val_loss: 0.0494 - val_rmsle_loss: 0.0509
Epoch 4/15
32256/32256 [==============================] - 1s 32us/step - loss: 0.0495 - rmsle_loss: 0.0495 - val_loss: 0.0492 - val_rmsle_loss: 0.0507
Epoch 5/15
32256/32256 [==============================] - 1s 34us/step - loss: 0.0494 - rmsle_loss: 0.0494 - val_loss: 0.0491 - val_rmsle_loss: 0.0506
Epoch 6/15
32256/32256 [==============================] - 1s 30us/step - loss: 0.0493 - rmsle_loss: 0.0493 - val_loss: 0.0489 - val_rmsle_loss: 0.0504
Epoch 7/15
32256/32256 [==============================] - 1s 31us/step - loss: 0.0492 - rmsle_loss: 0.0491 - val_loss: 0.0488 - val_rmsle_loss: 0.0502
Epoch 8/15
32256/32256 [==============================] - 1s 31us/step - loss: 0.0490 - rmsle_loss: 0.0490 - val_loss: 0.0486 - val_rmsle_loss: 0.0500
Epoch 9/15
32256/32256 [==============================] - 1s 32us/step - loss: 0.0489 - rmsle_loss: 0.0489 - val_loss: 0.0485 - val_rmsle_loss: 0.0498
Epoch 10/15
32256/32256 [==============================] - 1s 30us/step - loss: 0.0488 - rmsle_loss: 0.0488 - val_loss: 0.0484 - val_rmsle_loss: 0.0496
Epoch 11/15
32256/32256 [==============================] - 1s 30us/step - loss: 0.0487 - rmsle_loss: 0.0487 - val_loss: 0.0483 - val_rmsle_loss: 0.0495
Epoch 12/15
32256/32256 [==============================] - 1s 31us/step - loss: 0.0486 - rmsle_loss: 0.0486 - val_loss: 0.0482 - val_rmsle_loss: 0.0494
Epoch 13/15
```

```
32256/32256 [==============================] - 1s 39us/step - loss: 0.0485 - rmsle_loss: 0.0485 - val_loss: 0.0481 - val_rmsle_loss: 0.0493
Epoch 14/15
32256/32256 [==============================] - 1s 40us/step - loss: 0.0484 - rmsle_loss: 0.0484 - val_loss: 0.0480 - val_rmsle_loss: 0.0492
Epoch 15/15
32256/32256 [==============================] - 1s 44us/step - loss: 0.0484 - rmsle_loss: 0.0483 - val_loss: 0.0480 - val_rmsle_loss: 0.0491
```



Extract 12: Household from Comfortable-Tou group with LSTM Multivariate Model

```
Epoch 1/15
32256/32256 [==============================] - 3s 87us/step - loss: 0.0604 - rmsle_loss: 0.0604 - val_loss: 0.0543 - val_rmsle_loss: 0.0548
Epoch 2/15
32256/32256 [==============================] - 1s 35us/step - loss: 0.0537 - rmsle_loss: 0.0537 - val_loss: 0.0537 - val_rmsle_loss: 0.0542
Epoch 3/15
32256/32256 [==============================] - 2s 50us/step - loss: 0.0534 - rmsle_loss: 0.0534 - val_loss: 0.0536 - val_rmsle_loss: 0.0541
Epoch 4/15
32256/32256 [==============================] - 2s 47us/step - loss: 0.0533 - rmsle_loss: 0.0533 - val_loss: 0.0536 - val_rmsle_loss: 0.0542
Epoch 00004: early stopping
```



Here we show each model's more detailed results, so we have the minimum RMSLE for a given ACORN-Tariff, the maximum RMSLE and the average RMSLE for the whole group.

1) Average Basic Model

Table 22 shows the results for the benchmark model 'Average Basic' for each group.

