FIT5149 S1 2022 Assessment 2: Electric Vehicle Charging Event Classification

April 2022

Marks	35% of all marks for the unit
Due Date	11:55 PM Friday, 27 May 2022
Extension	An extension could be granted for circumstances. Please refer to the university webpage on special consideration. A special consideration application form must be submitted. Please note that ALL special consideration, including within the semester, is now to be submitted centrally. All students MUST submit an online special consideration form via Monash Connect.
Lateness	For all assessment items handed in after the official due date. Without an agreed extension, a 10% penalty applies to the student's mark for each day after the due date (including weekends and public holidays) for up to 5 days. Assessment items handed in after 5 days will not be considered/marked.
Group Member	This assignment is a **group assignment**, and you are highly encouraged to form **a group with three members, including yourself**. A group with more than three members (say >= 4) is not allowed. You may complete this assessment by yourself or form a group with two members, if you have difficulty forming a group.
Authorship	The final submission must be **identifiable your group's own work in the group**. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary actions.
Submission	Each group is required to submit two files, one PDF file contains the report, and another is a ZIP file containing the implementation and the other required files. The two files must be submitted via Moodle. All the group members are required to log in to Moodle to accept the terms and conditions on the Moodle submission page. A draft submission won't be marked.
Programming language	Either R or Python

Note: Please read the description from the start to the end carefully before you start your work! Given that it is a group assessment, **each group should evenly distribute the work among all the group members**.

1 Introduction

Background: Burning fossil fuels, such as gasoline and diesel, releases carbon dioxide (CO2), known as a greenhouse gas, into the atmosphere. The buildup of CO2 causes the Earth's atmosphere to warm, resulting in climate change. Greenhouse gas (GHG) emissions from transportation account for about 29% of the total greenhouse gas emissions, according to the U.S. Environmental Protection Agency[1].

Research has shown that electric vehicles (EVs) are better for the environment. They emit fewer greenhouse gases and air pollutants than petrol or diesel cars. And this takes into account their production and electricity generation to keep them running[2]. The Climate Council is calling on Australian state and territory governments to further incentivise EV uptake, including target 100% of new car sales and trucks to be EVs by 2030 or earlier and commit to all new government fleet vehicles being electric immediately, with the electrification of the entire fleet before 2030[3].

The increasing uptake of EVs and their charging demand pose challenges to today's energy supply systems, causing unexpected peak load and voltage problems in the energy distribution network. There is a growing interest in understanding when EVs are charged, such that the Australian energy system operator and energy service providers can get better prepared to host a large number of EV charging in the system. However, it is challenging to keep track of EV charging at home at scale and in a cost-effective way. The system operator and service providers do not know how EV owners charge their EVs.

Problem Setup: Thanks to the wide adoption of smart meters (here in Victoria and worldwide), the household energy consumption is recorded in real time (say every one minute), as shown in Figure 1. The smart meter records the household's total energy consumption, including EV charging and other appliances' energy use (such as oven, light, heater, and many other appliances) at home. With smart meter data collected, this sheds light on using data analysis techniques to detect EV charging (or we call it EV charging event classification) from the smart meter data, providing an effective yet non-intrusive solution method to understand when EVs are charged.

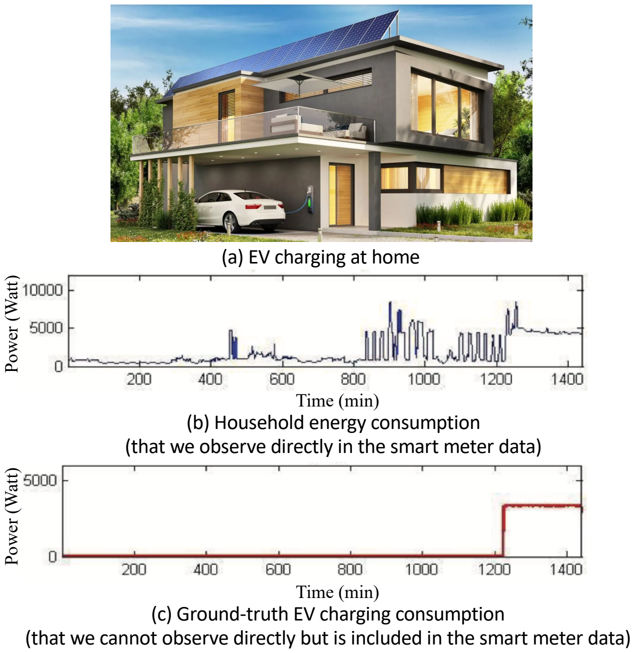

Figure 1: Illustration of (a) the EV charging at home, (b) household total energy consumption recorded by a smart meter, and (c) EV charging demand. Note that this is just an illustration, and the real-world EV charging pattern could be less straightforward but more complex.

