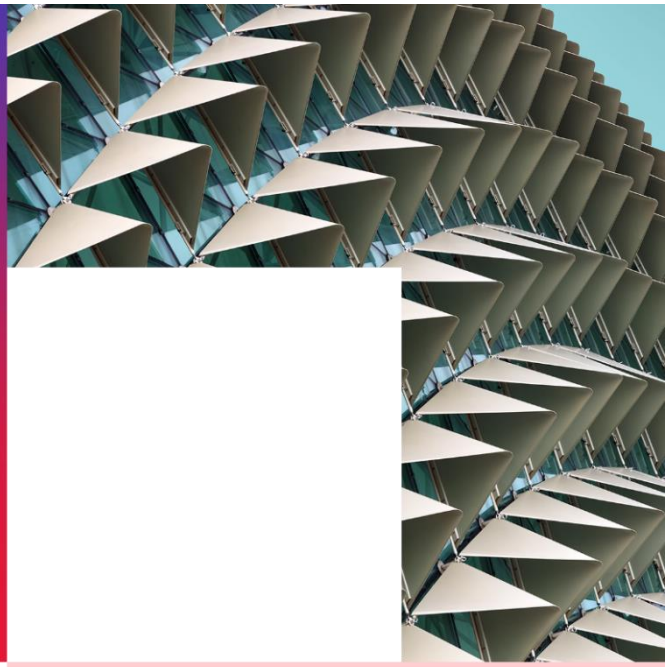


Voltage Validation of Unregistered Equipment Report (WP4-D1)

SMITN

24/05/2023

Prepared by

Iro Psarra, Alex Gardner, Nadim Al-Hariri

CGI

Amendment History

Date	Issue	Status
12/03/2023	0.1	Initial version
12/03/2023	0.2	Revised Version
31/03/2023	2.0	Final Version
14/04/2023	2.1	Updated version following review
27/04/2023	3.0	Clean final version following approval

Contents

1 Introduction

1.1 Background

Smart Meter (SM) data opens opportunities for Distribution Network Operators (DNOs) to improve their database records, develop a Low-Voltage (LV) model which will be useful for network planning, fault detection and phase balancing.

Due to the growth of smart technologies and embedded renewable generation, the energy market is becoming increasingly complex and evolving. This poses significant challenges for Distribution Network Operators in managing their network, especially with the rapidly adoption of Low Carbon Technologies (LCTs) such as embedded generation and Electric Vehicles (EVs).

The SMITN project is an innovation project that will investigate how the use of SM data can be used to support network operations. The third use case, “LCT Detection”, is intended to identify LCTs such as Photo Voltaic solar panels (PVs), electric vehicles and heat pumps. To identify LCTs, several datasets collected from SMs will be utilized.

The ability to identify where LCTs are connected to distribution network at a local level will enable National Grid network planners to assess the current network capacity and anticipate future changes. This enhanced visibility will bring many benefits to DNOs and the customers, such as the potential to avoid or reduce costs associated with network reinforcement and less disruption caused by network upgrades.

Our analysis is focused on 46 secondary substations (SS) in the Milton Keynes area with installed GridKey monitoring devices. For the LCT Detection use case, data with known LCTs installations have been requested from areas outside from the core area.

1.2 Document Purpose

This document presents a high-level description of the algorithms selected for the LCT Detection use case and will present and discuss the results.

The purpose of this analysis is to examine:

- how export data can be used to identify PV and other types of generation,
- how monthly consumption data can be used to identify unregistered LCTs,
- how voltage data can be used to identify unregistered LCTs,
- differences in Machine Learning (ML) algorithms used,
- data issues and limitations.

This will provide useful information to enable National Grid Electricity Distribution (NGED) and other DNOs to implement Business As Usual (BAU) processes to validate existing LCT data and backfilling missing data.

1.3 Overview

Section 2 of this document presents the analysis of HH export data for LCT identification.

Section 3 presents the analysis of aggregated monthly consumption data for LCT identification.

Section 4 presents the analysis of voltage data for LCT identification.

Section 5 discusses the key learning points of this analysis.

1.4 Abbreviations

Term	Definition
BAU	Business As Usual
CB	Circuit Breaker
CIM	(IEC 61970/61968/62325) Common Information Model for the electricity industry.
CSV	Comma-Separated Value
DCC	Data Collection Company (for SM data)
DD	Data Dictionary
DER	Distributed Energy Resource
DMS	Distribution Management System (such as GE PowerOn)
DNO	Distribution Network Operator
EAC	Estimated Annual Consumption
EAM	Enterprise Asset Management (such as ARM, SAP or Oracle)
EO	Electric Office (NGED's GIS)
ERD	Entity-Relationship Diagram
ETL	Extract, Transform and Load
EV	Electric Vehicle
FPR	False Positive Rate
GIS	Geographic Information System (such as ESRI or GE Electric Office/Smallworld)
GUID	Globally Unique Identifier
HH	Half-Hourly
HV	High Voltage
IEC	International Electrotechnical Commission
INM	Integrated Network Model
JSON	JavaScript Object Notation
LCT	Low-Carbon Technology e.g. heat pumps, electric vehicles or photovoltaic generation
LR	Logistic Regression
LV	Low Voltage
MAE	Mean Absolute Error
MC	Measurements Calculator
MDM	Master Data Management
ML	Machine Learning
MPAN	Meter Point Administration Number (core – the 13-digit format)
NGED	National Grid Electricity Distribution
NHH	Non-Half-Hourly

Term	Definition
NOP	Normally Open Point
ODS	Operational Data Store
OGC	Open Geospatial Consortium
PU	Positive and Unlabelled
PV	Photovoltaic solar generation
RDBMS	Relational Database Management System
RMS	Root Mean Square (= $\sqrt{\frac{1}{n}\sum x^2}$)
RMU	Ring Main Unit
ROC	Receiver Operating Characteristic
SCADA	Supervisory Control and Data Acquisition
SM	Smart Meter
SMITN	Smart Meter Innovations and Test Network
SQL	Structured Query Language
SS	Secondary Substation
SVM	Support Vector Machine
TK	Unique INM record identifier
TPR	True Positive Rate
Tx	Transformer
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
VIPQ	Voltage, Current, Active and Reactive Power
WKT	Well-Known Text (an OGC spatial geometry representation format)
XML	Extensible Markup Language

2 HH Export Data

Export data can help DNOs to identify embedded generation which hasn't been recorded in their systems. Export data could be analysed to identify patterns in combination with local weather patterns and other factors that may impact generation. Export data is considered non-personal and can be collected in HH intervals without the need of aggregation.

For example, HH export data can help to identify solar panels by providing information about the electricity generation and export patterns of a property. If the export data shows consistent patterns of electricity being generated and exported during daylight hours, this could suggest that the building has solar panels installed. If there is a sudden spike in export activity during a cloudy day or export activity during night-time it may suggest the presence of a battery storage system.

In this analysis, we are focusing solely on the identification of PVs using HH export data, as there is a low presence of energy storage or other types of generation in the core area.

2.1 Using HH Export Data to Identify PVs

2.1.1 Data Preparation

The core trial area comprises 46 SS with installed GridKey monitoring devices from the Milton Keynes area. HH export readings have been requested for 3785 properties in the core area, from June 2022 until December 2022.

Analysis has been undertaken to identify anomalies in the time-series HH export data. In particular, unusuallyhigh export readings in the HH data (i.e. 4.3GWh in a 30 minute period). Export readings with a value above 2 MWh have been removed.

We requested export data for all the MPANs in the core area even for the ones that do not have an export MPAN registered. Most of the MPAN return 0 export value for every HH interval as they do not have any generator installed in the property. Export readings are available from June 2022 until December 2022.

2.1.2 Exploratory Data Analysis

Figure 1 illustrates the daily average export of the devices with known PV installation in CROWN records. We can observe that electricity is being exported during the daylight hours. Moreover, we can observe that during the summer the export activity is much higher than during the winter. However, we do not have visibility of smaller PV systems where all the energy is self-consumed. These PVs are of less concern due to their smaller capacity.

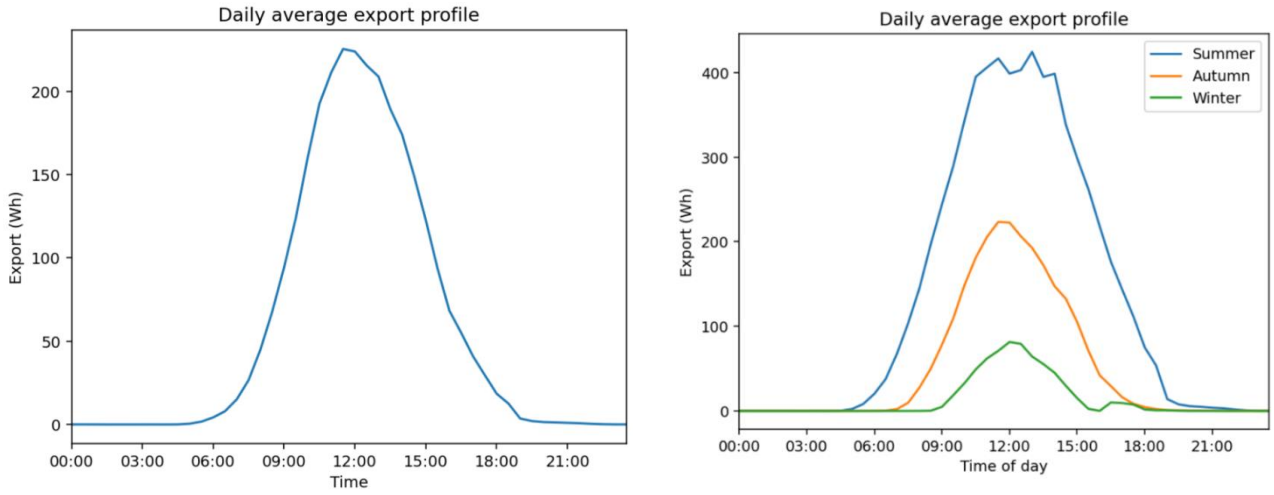


Figure 1: Daily average export for known PV installations.[1]

2.1.3 Modelling Approach

A simple algorithm has been created which compares the daily average maximum export of the devices with "unknown" PV installation with the daily average maximum export during the night.

During the analysis, we found that:

- from the 3785 properties, 137 (3.62%) have been recorded as having a PV in CROWN records.
- during the phase survey, HAYSYS found extra 15 properties with PV installations that haven't been recorded in CROWN.
- out of all the PVs that exist in CROWN or HAYSYS, 94.1% have average export higher than 0 during the day. This means that most of the PVs have exported at some point during the day.
- There are 27 properties that have export but there is no LCT information in CROWN records or in HAYSYS. Figure 2 illustrates an example of a property with export activity during day light hours. Using Google Maps, we were able to verify that 20 of them have PVs (Figure 3). For the rest, a site visit is required in order to verify if they have a PV.

[1] Export data are available from June 2022 until December 2022. For this reason, spring is missing from the diagram.

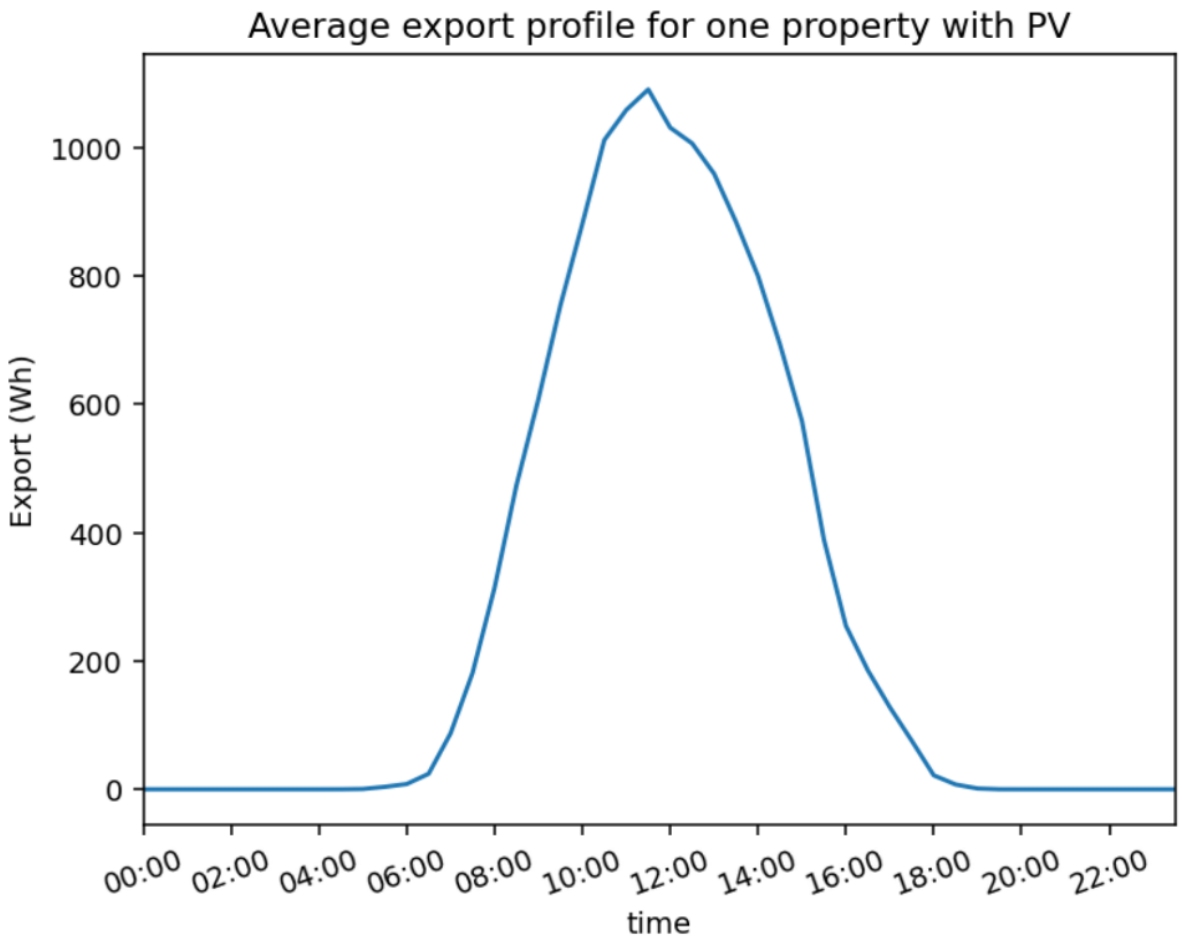


Figure 2: Daily average export profile for a property

Figure 3: Properties with PV that do not exist in CROWN records.