Table 22: Minimum/Average/Maximum RMSLE for reach group

		Minimum	Average	Maximum
Average basic	*Adversity – Std*	0.028	0.152	0.430
	Adversity – ToU	0.058	0.158	0.273
	Affluent – Std	0.025	0.171	0.526
	Affluent – ToU	0.076	0.166	0.349
	Comfortable – Std	0.001	0.147	0.437
	Comfortable – ToU	0.079	0.141	0.254

2) Average Weekly

Table 23 shows the results for the benchmark model 'Average Weekly' for each group.

Table 23: Minimum/Average/Maximum RMSLE for reach group

		Minimum	Average	Maximum
Average Weekly	*Adversity – Std*	0.022	0.130	0.369
	Adversity – ToU	0.051	0.139	0.226
	Affluent – Std	0.026	0.149	0.472

Average Weekly	*Affluent – ToU*	0.060	0.144	0.299
	Comfortable – Std	0.002	0.128	0.424
	Comfortable – ToU	0.066	0.125	0.256

3) ARIMA

Table 24 shows the results for the ARIMA models for each group.

Table 24: Minimum/Average/Maximum RMSLE for reach group

		Minimum	**Average**	**Maximum**
ARIMA	*Adversity – Std*	0.028	0.153	0.430
	Adversity – ToU	0.058	0.158	0.273
	Affluent – Std	0.025	0.171	0.526
	Affluent – ToU	0.076	0.166	0.349
	Comfortable – Std	0.001	0.147	0.437
	Comfortable – ToU	0.079	0.141	0.253

4) LSTM Univariate Model

Table 25 shows the results for the LSTM Univariate model for each group.

Table 25: Minimum/Average/Maximum RMSLE for reach group

		Minimum	**Average**	**Maximum**
LSTM Univariate	*Adversity - Std*	0.030	0.132	0.432
	Adversity - ToU	0.040	0.153	0.315

LSTM Univariate	*Affluent - Std*	0.010	0.160	0.574
	Affluent - ToU	0.051	0.130	0.315
	Comfortable - Std	0.022	0.125	0.295
	Comfortable - ToU	0.051	0.120	0.216

5) LSTM Multivariate Model

Table 26 shows the results for the LSTM Multivariate model for each group.

Table 26: Minimum/Average/Maximum RMSLE for reach group

		Minimum	**Average**	**Maximum**
LSTM Multivariate	*Adversity - Std*	0.026	0.111	0.274
	Adversity - ToU	0.051	0.114	0.174
	Affluent - Std	0.017	0.112	0.260
	Affluent - ToU	0.058	0.106	0.247
	Comfortable - Std	0.002	0.103	0.257
	Comfortable - ToU	0.048	0.098	0.158

Here we also present a more detailed table of the resulting RMSE for each model and for each ACORN-Tariff group.

Model	Result	Adversity - Std	Adversity-Tou	Affluent-Std	Affluent-Tou	Comfortable-Std	Comfortable-Tou
Avg basic	Min	0.029	0.067	0.032	0.095	0.033	0.101
Avg basic	Avg	0.238	0.246	0.280	0.255	0.226	0.202
Avg basic	Max	0.949	0.486	1.317	0.618	0.919	0.394
Avg weekly	Min	0.024	0.059	0.035	0.08	0.027	0.083
Avg weekly	Avg	0.2056	0.2217	0.247	0.226	0.202	0.183
Avg weekly	Max	0.836	0.422	0.863	0.545	0.885	0.396
ARIMA	Min	0.029	0.067	0.095	0.095	0.033	0.101
ARIMA	Avg	0.238	0.247	0.255	0.255	0.226	0.202
ARIMA	Max	0.949	0.486	0.618	0.618	0.92	0.392
LSTM Uni	Min	0.033	0.045	0.01	0.059	0.025	0.06
LSTM Uni	Avg	0.171	0.203	0.225	0.171	0.157	0.152
LSTM Uni	Max	0.704	0.469	1.004	0.485	0.476	0.288
LSTM Multi	Min	0.027	0.061	0.017	0.073	0.003	0.061
LSTM Multi	Avg	0.180	0.187	0.189	0.172	0.165	0.145
LSTM Multi	Max	0.624	0.309	0.73	0.458	0.567	0.253