[1] https://www.epa.gov/transportation-air-pollution-and-climate-change/carbon-pollution-transportation
[2] https://www.edfenergy.com/for-home/energywise/electric-cars-and-environment
[3] https://www.climatecouncil.org.au/electric-vehicles-electric-buses-can-keep-australia-moving/

Load monitoring (also known as load detection and load disaggregation) is a promising data analysis technique to provide detailed electricity consumption information of individual appliances or to detect appliance use. Take EV charging as an example, in Figure 1, though we do not know the EV charging demand or when EV is charged, we can infer from the smart meter data (recording the household total energy consumption) to classify the EV charging events (when EV charging occurs). This data analysis technique also applies to other residential appliance use event classification, but we will focus on EV charging events only in this assessment.

Here is optional reading about a recent review of load monitoring entitled "Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools-A review"[4]. Note that you are able to understand the task by reading this assessment description without reading this optional material. On the other hand, you can also find other references on load monitoring online yourselves through Monash University Library database subscriptions.

In this assessment, you are given household total energy consumption (i.e., smart meter reading) with EV charging event labels (i.e., whether EV is being charged or not, regardless of the charging rate). This is a typical classification problem, and there are many machine learning methods (such as logistic regression, SVM, and tree-based methods you learned/will learn in this unit) that can be used in this classification task. In the training phase, the inputs are labels and extracted features, and the response variables are EV charging events (i.e., being charged or not). Figure 2 shows a typical framework used in the supervised classification.[5] As shown in Figure 2, there are three major steps: generating features, developing a proper classifier, and applying the classifier to the unseen data. The feature extractor is shared by both training and prediction, which tells us that data used in training and prediction should share the same feature space.

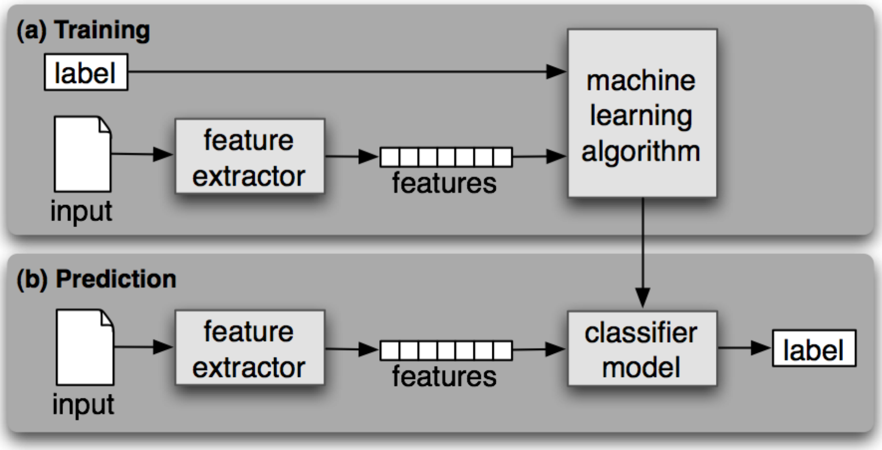

Figure 2: A general framework for the supervised classification.

The aim of this challenge is to develop different classifiers for EV charging events, e.g., to detect whether the EV is being charged in each time interval (e.g., 1-minute interval) as correctly as possible. The evaluation emphasises on the performance and interpretability of the classifiers you develop. Specifically, you are expected to

- **develop three different classifiers** to detect EV charging events as accurate as possible;
- and **explain the outcomes** of your classifiers. For example, what features are strongly associated with the response variables and effective in the classification task.

2 Dataset

We provide the following data sets (Table 1).