2.1.4 Other types of Generation and Next Steps

2.1.4.1 Limitations

In the core area, there are only 2 devices that have been recorded as having a battery storage, while there are no properties with other types of generation such as wind turbines. Export data from properties with battery storage and wind farms have been requested from the NGED Smart Meter team outside of the core area. These data were not available at the time that this report was written.

2.1.4.2 Analysis

The export data for 15 devices shows a small amount of activity during night-time, as depicted in Figure 4. We need to note that SM clock drift could impact the analysis (see “Phase Identification” report). It is also possible there is another kind of configuration error that confuses 12 hour and 24 hour clock settings, as the export curve appears to reflect that for PV but shifted by 12 hours. There is also the possibility of another type of generator being present on the property. However, a further analysis of export data from other types of generators is necessary to accurately evaluate the existence of other types of generator.

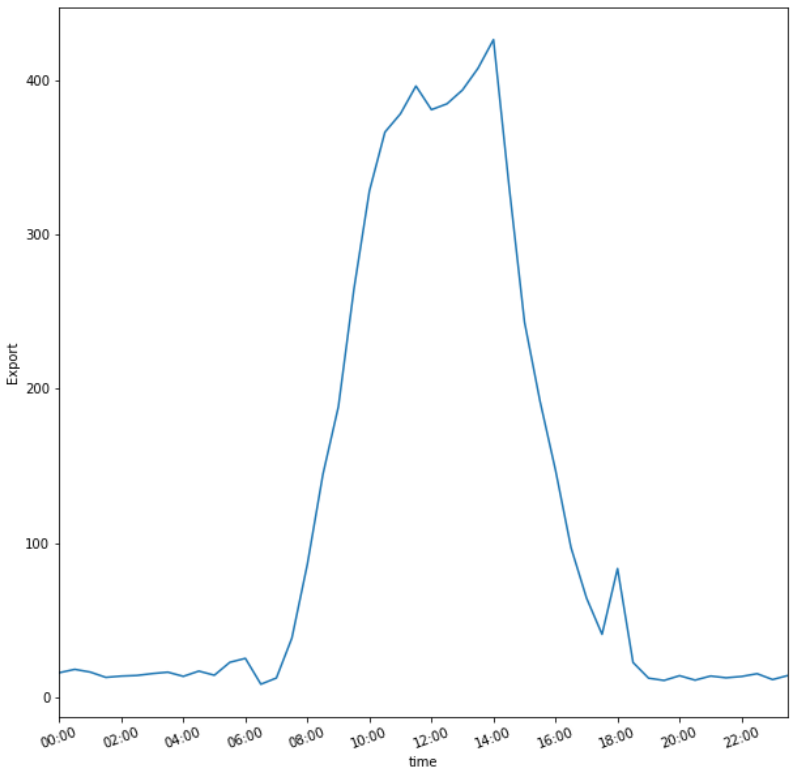


Figure 4: Property with export during the night.

3 Aggregated Monthly Consumption

Monthly aggregated consumption data can help the identification of LCTs by providing insight into patterns of electricity consumption. For example, having an EV, a property may have higher energy demand compared to a property without an EV charger. By analysing the monthly consumption data of individual households, we could identify unusual patterns in energy demand during certain months, which can be indicative of an EV charger. This information can be used by DNOs to better understand the impact of EV charging on the grid and plan accordingly. In addition, it can also help DNOs to identify areas with higher concentration of EVs and understand the usage patterns, allowing them to better manage their networks.

Moreover, when observed over a yearly time span, the monthly demand would be expected to vary significantly if a heat pump is installed, but would be more constant for EV chargers. In this section, we examine these assumptions and try to identify if these relationships exist in the data. We focus on identifying EVs using the monthly aggregated consumption data.

3.1 Data Requests

Monthly aggregated demand data is available for individual domestic MPANs. Monthly consumption data have been requested for 3,938 MPANs in the core area. Due to low number of properties that have been recorded as having an LCT in CROWN, we extended the core area, and we received 6,574 more properties outside of the core area. The data was available from February 2022 until December 2022.

Moreover, Maximum consumption (MC) data and their time that have been recorded are available for all the devices. MC is the highest Wh in any half hour period within a month. The daily MC data was requested but the data were not available at the moment that this report is written.

3.2 Data Issues

- Spurious Large Values

Analysis has been undertaken to identify anomalies in the time-series monthly consumption data. These data quality issues could degrade the performance of the analysis and to the model, so data cleaning was required. In particular spurious large values have been identified in the data (i.e. 2,147,483,647 Wh). The readings that are above 2MWh have been removed from the analysis.

- Missing Data

For the properties where there is only one month missing, we filled the missing month using linear interpolation. Linear interpolation is helpful while searching for a value between a given set of points. It is a method to construct new data points within the range of a discrete set of known data points.

If there was more than one month missing in the consumption data, the property has been dropped from the analysis, as there was not a sufficient number of data points.

3.3 Limitations

There are some limitations and restrictions that affected the performance of the analysis.

- Limited data points

Only 12 months of aggregated monthly consumption data are available for all the requested MPANs at the time that this report was written. MPANs have 12 months range from February to January 2023. Because we have only 12 data points for each MPAN, it is difficult to identify any consumption pattern changes from the installation of an LCT within the 12 months.

- No negative set

The goal of a classification model is to be able to distinguish between positive and negatives examples. To do so, we need data that contains both positive and negative examples which are fully labelled. However, in the LCT Detection use case, the training data consists of positive and unlabelled properties. We have knowledge about LCT installations in CROWN, which is our positive set, but for the rest, we don't have any knowledge if the properties have an LCT installation.

To deal with this issue, we explore several algorithms that are suitable for learning from positive and unlabelled data or PU learning. The assumption is that the unlabelled data can contain both positive and negative examples.

- Limited Data for properties with more than one LCT installation

Very few properties have known installation of more than one type of LCT (i.e., PV and EV) in our datasets. Having two or more types of LCTs can significantly impact the consumption patterns that we want to analyse for each individual LCT installation. For this reason, we excluded these properties of the analysis.

3.4 EV Identification

3.4.1 Exploratory Data Analysis for EVs

In this analysis, we are going to investigate the consumption patterns between MPANs that have EV and MPANs without or "unknown" EV.

Figure 5 illustrates the density plots for the average Active Import (AI) over all months for each customer, the average consumption over the summer and over the winder as well as the density plot for the average Reactive Import. From Figure 5, we can observe that many properties with EVs have higher average consumption. However, we can notice that there is an overlap in average consumption between the properties that have EV and the properties without LCT. Looking on the density plot for the RI, there is no major difference between properties with EV and without EV. However, EVs seems to have lower reactive import than the properties without EVs. Figure 6 illustrates the median demand for each month for properties with EVs and for properties without EVs. From Figure 6 we can observe that the median AI is much higher for the properties with EVs during the whole year

From the scatter plot in Figure 7, we can observe that most of the properties with EVs have high average consumption during the winter and high maximum consumption during the summer. Moreover, we can observe that many properties without EVs, have high consumption in the winter but low maximum consumption during the summer.

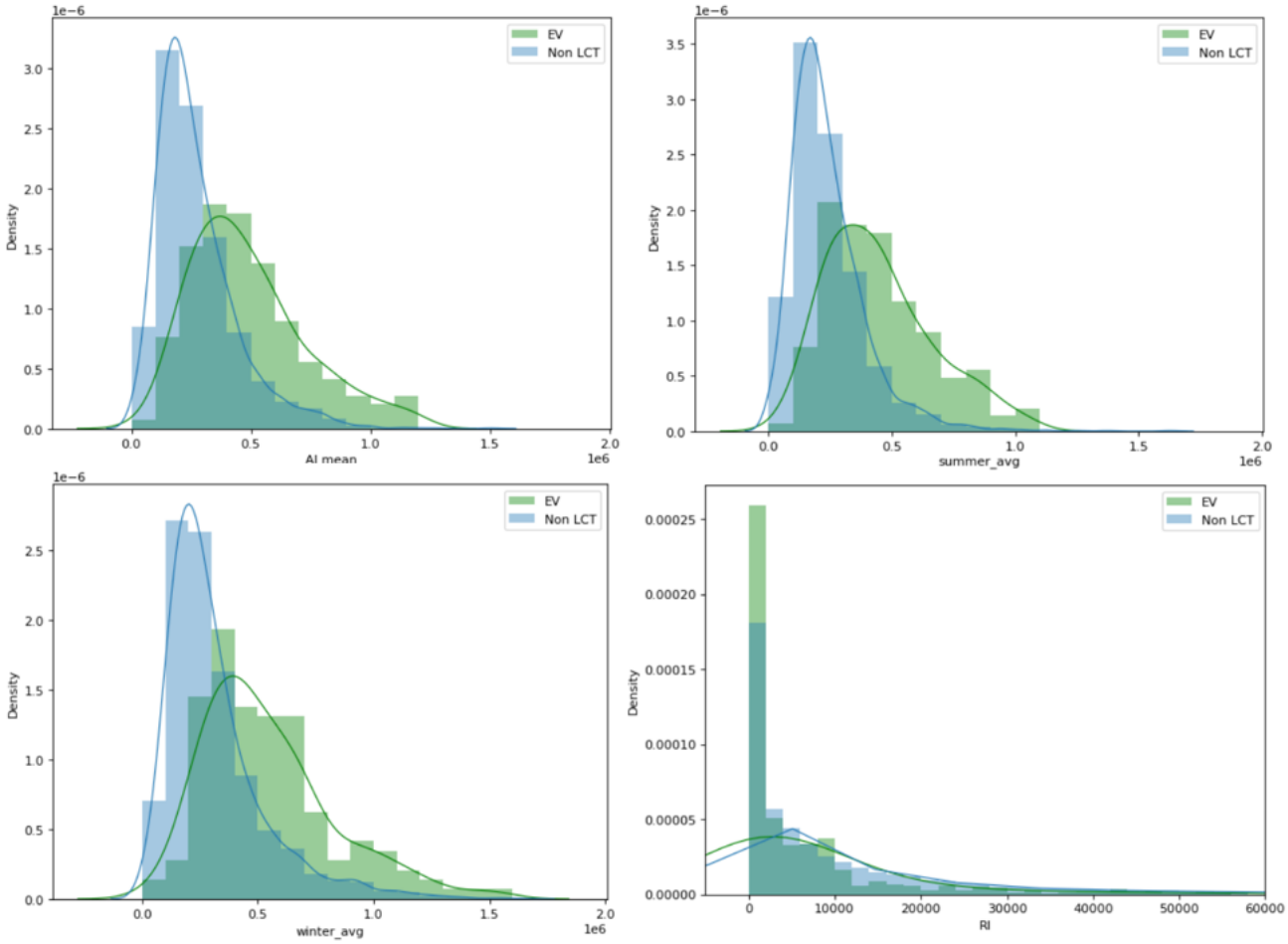


Figure 5: Comparison of mean demand between EV and Non LCT using density plots.

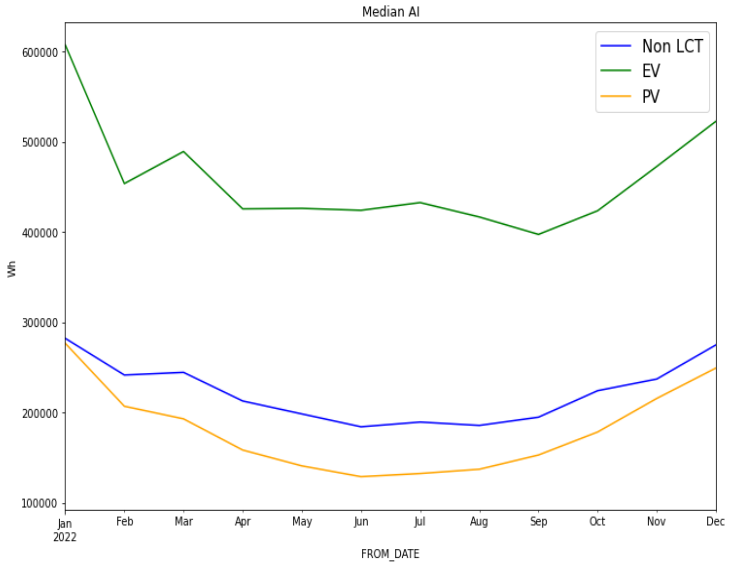


Figure 6: Comparison between EV and Non LCT using the median AI.

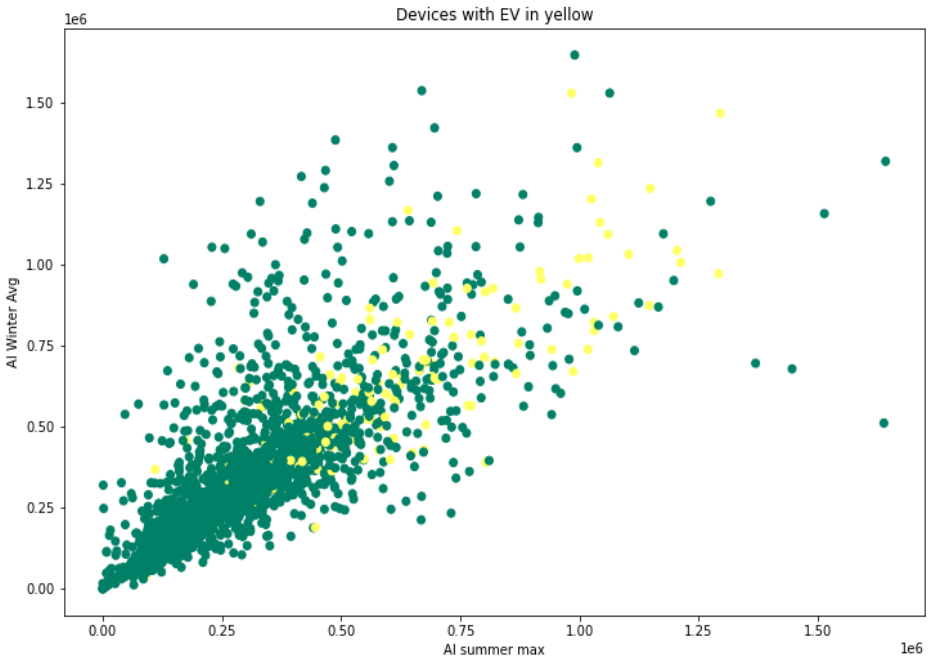


Figure 7: Scatter plot between average AI during the winter and maximum AI during the summer.

Additionally, we analysed the MC data for each month. Figure 8 illustrates the density plot for the median MC over all months for each customer and the density plot for the maximum consumption that have been recorded in any half-hour period during the whole year. From the Figure 8, we can observe that properties with EVs tend to have higher MC that the properties without LCT installation. However, we can notice that there is also an overlap in MC between the properties that have EV and the properties without LCT.

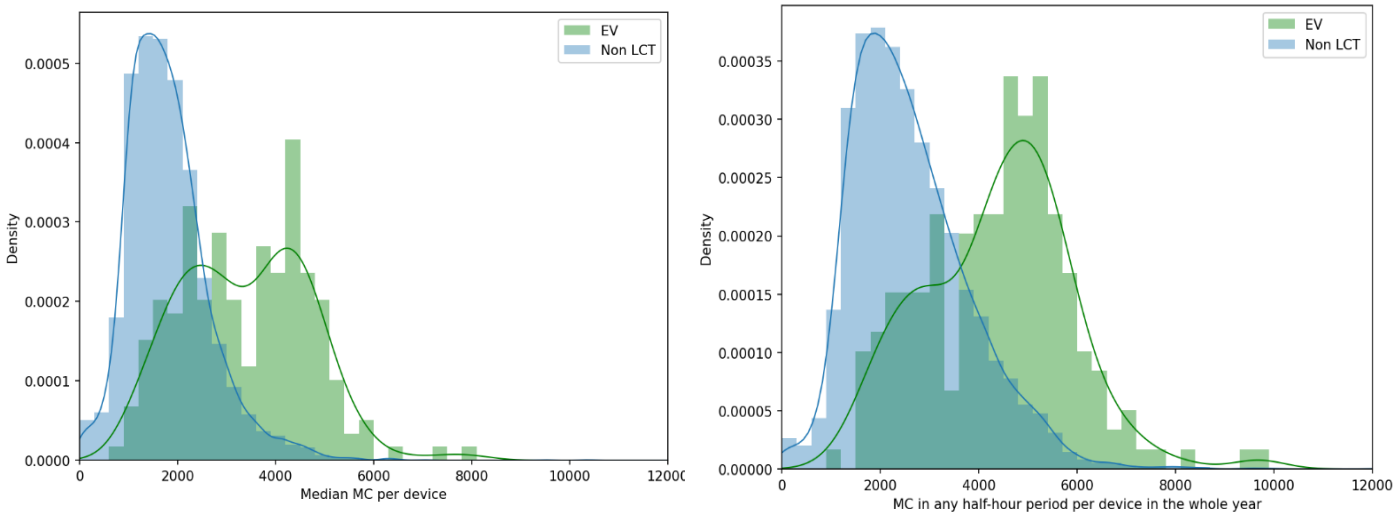


Figure 8: Density plots for Maximum Consumption.

Moreover, analysis was performed in the time that the MC have been recorded for each month. From Figure 9, we can observe that there is not any separation on the MC time between the properties with EVs and properties without LCT installations.

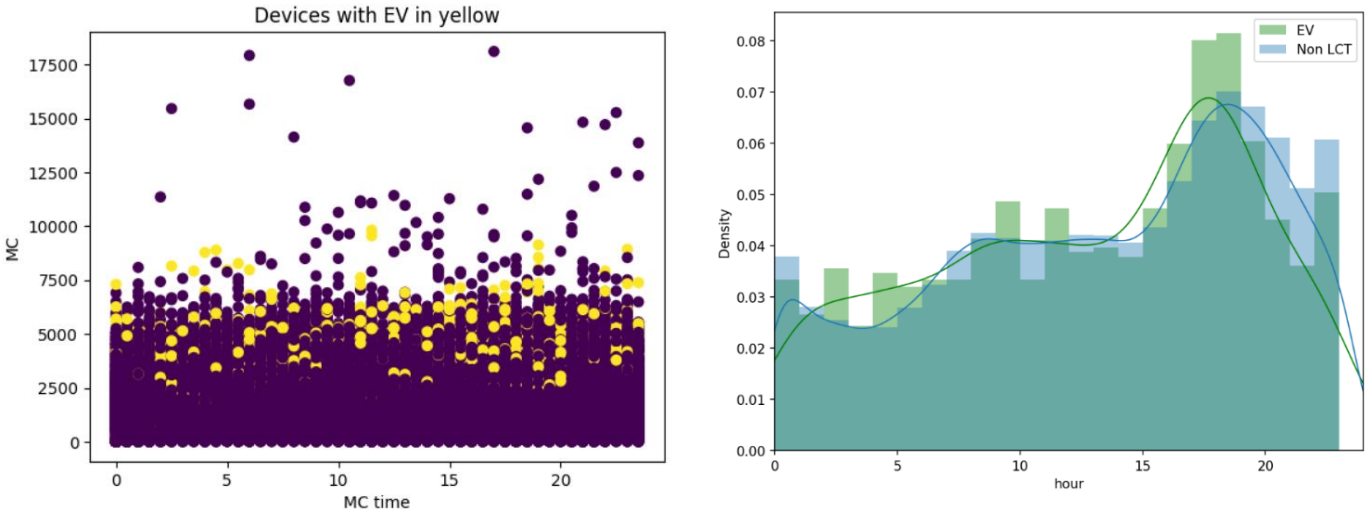


Figure 9: Comparisons between the Maximum Consumption time between properties with EVs and non-LCTs.

3.4.2 Modelling Approach

3.4.2.1 EV Detection through closeness to median curves

A simple approach has been developed to compare the consumption pattern of the properties that have known EVs and the consumption of the non-EV – "unknown" MPANs. The approach is summarized below:

1. Calculation of the median consumption profile for the known EVs.

2. Calculation of the median consumption profile for the "unknown" dataset, assuming that the median curve will be representative of the non-EV consumption data.

3. Compared the consumption profiles for all the MPANs with the two calculated profiles and calculated the Mean Absolute Error (MAE).

4. An MPAN is classified as EV if the MAE with the EV median curve is lower than the MAE with the non-EV median curve.

The results from this process are conventionally presented in a table described as a confusion matrix. Table 1 illustrates the confusion matrix from this approach. Of the 208 MPANs with EVs, 155 have been correctly identified, with 53 not being detected. Figure 10 illustrates an example of a property with EV which classified as non-EV. The consumption is very low in comparison with the median curve of known EVs, so the algorithm was not able to classify the MPAN successfully as an EV. Moreover, 1359 out of the 5844 MPANs (23.8%) with "unknown" EVs have been predicted as having an EV which is a high and probably unrealistic number. This means that using only the monthly values of the consumption data is not enough to have good separator between the MPANs with EVs and non-EVs.

		Predicted	
		True	**False**
Actual	**True**	155	53
	Unlabelled	1359	4335

Table 1: Confusion matrix

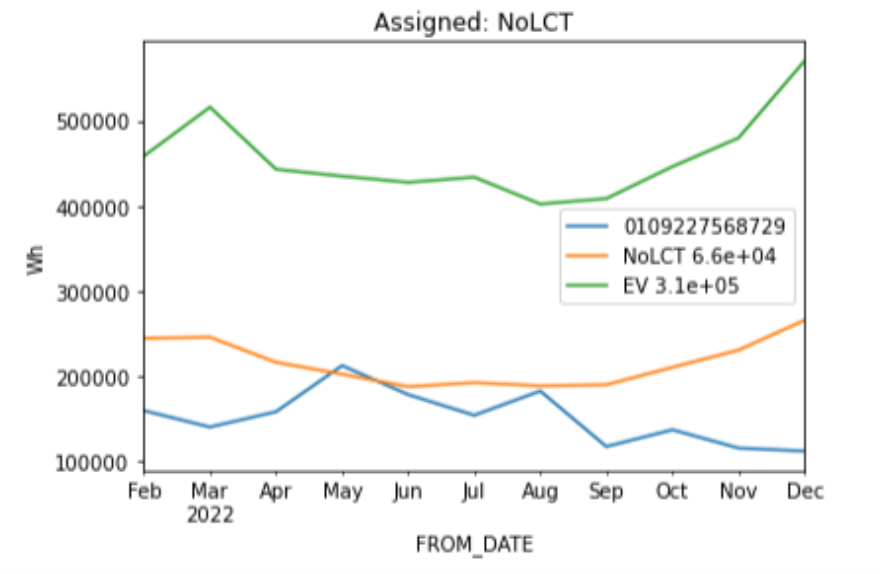


Figure 10: An example of a property with EV which classified as non-EV

3.4.2.2 Binary classification and PU learning

The goal of binary classification is to train a classifier that can distinguish between two classes of instances based on their attributes. In our case if a property has an EV or not. Traditional learning algorithms work in a supervised setting, where the training data is assumed to be fully labelled. The goal of PU (Positive – Unlabelled learning) is to develop models that can learn from partially labelled data, where only positive and unlabelled examples are available, and there are no known negative examples.

3.4.2.2.1 Evaluation Metrics

- True Positive Rate (TPR) or Recall

Recall is a metric used to evaluate the performance of a ML model in correctly identifying all positive samples in a dataset. It is defined as the ratio of true positives to the sum of true positives and false negatives. A high recall value indicates that the model is good identifying positive samples, while a low recall value suggests that the model is missing a large number of positive samples. However, it should be noted that a high recall value may come at a cost of increased false positives, which are incorrectly labelled as positive samples. In our case, the recall is a useful metric, because we can evaluate how the algorithm performs of classifying the known EVs as EVs.

- Receiver Operating Characteristic (ROC)

ROC (Receiver Operating Characteristic) is a popular evaluation metric used in binary classification problems. It is a plot of the true positive rate (TPR) or Recall against the false positive rate (FPR) at various classification thresholds. A better classifier is one whose ROC curve is closer to the top left corner of the plot, indicating a higher TPR for a given FPR. The Area Under the ROC Curve (AUC) is a summary statistic of the ROC curve and is used as a measure of the overall performance of the binary classification model. The AUC ranges from 0 to 1, with a value of 0.5 indicating a random classifier and a value of 1 indicating a perfect classifier. A higher AUC indicates a better overall performance of the classification model.

- Partial ROC

The partial ROC curve is a method for evaluating the performance of a model. A partial ROC curve is a plot of TPR versus FPR for a range of classification thresholds. A partial ROC curve is a subset of the full ROC curve that is created by considering only a specific range of FPR values. This is useful in our approach as we are interested of the performance of the model when the FPR is less than 0.1.