[4]The link to the paper is https://onlinelibrary.wiley.com/doi/epdf/10.1002/widm.1265
[5]The figure is download from https://www.nltk.org/book/ch06.html

Data Source	Type	Training labels	# training examples	# test examples
Dataport	Energy consumption	EV charging	∼419,808	∼104,952

Table 1: One-minute interval residential smart meter data with EV charging labels (in the training set only).

- train_data_withlabels.csv contains ∼419,808 instances, each of which has 10 columns, including 'dataid', 'localminute', 'total_load', 'ab_diff', 're_diff', 'window_mean', 'window_dev', 'window_crossing_point', 'window_peak', and 'target' (which represents the EV charging labels). Note that 'dataid' is the ID of the household in the pilot study, and you can find that this dataset is from one household; 'localminute' is the timestamp; 'total_load' is the total energy consumption of the household; for the labels in 'target', 1 indicates EV charging, and 0 indicates no EV charging. The remaining columns, such as 'ab_diff', 're_diff', 'window_mean', 'window_dev', 'window_crossing_point', 'window_peak' are sample features (we provide to you for training) extracted from 'total_load'. We will introduce how to extract these features as an example and provide the R code to you. All the load data and features are minute-by-minute readings in a sequential manner.

- test_data_nolabels.csv: It contains ∼104,952 instances, each of which has 9 columns, including the same columns as those in train_data_withlabels.csv but no labels.

Note: The timestamps 'localminute', total energy consumption 'total_load', and labels 'target' are the original data. Other attributes/features are all extracted from the load data using various measures[6]. These sample attributes are optional for potentially improving the performance. You should be able to achieve reasonably good performance using the provided features. But any information about the test data cannot be used for training the classifiers.

In addition, if you are interested, you can extract more/different features and evaluate their effectiveness in the classification task.

3 Feature Extraction

3.1 Introduction to The Sample Features (with Code Provided)

In the provided data sets, a few features have been provided. The following is a brief introduction to the features in train_data_withlabels.csv and test_data_nolabels.csv. For each load data point (per 1 minute), we define a neighborhood (consecutive) time window with a radius of $R = 50$ minutes around the data point. Using the load data in the time window (with $R + 1 + R$ data points, meaning R data points on the left plus 1 current data point plus R data points on the right), we calculate several statistical values as features. Note that these are sample features, and you can follow a similar approach and extract additional or different features, as introduced in Section 3.2.

- 'ab_diff' – The absolute value of the difference between the first data point (at the boundary on the left) and the last data point (at the boundary on the right) in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.

- 're_diff' – The difference between the first data point (at the boundary in the left) and the last data point (at the boundary in the right) in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.

- 'window_mean' – The mean value of the total load in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.

- 'window_dev' – The standard deviation of total load in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.

[6] A brief description and some hints will be provided in Section 3.

- 'window_crossing_point' – The number of intersections of the total load curve cross the mean line (averaged total load) in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.
- 'window_peak' – The maximum value (i.e., peak) of the total load in a neighborhood time window with a radius of $R = 50$ minutes around each load data point.

Note: We also provide the R code **"feature_extraction_example_r.ipynb"** extracting the above features for your reference. (Note that you can choose another radius R to define the neighborhood. We recommend you choose R no greater than 360, meaning 6 hours. For any chosen R, it is natural that the first R data points and last R data points in the dataset do not support such a window/neighborhood. You can either mark the corresponding features as zero or drop these two segments of data, which are negligible and won't affect the performance as you can extract features for the vast majority of instances in the dataset.)

3.2 Hints on Feature Extraction (Optional)

Selecting relevant features and deciding how to encode them for a classification algorithm could be crucial for learning a good model. The sample features provided to you have been verified by the teaching team to be useful in training the model. However, you are always welcome and encouraged to try different and/or more effective features.

- Since EV charging may last for a while (e.g., from several minutes to even a few hours), you may consider a neighborhood time window around each load data point, as the sample feature extraction showed. You can treat the window size as a hyper-parameter to tune and find the optimal size to extract features.
- In addition to the sample features provided, you can use other functions/measures/packages to extract different features. You are free to use any functions/measures/packages for feature extraction as long as you clearly provide the reference(s).
- There are many useful online tutorials on sequential data in either R or Python, for example,
  - Introduction to the tsfeatures R package for sequential/time-series characteristics and installation tutorial [7]
  - Introduction to the tsfresh python package for sequential/time-series characteristics and and installation tutorial[8]
  - Feature extraction in Scikit-learn[9]

Note: Once you define the window size (either follow the sample example or choose your own), you can apply functions and packages to each window and quickly extract the corresponding features. The teaching team has already provided sample examples (with R code) to show how to define a neighborhood time window and extract features on sequential/time-series data. You are not expected to know any prior knowledge about sequential/time-series data when working on this project.