3.4.2.2.2 One Class SVM with PCA

The one-class Support Vector Machine (SVM) is a type of ML algorithm that is designed to classify examples into a single class, which is often referred to as the target class. It is commonly used for anomaly detection, where the goal is to identify examples that are significantly different from the normal data. In the context of PU learning, the one-class SVM can be used to identify positive examples from a set of unlabelled examples, where there are no negative examples available. The one-class SVM is trained using only the positive examples and it learns to identify regions of feature space that are most likely to contain positive examples.

In this algorithm, we used the 12 months of AI data and the average of AI over the months as input to the model. Principal Component Analysis (PCA) has been applied which is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while retaining the maximum amount of information possible.

The known EVs are split into a training and test set using a 75% - 25% ratio, where we develop the algorithm using the training set, and evaluate the performance on the test set. This is to avoid the issue of models overfitting to the data, where a model does not actually learn to identify useful patterns in the data, but it instead just memorises the data to make predictions – which means the model won't work well on new data. The confusion matrix from this model is shown in Table 2. The "Actual" is referring to known EVs and the "Predicted" is the algorithm predictions whether an EV exist in the property. The Recall of the model is 0.61 and the predictions of having an LCT is very high (around 30% of the total properties in the test set). Similar with the previous approach, the one class SVM model which takes as input the monthly consumption data cannot distinguish between the ones that are having an EV and the ones that do not have.

		Predicted	
		True	**False**
Actual	**True**	35	22
	Unlabelled	430	993

Table 2: Confusion Matrix on our test set

3.4.2.3 Weighted Logistic Regression/XGBoost/SVM

In this approach, we converted the time series data into statistics that we can train the model with. From the 12 months of AI data for each device, we created several features (statistics). Table 3 illustrates the statistics that have been initially created to be used as input to the model. We assessed the highly correlated predictor variables, and we removed the features that are highly correlated. This step is important in classification modelling because including irrelevant features in a model can lead to overfitting which occurs when the model is too complex and fits the training data too closely, leading to poor performance on new data.

We then trained ML models by giving it selected calculated statistics for the given devices and labelling the devices with known EV chargers as positive and the rest as negative.

Features	Description
Average	The average monthly consumption within the 12 months.
Maximum	The maximum consumption within the 12 months.
Standard Deviation	The standard deviation of the Active Import within the 12 months.

Minimum Difference	The minimum consumption change from one month to the next.
Maximum Difference	The maximum consumption change from one month to the next.
Median Absolute Difference	The median of the absolute consumption changes over the 12 months.
Maximum – Minimum	The difference between the maximum consumption value and the minimum consumption value.
Summer Average Absolute Difference	The average of the absolute consumption changes over the summer.
Winter Average Absolute Difference	The average of the absolute consumption changes over the winter.
Summer Maximum	The maximum consumption during summer.
Summer Average	The average consumption during summer
Winter Maximum	The maximum consumption during winter.
Winter Average	The average consumption during winter.
Standard Deviation / Average	The standard deviation of the Active Import divided by the average consumption.
MC Median	The median of Maximum Consumption in any half-hour period.
MC Mean	The mean of Maximum Consumption in any half-hour period.
MC Standard Deviation	The standard deviation of the MC within the 12 months.
MC Median Absolute Difference	The median of the absolute MC changes over the 12 months.
MC Max	The Maximum of MC within 12 months
MC Summer Max	The Maximum of MC within summer.
MC Winter Max	The Maximum of MC within winter.

Table 3: Features used as input to the model.

1. Weighted Logistic Regression (LR)

Weighted LR is a popular technique used in PU learning. In this approach, the positive examples are assigned a weight of 1, and the unlabelled examples are assigned a weight that reflects their estimated probability of being positive. In our algorithms, we optimise the assigned weight by assessing the performance of the models.

The weighted LR model is then trained using the weighted data, where the positive examples are given higher importance than the unlabelled examples. The resulting model can then be used to predict the probability that a new example is positive or negative.

2. Weighted XGBoost

XGBoost is another powerful ML algorithm used in machine learning competitions because of its high effectiveness at dealing with many ML problems. XGBoost uses gradient boosting to build an ensemble of decision trees. While it is not known for its effectiveness in PU learning, we tried it here to assess whether it can do better than the simpler algorithms we have tried. We apply the weighting to the unlabelled samples as we do with the other models.

3. Weighted Support Vector Machine (SVM)

Finally, we tested the performance of a Weighted Support Vector Machine (SVM) algorithm. SVMs are popular ML algorithm used for classification tasks, and they are one of the main model types in the PU literature. The basic idea of SVM is to find a hyperplane in the n-dimensional feature space that separates the data into two classes in a way that maximizes the margin between the classes. The weighted SVM can be adapted by assigning weights to both positive and unlabelled examples.

3.4.2.4 Performance Evaluation

In Figure 11, we can observe the ROC curves for the three algorithms. We can observe that the SVM performs better for a FPR less than 0.1.

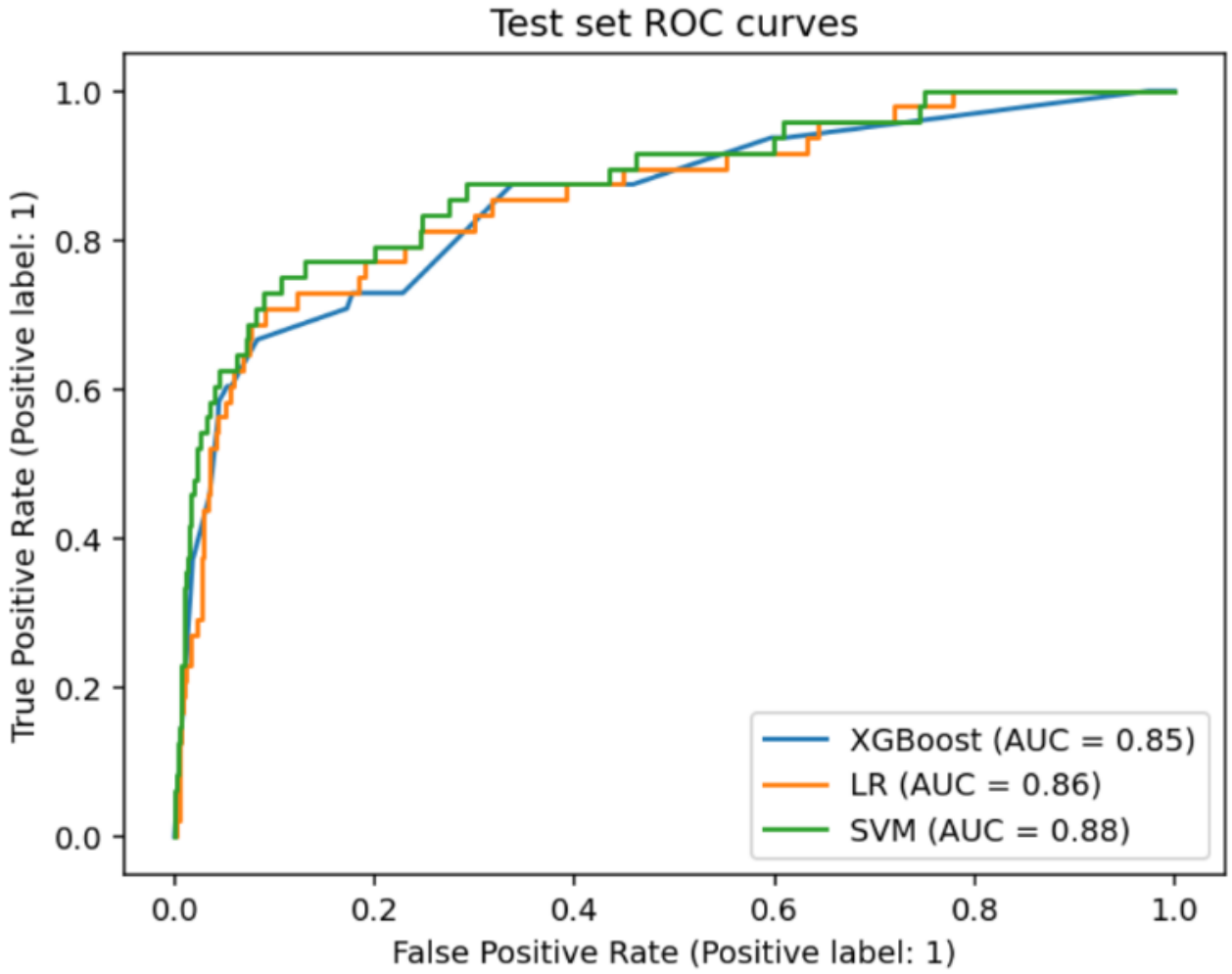


Figure 11: ROC curves for XGBoost/Linear Regression/SVM

To assess the performance of the models we applied cross-validation. Cross-validation is a popular technique used in machine learning to evaluate the performance of a model. It involves dividing a dataset into multiple subsets, using one subset for training a model and the remaining subsets for testing its performance. By repeatedly performing this process with different subsets, cross-validation provides a more robust estimate of a model's generalization ability than simply using a single training/test split. It can also help identify potential issues such as overfitting or underfitting.

Model	ROC-AUC	Standardised Partial ROC-AUC	True Positive Rate	False Positive Rate
Logistic Regression	0.88	**0.75**	0.71	0.12
SVM	0.87	**0.78**	0.71	0.08
XGBoost	0.85	**0.71**	0.7	0.14

Table 4: Logistic Regression vs SVM vs XGBoost

We need to note that we can increase or decrease the TPR by adjusting the classification threshold. A classification threshold is the value used to determine which class a predicted sample belongs to. Specifically, it

is the probability threshold that is used to determine whether a predicted probability for a given sample belongs to the positive class or the negative class. A lower threshold could increase the TPR, but it will also increase the FPR.

Standardised Partial ROC-AUC is a metric that can effectively compare the three approaches, and it only looks at classification thresholds that yield acceptable FPR values. Based on the Standardised Partial ROC-AUC scores, the SVM model performs better than the other models. Moreover, in Table 4 we picked a classification threshold in order for the TPR to be around 0.7 in the three models. We can observe that the FPR for the SVM is 0.08, while it increases significantly for the other two models.

Moreover, feature importance is a measure of the contribution of each feature in a machine learning model to the prediction task. Feature importance can be measured using different methods, such as coefficient value and F-score. F-score can be used to determine the importance of each feature by measuring how well it separates the classes. In linear regression models, the coefficient value is another way that can indicate the feature importance. A higher coefficient value indicates a more important feature. In Figure 12, we can observe the most important features used by the XGBoost model.

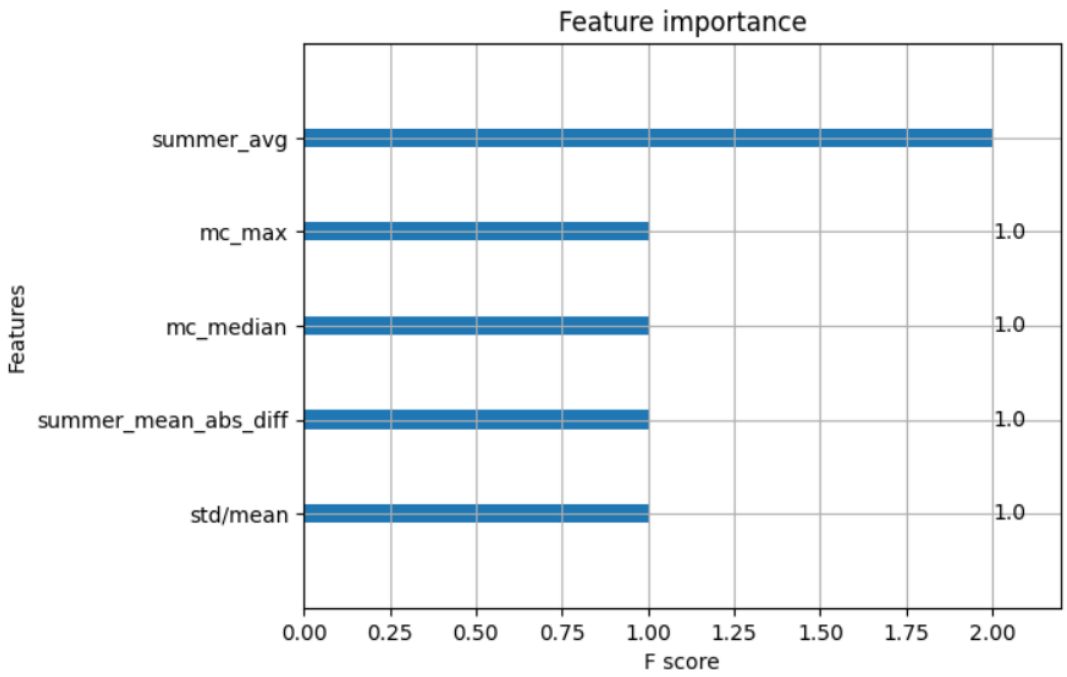


Figure 12: Feature Importance for Weighted XGBoost

3.4.2.5 Comparing model outputs

SVM vs Logistic Regression

Table 5 illustrates a comparison between the trained SVM and Logistic Regression models on a test set.

	Logistic Regression

		True	False
SVM	**True**	122	22
	False	71	1261

Table 5: SVM vs Logistic Regression

The numbers are counts of properties in each category. All of the properties predicted by the logistic regression model as having an EV are in the 'True' column, and all those predicted as having an EV by the SVM are in the 'True' row. We can see from this comparison that both the SVM and LR predict 122 properties as having EVs but the SVM identifies 22 more that LR didn't. Likewise, LR predicted 71 properties as having EVs that the SVM didn't identify (5 of these 71 were labelled as having an EV). This comparison shows that they generally have similar predictions, but the SVM is better at identifying fewer false positives.

3.4.2.6 Ensemble Modelling

As discussed in section 3.4.2.1, the closeness to median curve approach is able to identify the properties that have high consumption, and their consumption curve is very close to the median curve of known EVs. Even though the TPR of the approach is around 75%, the FPR is very high. Tables 6, 7 shows the confusion matrices from the two approaches using the same test set.

		Predicted	
		True	**False**
Actual	**True**	31	14
	Unlabelled	94	1214

Table 6: Confusion matrix for SVM

		Predicted	
		True	**False**
Actual	**True**	33	12
	Unlabelled	315	993

Table 7: Confusion matrix for "Closeness to median curve"

We combine the SVM and the "Closeness to median curve" into one confusion matrix. The predicted positive is when both approaches have predicted the property as having an EV. We can observe that combining the two approaches we decreased the number of the unlabelled properties from 94 properties to 78.

		Predicted	
		True	**False**
Actual	**True**	30	15
	Unlabelled	78	1230

Table 8: Confusion matrix for the combination of SVM and “Closeness to median curve”

Then, we combined the three ML algorithms, and we classified the property as having an EV if two of the three ML approaches suggest that the property has an EV. Table 9 illustrates the confusion matrix. The XGBoost and the LR predict a higher number of properties as having an EV, for this reason the number of the “unlabelled” properties (FPR), that have been predicted as having an EV, has increased.

		Predicted	
		True	**False**
Actual	**True**	32	13
	Unlabelled	138	1188

Table 9: Confusion matrix for the combination of SVM, LR and XGBoost

Subsequently, we combined the three ML algorithms and the “Closeness to median curve” approach and we classified the property as having an EV if three out of four approaches suggest that the property has an EV. In this case, the FPR has decreased in comparison to the previous approach, but it is higher from the FPR when we combined the SVM and the Closeness to median curve. This can be explained from the fact that the SVM is much more restrictive with what it classifies as EV.

		Predicted	
		True	**False**
Actual	**True**	30	15
	Unlabelled	117	1209

Table 10: Confusion matrix for the combination of SVM, LR, XGBoost and Closeness to median curve.

It is important to validate the “unlabelled” properties that have been predicted as EVs with site visits in order to understand if an ensemble modelling or the SVM itself performs better.

4 Voltage Analysis

4.1 Data Requests

In the core area, there are 8809 properties, from which 47% of them have installed a SM up to the end of December 2022. This is slightly higher than the overall average for NGED, which is nearer 40%, as substations with a higher proportion of smart meters were preferentially selected for inclusion in the SMITN test network.

Initially, HH voltage data was requested from the devices for the period 1st May 2022 until 31st August. Around 2200 devices gave a response, which gives a success rate of 57%. A request was then issued to all devices to amend the Average Voltage Measurement Period to 1 minute from the default of 30 minutes. 1800 devices gave a response. One minute voltage data was requested from the beginning of October until 30th November 2022.

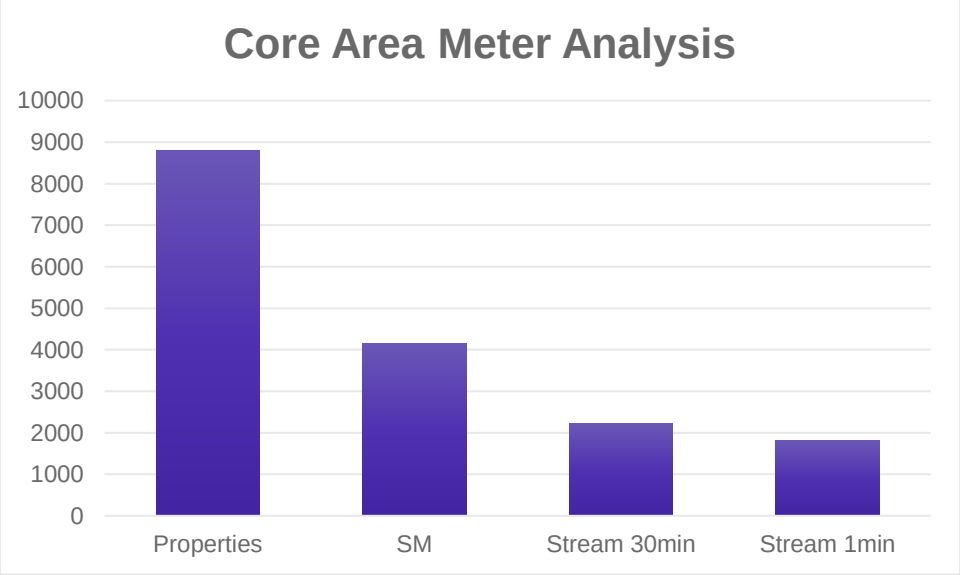


Figure 13: Core Area Device Analysis

4.2 Data Issues

The following validation checks were performed and where possible data was corrected.

1. Data Anomalies

Low SM voltage readings (e.g. 0V) may exist in the time-series voltage data due to reading failures. The presence of very low values may have a negative effect in the model, for this reason records with unreasonable voltage readings were assessed and removed or corrected.

2. Clock Offsets

Another issue that may impact the correlation results is the clock offsets between GridKey voltage data and the SM voltage data, i.e. a discrepancy in voltage data time logging between the SMs and the GridKey data. This issue is particularly present in voltage data with lower resolution (i.e., average 1-minute voltage data). An algorithm that identifies the offset difference between the two sources was created and applied. The aim is to synchronise the two voltage curves.

3. Measurement Interval Synchronisation

The HH voltage readings from SM's use intervals with an arbitrary starting time within clock half hours. Similarly, voltage data with higher time resolutions (i.e. 1 minute voltage data) may start in the middle of minutes. Time synchronisation is important as there is a direct comparison of the voltages from SMs and GridKey for each time period.

As a result, higher resolution voltage data provides not only an improved visibility of the short-term voltage variations, but also a reduction in the time offsets between the measurement intervals used by each SM.

Two approaches have been developed to deal with this issue:

- Timestamp rounding

For the 1-minute voltage data, the timestamp seconds have been rounded to the closest minute, while for the HH data, the minutes have been rounded to the closest 30-minute period.

- Linear Interpolation

Linear interpolation is helpful while searching for a value between a given set of points. It is a method to construct new data points within the range of a discrete set of known data points. In other words, Linear Interpolation allows us to estimate voltage values at times that we do not have them at, given that we have readings soon before and soon after that time.

4.3 Identifying EVs using Voltage Data

4.3.1 Exploratory Data Analysis (EDA)

Exploratory data analysis was performed to identify potential trends and voltage changes that could indicate an LCT installation in the properties with "unknown" status. In this analysis, we used the 1-minute voltage data. Devices with known EVs have occasional drops in voltage for several hours at a time. Figure 14 illustrates an example of three devices that are connected to the same substation and phase. Device 3, which has an EV, appear to have a big drop of the voltage for some hours during the day.

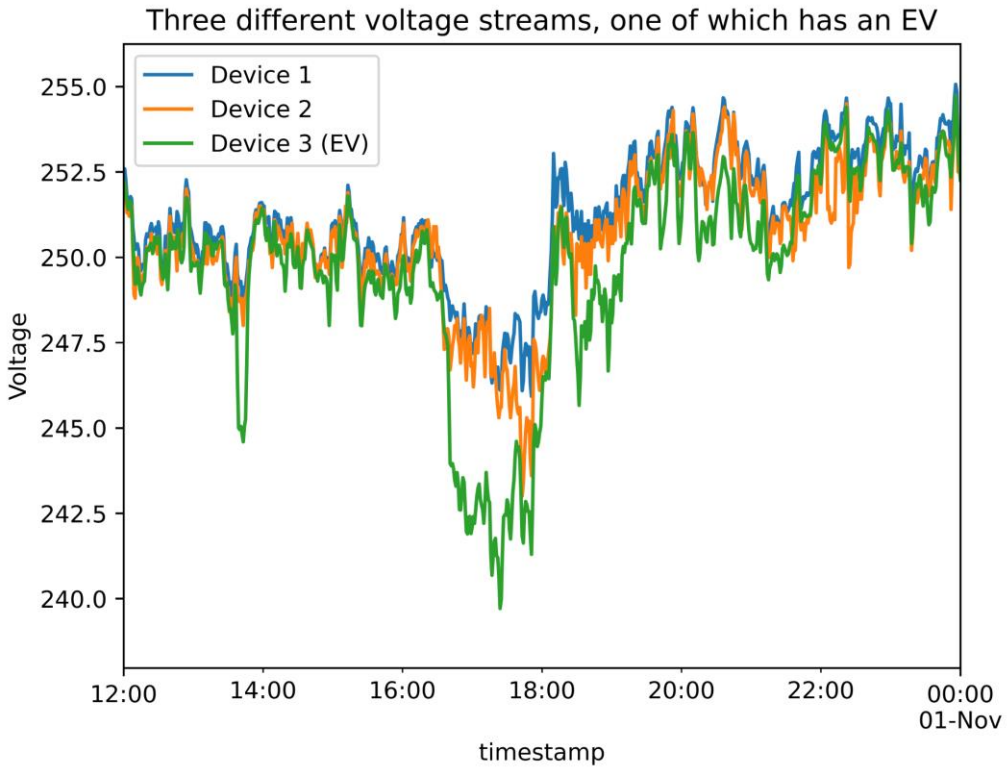


Figure 14: Voltage drops for a device with EV

To prepare the data, we looked at subtracting Grid Key voltage data from each smart meter device to try and cancel out the general voltage trends seen across all the devices on that phase, which should make it easier for a model to identify the drops caused by EVs.

In the absence of GridKey data, we subtract the voltage data from another device on the same phase. This gives a new time series of the difference in voltage between the two devices. The drops in voltage should be much more easily detected by an algorithm because the subtraction with another device removed a lot of the common variations between smart meter devices that connected to the same substation busbar (see Figure 15). In this analysis, we focused on developing the approach using the voltage data from the neighbouring smart meters that belong to the same phase.

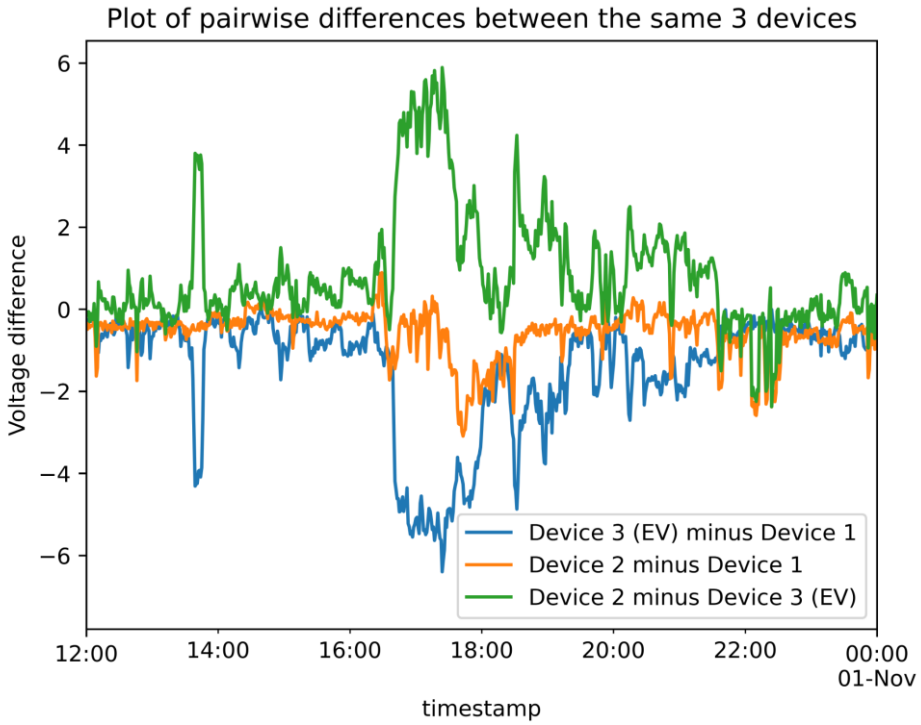


Figure 15: Voltage Difference between the devices.

Figure 16, 17 illustrates one more example of an EV with large drops. From these graphs we can see the large drops around midnight and between 20:00 and 22:30.