4 Task Description

In this assessment, you will focus on the following two tasks:

- **Prediction Task** – You are expected to develop classifier(s) that can give you the most accurate prediction in this EV charging event classification task. The algorithms/models that you can use are not limited to the algorithms/models covered in the lectures/tutorials, though we encourage you to try what we covered in this unit first. The goal is to find the most accurate classifiers for this task.

[7] https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html
[8] https://tsfresh.readthedocs.io/en/latest/
[9] https://scikit-learn.org/stable/modules/feature_extraction.html

- **Inference Task** – The purpose of the inference task is to identify the key features that have strong effects on the prediction results, indicating strong drivers of EV charging events.

4.1 Prediction Task

In order to find the most accurate classifiers, each group should empirically compare at least three different types of classifiers for this classification task, and then choose the one performing the best to participate in the challenge. Please note an algorithm with different input features is regarded as one type of classifier. For example, logistic regression will be counted as one type of classifier, no matter what features you use. Logistic regression, SVM, Random Forest, and XGBoost (using the same features) can be regarded as different types of classifiers.

The performance of the classification task not only depends on the classifier but also on the features. As introduced in Section 3, a set of features have been provided to you for evaluation. You can also try different/more effective features extracted by yourselves.

Note: You are asked to **submit** the numerical results (on the prediction) using the best-performing classifier you believe. But you are also asked to **report** at least three different types of classifiers you tried in the group report.

4.2 Inference Task

Inference can be based on variable correlation analysis, regression equations, or any other form. To finish this task, you are expected to use proper data analysis techniques to identify a subset of attributes that have a significant impact on the appliance usage. **Report** your identified features with statistical evidence and interpret the attribute subsets.

4.3 Group Forming

This is a group-based assessment, and we will create "Assessment 2 group selection" on Moodle. Please select an ID for your group and make sure all the members join the same group.

Note:

- You are free to form a group of up to 3 members from any tutorial classes. In other words, there is NO restriction on whether the group members are from the same tutorial class.
- The recommended number of group members is 3 (but no more than 3 strictly). A smaller group (1 and 2 members) is allowed under special circumstances.

4.4 Kaggle Data Competition (not marked)

We also launch a Kaggle Data Competition for this assessment, and you can find it through the link. You can get some feedback on the predictions your model produces and compete ('work') with your classmates to create better models and predictions. Although your Kaggle results will not be marked, we highly encourage every group to participate in the competition. It will be a lot of fun!

Note: The Kaggle results will not be marked. Please see Section 6 for more information about the required submission and also check the marking rubrics.

4.5 Some Hints

There are some hints that we summarise based on the past submissions made by previous cohorts.

- Avoid using the absolute path in your Jupyter notebook. Instead, use a relative path (.e.g, "./train_data_withlabels.csv") and place the data file in the same folder where your Jupyter notebook is.
- Avoid just plainly showing the results without meaningful interpretation/discussion (this has been stressed multiple times in both tutorials and lectures). For example, if you use any plot, you will need to clearly discuss the information delivered by the plots in the context of the task.

- Choose the appropriate plots or statistics to show the right information.
- While developing the model, make clear, for example, how the optimal parameters are chosen if there is any, etc.
- Be precise in the use of various tools and the corresponding discussion.
- Avoid submitting an extremely long Jupyter notebook, which could result in a lot of redundant information, easily losing the focus of your work.
- Make sure the logic (or the methodology) you used to develop the model is properly documented.
- Write your discussion using the markdown cells and avoid putting it in the code cell as we will use this to gauge your reasoning skills.
- Make use of the discussion forum and consultations to clear any doubts that you may have regarding the tasks you want to accomplish.
- Before making the final submission, you must make sure that your Jupyter Notebook runs without any errors. A simple step is to click "Kernel → Restart & Run All".