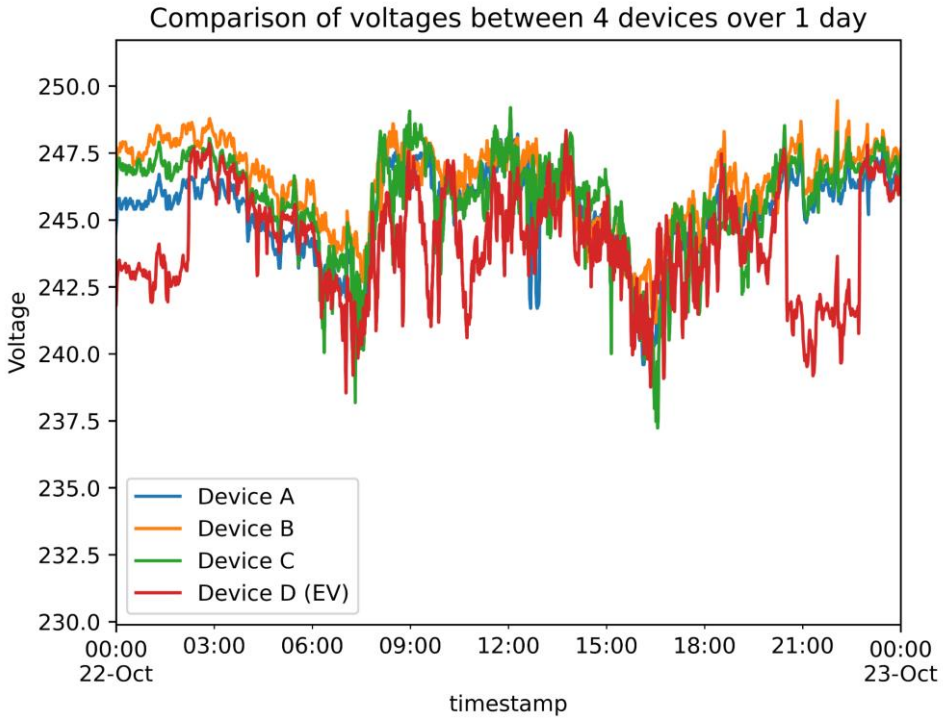


Figure 166: Example of voltage drops for a device with EV

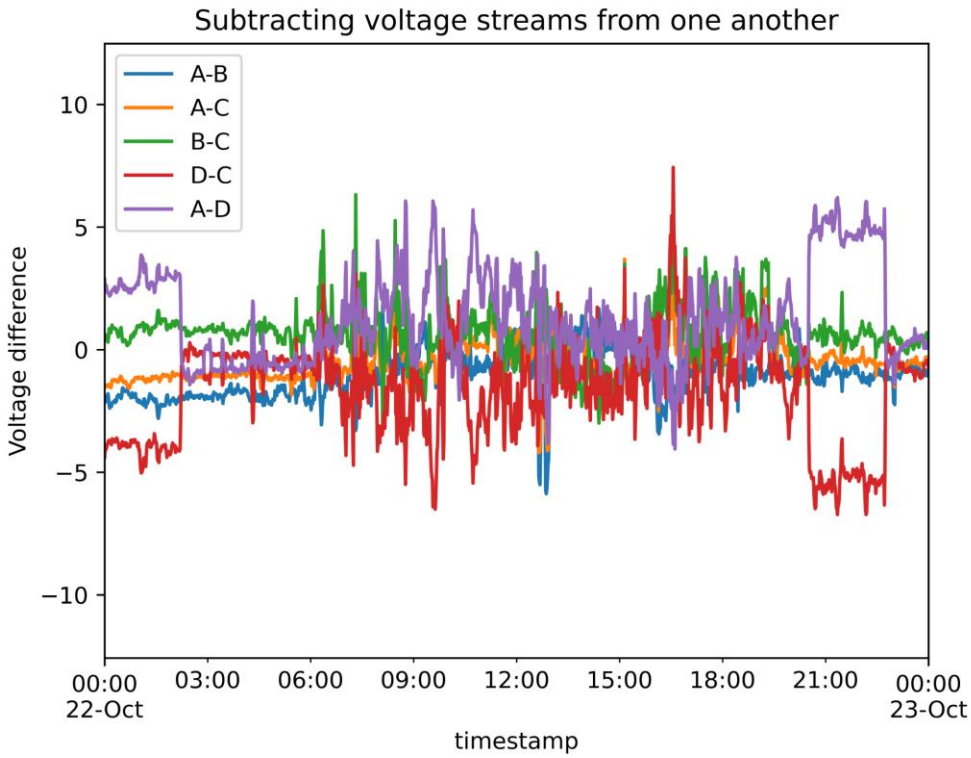


Figure 177: Voltage difference between the devices.

4.3.2 Modelling Approach

To identify the neighbouring SMs which will be used as reference for each examined SM, we developed a function which picks the closest SMs that belong to the same phase and feeder with the examined SM (i.e. the closest 5 SMs). If there was not enough SMs in the same feeder and phase, we expanded our selection criteria to include SM from another feeder but in the same phase. The closer neighbouring SMs will be used as a reference.

For each pair of SMs the algorithm tries to remove the common voltage variations. To do so, the algorithm subtracts the neighbouring SM voltage curve from the examined SM's voltage curve. However, there is a common and continuous voltage difference between the compared devices. To remove the common voltage drop between the devices, we subtracted the median voltage difference between the two devices from the resulted "voltage difference" profile that they had the previous day. Figure 18 illustrates an example of the "voltage difference" profile minus the median of voltage drop. We can observe that the resulted "voltage difference" curve is much closer to zero.

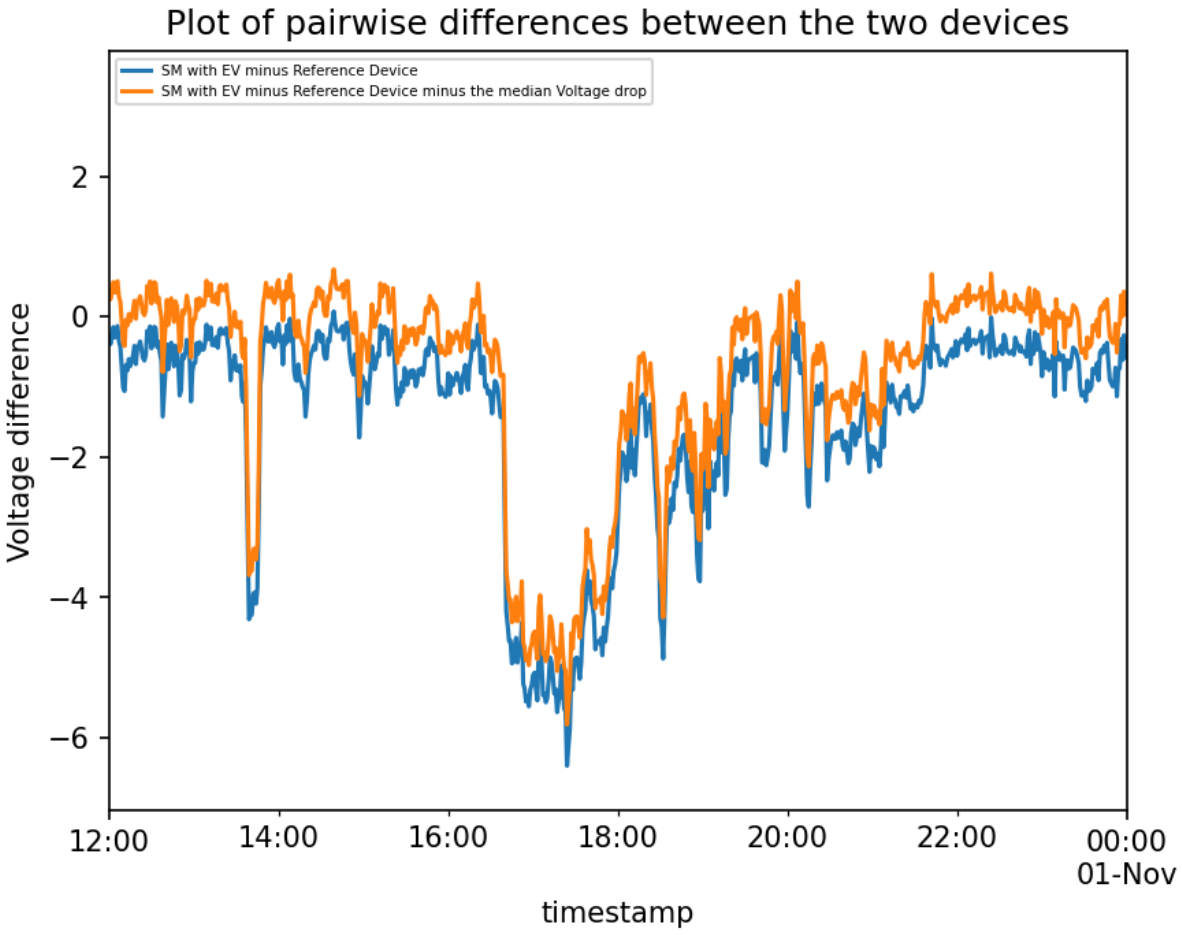


Figure 18:Pairwise difference between SM with EV and reference Device.

Finally, we applied Moving median technique in the resulted “voltage difference” time-series data. A moving median is a statistical method used to smooth out time series data by taking the median value of a sliding window observations over a specified time period. In our case we used a 30 minute time period.

To calculate the moving median, we first choose the window size (i.e. the number of observations to include in each window) and a starting point. Then for each subsequent time point, we slide the window over the data and calculate the median of the observations within the window.

The moving median is very useful technique for identifying trends and patterns in time series data as it can smooth out random fluctuations and highlight underlying patterns. Figure 19 illustrates an example of the voltage difference curve when we apply moving median technique. We can observe that it smooths small and random fluctuations while it enhances the high differences that we want to extract from the time series data.

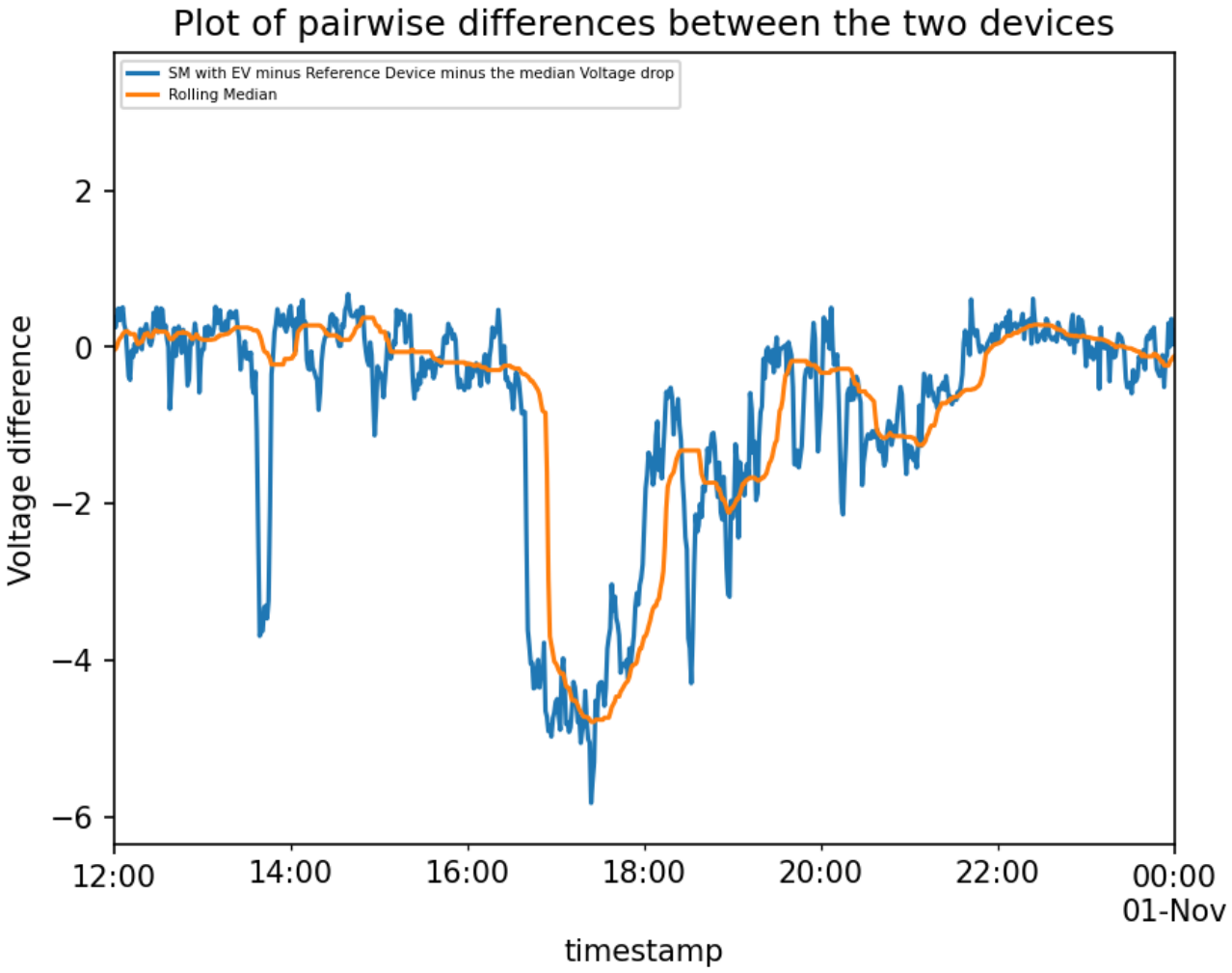
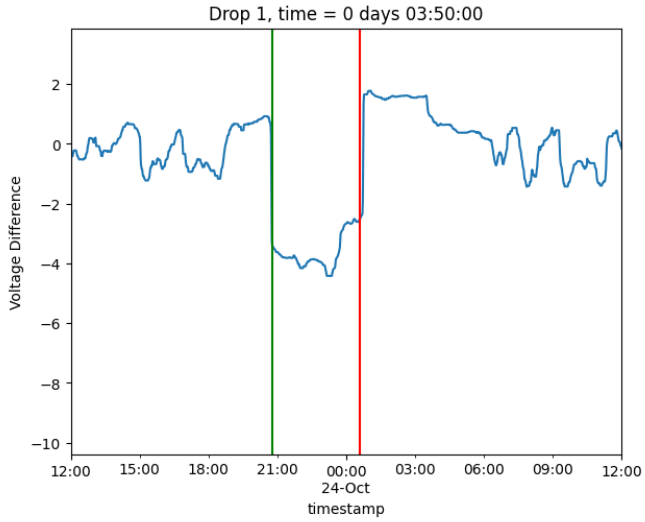
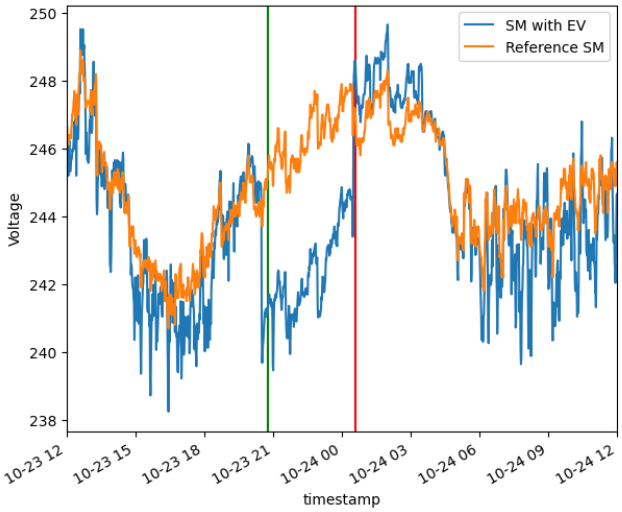