5 Evaluation

The evaluation metric used in test is the F1 score, which is defined as

$$\text{F1} = 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

where

$$\text{Precision} = \frac{\textbf{Number of True Positive}}{\textbf{Number of True Positive and False Positive}}$$

and

$$\text{Recall} = \frac{\textbf{Number of True Positive}}{\textbf{Number of True Positive and False Negative}}.$$

You can use the existing python/R code to compute F1, precision, and recall, for example

- F1 score in Python[10], Precision in Python[11], and Recall in Python[12],
- F1 score in R[13], Precision in R[14], and Recall in R[15].

6 Submission

To finish this data analysis challenge, all the groups are required to submit the following files:

- **"pred_labels.csv"**, where the label prediction on the testing documents is stored.
  - In your "pred_labels.csv", there will be two columns: the first one is the **column index from** 1 **to the number of total load data points**. The second column should include **predicted results** (0 **or** 1) for EV charging event classification.

[10] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html?highlight=f1#sklearn.metrics.f1_score
[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html#sklearn.metrics.precision_score
[12] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html#sklearn.metrics.recall_score
[13] https://www.rdocumentation.org/packages/measures/versions/0.3/topics/F1
[14] https://www.rdocumentation.org/packages/Metrics/versions/0.1.4/topics/precision
[15] https://www.rdocumentation.org/packages/Metrics/versions/0.1.4/topics/recall

- The "pred_labels.csv" must be reproducible by the assessor with your submitted R/Python code.

- The **R/Python implementation** of your **final** classifier with A README file that tells the assessor how to set up and run your code. The output of your implementation must include the label prediction (see the above description about **"pred_labels.csv"**). The use of Jupyter notebook or R Markdown is **not required**. All the files that are required for running your implementation must be compressed into a **zip** file, named as "**groupName_ass2_impl.zip**". Please note that the unnecessary code must be excluded in your final submission. For example, if you tried three different types of models, say logistic regression, SVM, and classification tree, and your group decides to submit classification trees as the final model. You should remove the code for the other models from the submission. **The discussion of the comparison should be included in your report.** *However, you should keep a copy of the implementation used for comparison for the purpose of the interview if scheduled.*

- **A PDF report**, where you should document in detail the development of the submitted classifier. **The maximum number of pages allowed is 8**. The report must be in the PDF format, named a **"groupdName_ass2_report.pdf"**. The report must include (but not limited to)

  - The discussion of how the data preprocessing/features selection has been done.
  - The development of the submitted classifier: To choose an optimal classifier for a task, we often carry out empirical comparisons of multiple candidate models with different feature sets. In your report, you should include a comprehensive analysis of how the comparisons are done. For example, the report can include (but not limited to)
    - A description of the classifier(s) considered in your comparison.
    - The detailed experimental settings, which can include, for example, the discussion of how the cross-validation is set up, how the parameters for the model considered (if applicable) are chosen, or the setting of semi-supervised learning (if applicable).
    - Classification accuracy with comprehensive discussion.
    - The justification of the final model submitted.

  Warning: If a report exceeds the page limit, the assessment will only be based on the first 8 pages.

- A signed group assignment cover sheet, which will also be included in your zip file.
  Warning: typing name is not counted as a signature in the cover sheet.

7 How to submit the files?

The Moodle setup allows you to upload only two files

- "**groupdName_ass2_report.pdf**": A pdf report file, which will be submitted to Turnitin.
- "**groupName_ass2_impl.zip**"': a **zip** file includes
  - the implementation of the final submitted model
  - "predict_label.csv", where the label prediction on the testing documents is stored.
  - the signed grouped assignment cover sheet

While submitting your assignment, you can ignore the Turnitin warning message generated for the ZIP file.

Please note that

- **Only one group member needs to upload the two files. But all the group members have to login in to their own Moodle page and click the submit button in order to make the final submission.** If any member does not click the submit button, the uploaded files will remain as a draft submission. A draft submission won't be marked.
- **The two files must be uploaded separately.**

8 Academic integrity

Please be aware of University's policy on academic integrity. Monash University takes academic misconduct[16] very seriously. You can learn from the above materials and understand the principle of how the analysis was done. However, you must finish this assessment with your own work.

[16] https://www.monash.edu/students/study-support/academic-integrity