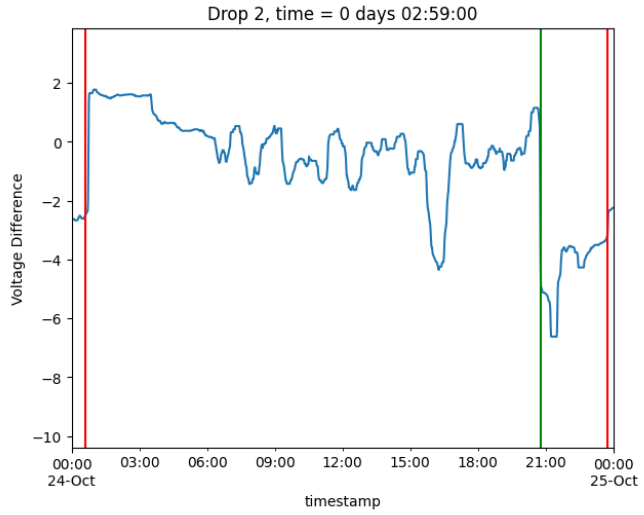
Figure 19:Rolling Median

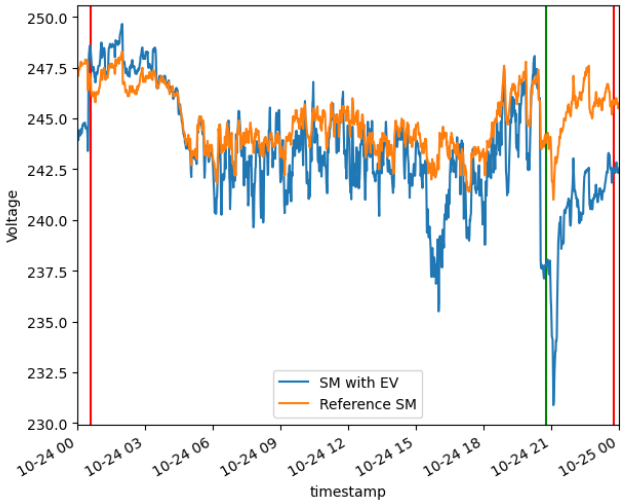
The algorithm flags a voltage drop when the voltage difference goes below a certain threshold (i.e. -2.5 V), and counts the time that the voltage difference remains below of the threshold. To remove drops that happened only for a short period of time and probably caused by other reasons, we set a time threshold (i.e. 1 hour). Figure 20 illustrates some examples of voltage drops and their time identified by the algorithm. Finally, the algorithm repeats the same process for all the SMs that have been identified as reference.

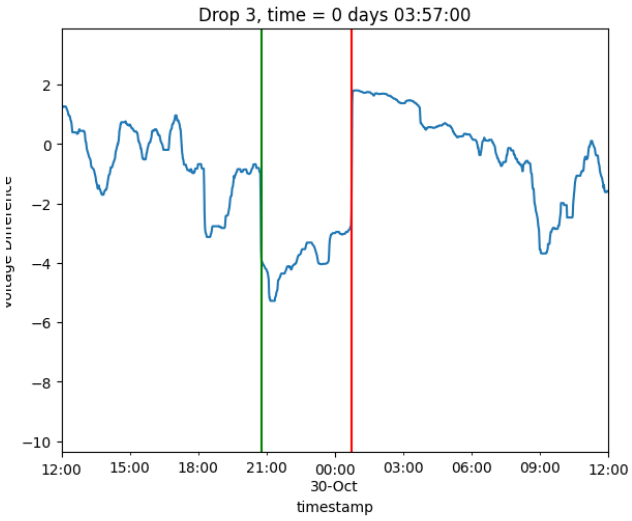
Note that the drop start and end lines on the right plot are about 15 minutes late, because the rolling median over 30 minutes means that all the drops it detects are delayed by around 15 minutes.

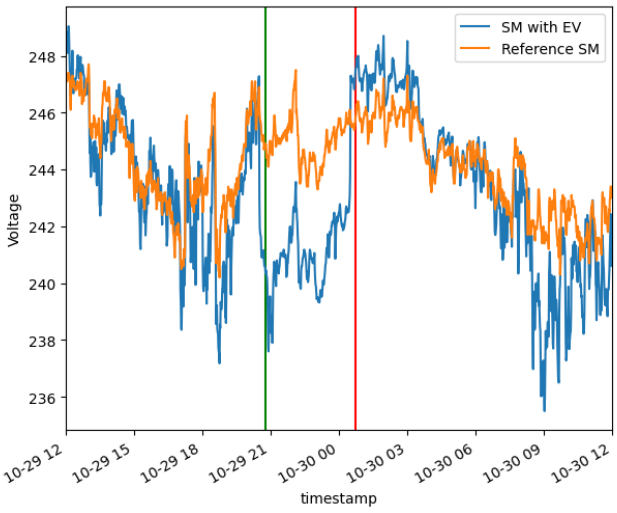












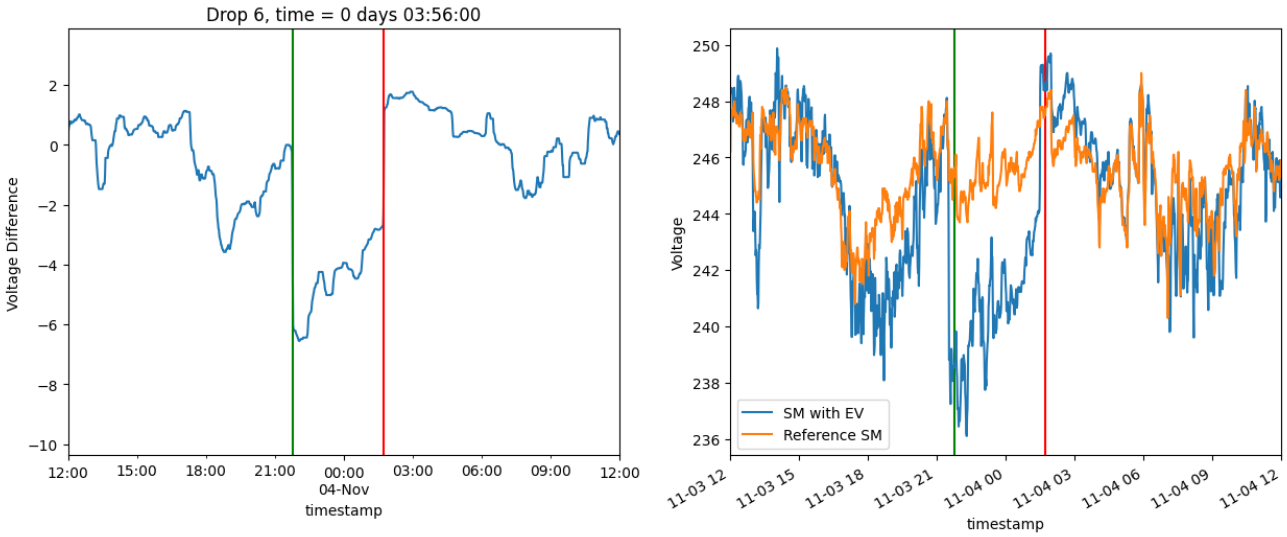


Figure 20: Voltage drops and their time identified by the algorithm

Using this drop detection algorithm, we configured it to find drops of more than 2.5V for longer than 100 minutes.

Subsequently, the algorithm creates some features (see Table 11) that can be used to train a model and separate the voltage drops that are caused by EVs, from the ones that are caused from other large loads.

Features	Description
Number of drops per compared SM	Taking the total number of drops detected for a SM and dividing by the number of SMs that the examined SM was compared to.
Average drop length (time)	The average number of minutes of each detected drop.
Percentage of compared SMs with drops detected	For a given SM, we take the number of compared SMs which we detected drops relative to our currently considered SM, and then divide this number by the number of compared SMs. For example, if we compare one SM to 6 others on the same phase and feeder and we detect drops relative to 4 of them, then this is 67%.

Table 11: Features used as input to the model.

Using some simple filters requiring drops to be detected consistently when comparing to multiple different SMs on the same feeder, and requiring the drops length to be longer than 100 minutes, we get the following confusion matrix:

		Predicted	
		True	**False**
Actual	**True**	31	17

	Unlabelled	468	1268

Table 12: Confusion Matrix

The model has a TPR of about 65% and an FPR of 27%, which is probably too high. It turns out that many properties that have not been recorded as EVs have been identified by the algorithm to have these voltage drops. This can be explained by the fact that other large loads with long duration could cause the same voltage impacts as EVs. However, we could use this approach to validate and reduce the number of the unlabelled properties that have been predicted as having an EV from the approaches in section 3.4.2.

Moreover, we need to note that the voltage drops caused by an EV could be small and very difficult to detect if the service cable impedance is low. Additionally, if the customer charges in lots of short sessions, the voltage drop may not last very long, making it harder to detect.

Additionally, we could use the features in Table 11 to train a ML model, but we need to extend the core area to receive voltage data from more known EVs, as the number of available voltage data of known EVs is small in the core area. We trained a SVM model but the results did not improve. Receiving more voltage data from known EVs could significantly increase the performance of the model and will enable us to train a more powerful time-series model (i.e., Deep Learning).

Moreover, the features in Table 11 could be used as extra attributes in the ML algorithms that were discussed in section 3.4.2. However, the number of properties that we have available voltage data, 12 months of consumption data and they are known to have EVs is very small in the core area in order to test this approach. Finally, we need to note that the improvements in phase and feeder data could improve the performance of the algorithm, as the reference data will be improved. Similarly, the performance of the model can be increased as more customers install a SM.

5 Summary of Learning Points

- HH Export data are very promising to identify PV and other types of generation.
- The EV detection through the closeness to median curve has a high TPR but it classifies many properties as having an EV. It cannot separate the properties with high consumption with the ones that have high consumption because of an EV existence.
- Weighted ML algorithms are very promising on identifying EVs (PU learning).
- The summer consumption and the maximum consumption features have higher impact on the models.
- Voltage data looks promising to identify EVs due to big drops they caused in the voltage, but further investigation is needed to improve the algorithm's performance.
- EV can cause significant drops in voltages at times when an EV car is getting charge.
- Subtracting the GridKey voltage data can remove the common voltage variations on the phase that an SM is connected to, but subtracting other SMs' voltages on the same feeder and phase was more effective at removing common voltage variations.
- EVs are not the only the only things causing voltage drops over 100 minutes, but we can train a model to identify which properties have EVs based on frequency of drops, average drop lengths, and percentage of other SMs on the same feeder that the algorithm detects drops relative to.

Next Steps

- An ML algorithm can be created using HH AE data and other relevant datasets such as weather data to identify the different types of generation that are present at a property. Therefore, it is crucial to extend the core area and obtain export data from other generator types such as batteries and wind turbines to facilitate the development of such an algorithm.
- Continue the analysis in voltage data for known PVs to try and identify if there are any voltage changes caused by PVs.
- Investigate the consumption patterns for MPANs that have Heat Pumps.
- Identify trends in customer EACs before and after an EV charger/Heat Pump/ PV is registered.

6 Appendix

6.1 Logistic Regression

Logistic regression is one of the most basic machine learning models. It uses the following formula to make predictions:

$$\ln \frac{p}{1-p} = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

where the x_i are the features of a given sample, p is the probability that the sample is positive, and the a_i are learned parameters. Given a trained logistic regression model (where we have the a_i values), we can calculate a probability that a given sample is positive by using the x_i feature values for that sample and then solving for p. The training process is where we provide a set of labelled positive and negative samples, and it then calculates optimal a_i that makes the model generally predict high probabilities for positive samples and low probabilities for negative samples, as best as it can.

In the PU literature, weighted logistic regression is one of the models that can be used for PU learning tasks. The way that the 'weighting' works is by treating the unlabelled samples as negative, but the training algorithm penalises false positives (negative samples predicted as positive) less than false negatives (positive samples predicted as negative). Because we expect that the unlabelled set will contain some positive samples, we don't want to the training process to penalise these much. After the model has been trained, the weighting does not need to be considered anymore, as it only affects the training process.

According to a paper we read in the PU literature, we should weight the positive samples by the probability that a randomly picked training sample is negative, and the negative samples should be weighted by the probability that a training sample is positive. In our case, we were expecting around 3% to 5% of properties to have EVs in the core area. However, we also requested data on LCTs outside of the core area, so we have a noticeable sample bias that means that taking 5% as the probability of EV is not the most optimal.

We implemented this using scikit-learn's `sklearn.linear_model.LogisticRegression` class.

6.2 Support Vector Machine (SVM)

Support Vector Machines are supervised machine learning models which are mainly used for classification. They are well known for being a robust and effective model for many classification tasks. The model takes in the training data in an n-dimensional space and tries to find the best hyperplane that separates the positive samples from the negative samples. Points that lie on one side of this hyperplane are classified as positive, and any points on the other side are classified as negative. While a simple SVM tries to find a linear hyperplane, it can be configured to use a non-linear kernel function that allows the SVM to find non-linear relationships between the features and target variable, effectively creating a non-linear classification boundary which can capture more complex patterns in the data.

While support vector machines normally only provide binary predictions (on the positive or negative side of the classification boundary), we use a technique called "Platt scaling" to make the SVM output probabilities. This is built in to scikit-learn by specifying `probability=True` when creating the model.

The weighted SVM is one of the models mentioned in the PU learning literature. The 'weighting' is only used within the training process to bias the model towards learning to detect positive samples, and it is not used after a model has been trained. By weighting the positive samples to be more important than the negative/unlabelled

samples, the training algorithm penalises false positives (unlabelled samples predicted as positive) less than false negatives (positive samples predicted as negative).

SVMs are more powerful than logistic regression models because they can capture more complex patterns in the data while retaining the ability to generalise.

In scikit-learn, we used the `sklearn.svm.SVC` class to build our SVMs.

6.3 XGBoost

XGBoost is an open-source library which implements gradient boosting models, in which several decision trees are trained and combined into a single model. XGBoost is known for its strength in machine learning competitions, being frequently used to win many of these competitions. While it is not designed to be used with PU learning, we experimented with it to see if it could do any better than the models mentioned in the PU literature.

6.4 ROC Curve

A Receiver Operating Characteristic curve, or ROC curve, is a plot that illustrates the trade-off between true positive rate and false positive rate as the decision threshold is varied.

All of the models we have used in our analysis can output each prediction as a number between 0 and 1 which can be considered as a probability that a given sample is positive. Using these probabilities, we can specify a 'decision threshold' which classifies any prediction above the threshold as positive and anything below it as negative.

	Truth Label	Model probability	Classification at 0.3 threshold	Classification at 0.7 threshold
Sample 1	TRUE	0.9	TRUE	TRUE
Sample 2	FALSE	0.2	FALSE	FALSE
Sample 3	FALSE	0.4	TRUE	FALSE
Sample 4	TRUE	0.4	TRUE	FALSE
Sample 5	FALSE	0.6	TRUE	FALSE
Sample 6	FALSE	0.2	FALSE	FALSE
Sample 7	FALSE	0.8	TRUE	TRUE
Sample 8	TRUE	0.8	TRUE	TRUE

Confusion matrices:

0.3 Threshold

		Predicted: TRUE	Predicted: FALSE		
Actual	TRUE	3		TPR =	100%
	FALSE	3	2	FPR =	60%

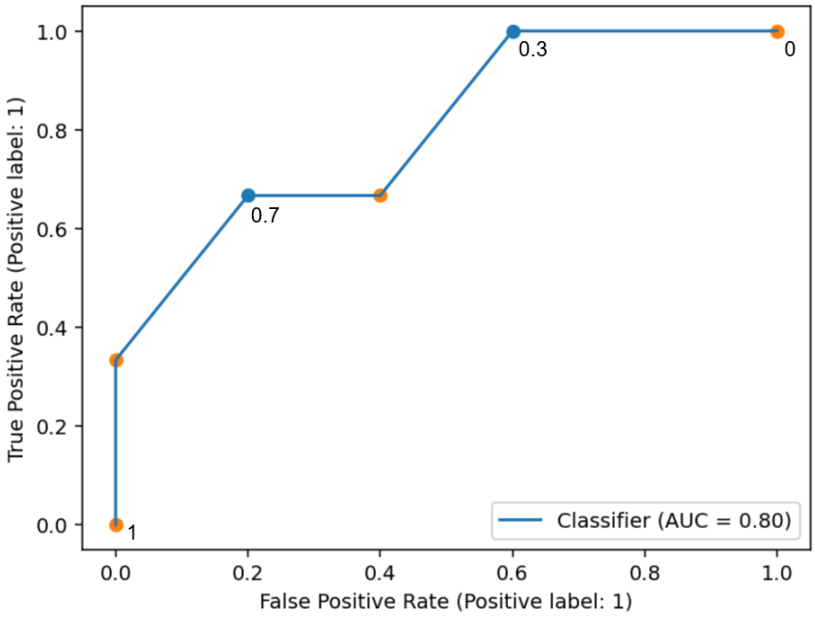
0.7 Threshold

		Predicted: TRUE	Predicted: FALSE		
Actual	TRUE	2	1	TPR =	67%
	FALSE	1	4	FPR =	20%

We can see that when we set the threshold at 0.3, all positive samples are predicted correctly, however lots of the negative samples are also predicted true, giving a false positive rate of 60%. When setting the decision

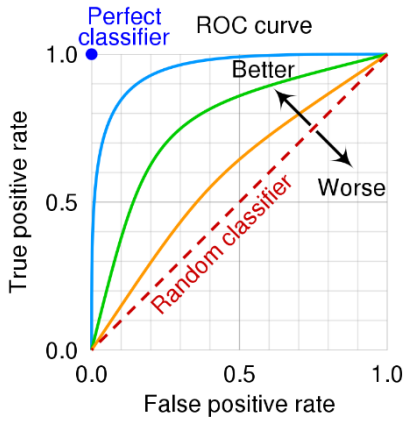
threshold at 0.7, there are way fewer negative samples predicted as positive, but there are also fewer positive samples predicted positive. This is the trade-off between TPR and FPR.

To illustrate this visually for all possible thresholds, we use a ROC curve. This plots the TPR and FPR at various decision thresholds. For our example above, here is the ROC curve:



When the decision threshold is 1, all samples are classified as negative, so both the TPR and FPR are 0. When the decision threshold is 0, all samples are classified as positive, so both the TPR and FPR are 1. The points where the classification threshold are 0.3 and 0.7 are also labelled in the plot. In the bottom right, it shows "AUC = 0.80", which means that the "Area Under Curve" is 0.8. The AUC is a useful metric for evaluating models which have a heavy class imbalance (like our EV case, where less than 10% of samples are positive). This class imbalance is why we don't use accuracy as our evaluation metric, because if we have 95% of samples being negative, a classifier that predicts everything as negative will have 95% accuracy (despite it being a rubbish model).

In general, we want our classifier's ROC curve to reach as close to the top left as possible, where TPR = 1 and FPR = 0, which would be a perfect classifier. Here is an illustration from Wikipedia:

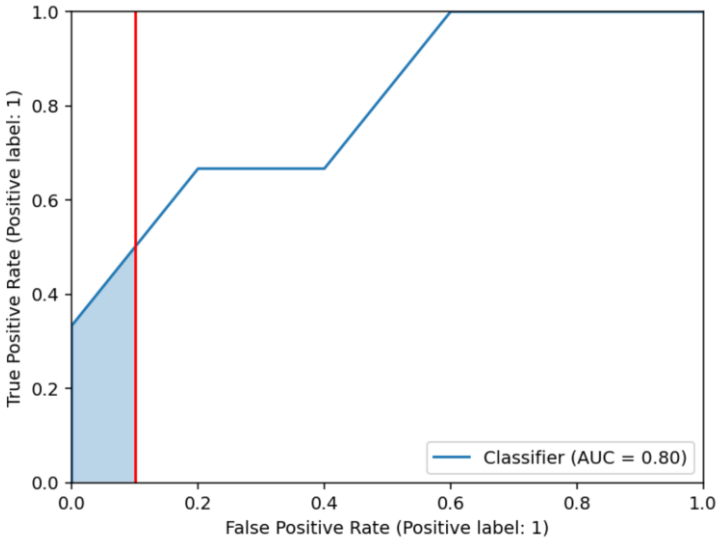


https://en.wikipedia.org/wiki/Receiver_operating_characteristic#/media/File:Roc_curve.svg

6.5 Partial AUC

Using the ROC curve, we can take the area under it as an evaluation metric, the AUC. However, using this as an evaluation metric includes classification thresholds which are of no use to use because they are way too low. If we have an FPR value of 0.2, we are predicting 20% of the unlabelled set as EVs which is way too high. In our case, it makes much more sense to limit the area of the ROC curve that we are interested in to the area with FPR < 0.1, so we are looking at much more reasonable classification thresholds, as we do not care how a model does when the FPR is above 10%.

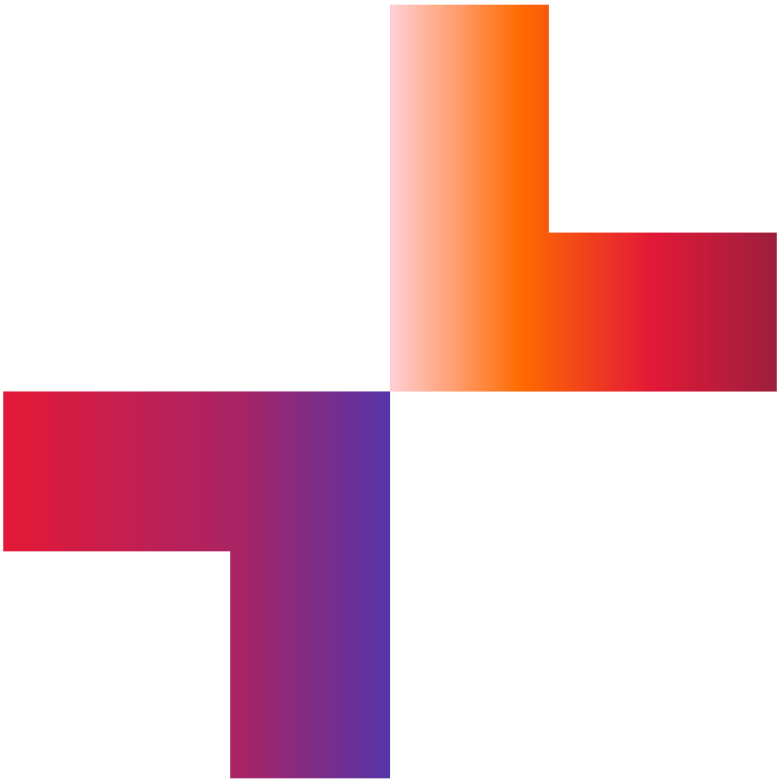
The partial AUC is the area under the ROC curve, but just limited to our desired range of FPR values. Using the example in the previous section on the ROC curve, here is the relevant area where FPR < 10%:



This shaded region has area = 0.041666…, but after this we scale this value to between 0 and 1 using the formula (where e is the max FPR, 0.1)

$$\widetilde{A}_e = \frac{1}{2}\left(1 + \frac{A_e - e^2/2}{e - e^2/2}\right) = \frac{1}{2}\left(1 + \frac{\int_0^e ROC(f)df - e^2/2}{e - e^2/2}\right)$$

Which gives us the ‘standardised partial area under curve’ (in our example, this is 0.6930).
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3744586/



cgi.com