GridKey Project Report

Electric Nation Functional Requirements Document and Close Down Report

GK11000004 – Issue 1.3

A collaboration between Lucy Electric and Sentec

Project Signatures

Project Name	Electric Nation
Project Sponsor	Mark Dale (Western Power Distribution)
Project Manager	Craig Holahan (Lucy Electric GridKey)
Date	March 2018

Prepared By

Document Owner	Craig Holahan, Paul Beck, Chris Hole (TTP)
Organisation Role	Project Manager, Director Smart Systems, Manager
Document Revision	Issue 1.3
Date	June 2018

Contents

1. Introduction

This report provides an overview of the project activities carried out as part of the Western Power sponsored Electric Nation project by Lucy Electric GridKey Ltd and its subcontractor – The Technology Partnership (TTP).

The problem statement was to determine whether an EV charging signature could be automatically detected though advanced data analytics when embedded in the other typical electrical loads seen on a substation feeder supplying domestic properties.

2. Project Background Overview

The increasing growth rate of Electric Vehicle (EV) sales and usage has led to a corresponding increase in the number of charging points through Government sponsored initiatives such as Plugged-in Places and the Homecharge Scheme. Fast charging points are now being installed in many commercial and public places (for example shopping centres, car parks, on-street parking etc.) and whilst these will undoubtedly be used, EV users will still want the ability to charge their vehicles at home. This will cause a substantial increase in the demand on the network both at Medium Voltage (11,000 volts and above) and Low Voltage (LV, 230/415volts).

The My Electric Avenue project (http://myelectricavenue.info/about-project) highlighted the problem home charging may cause on the LV Distribution network – the final report from that project stated: "Across Britain 32% of local electricity networks (312,000 circuits) will require intervention when 40% - 70% of customers have EVs" – this analysis was based on low power (3kW) chargers supplied with the first generation Nissan Leaf. The latest vehicles are now being supplied with larger (7kW) single phase chargers so the size of the problem will increase.

The combination of these growth factors produces a significant and potentially unpredictable load on the electricity network. The nature of this additional strain is exacerbated by the fact that vehicles, being mobile, are not restricted to where or when they may be charged. Although EV registration and home charger installation data is available to Distribution Network Operators (DNOs), due to the variabilities in the use of the chargers, this is not adequate for accurate modelling and subsequent planning of reinforcement to maintain operation of the network.

3. Functional Requirements

The intention of this project is to develop a system using data from LV monitoring devices coupled with a specially developed analytics algorithm to inform the planning decisions for network reinforcement as well as potentially providing an input to a future Active Network Management solution.

The requirements for the solution were:

- To use a commercially available LV Monitoring system to collect the required data
- To collect the data for analysis on a secure Data Centre platform
- To develop algorithms to be able to detect that a circa 7kW electric vehicle charger had been switched on to start charging

- To be able to detect that the circa 7kW electric vehicle charger had either been switched off or had completed its charge cycle
- As a stretch target to be able to detect the number of vehicles and the type or family of vehicle types being charged
- That the detection should be able to detect vehicles charging with the normal background electrical noise associated with a variety of type and number of domestic dwellings

4. Methodology

4.1 Overview of Planned Approach

A decision was required at the start to determine whether the processing of the data would be done locally at the substation or at the head end (i.e. the Data Centre). Due to the level of processing expected to be required to carry out this analytics and that there was no need for that real time information to be available at the substation the decision was made to target the algorithms at the Data Centre.

Based on this, the top level method planned was:-

1. Collect sample signatures from as wide a variety of vehicles (pure EV and hybrids) as possible and starting and finishing in different states of vehicle charge
2. Once collected these signatures could be analysed to look for similarities/differences and also for any common characteristics particularly at switch on and completion of charging and then the development of an algorithm to look for these characteristics
3. Synthetically combine these sample signatures with a range of different profiles from LV monitoring
4. Test the algorithm on these combined load profiles to determine its effectiveness

4.2 Method Used

4.2.1 Collection of Sample Signatures

The initial step was to study and understand the charging profile of a variety of vehicle types. This work was done in parallel with the WPD EV Emissions Testing study in which the Power Quality signature (specifically the harmonic content) of a variety of vehicle types were measured and analysed. This project was run at the Millbrook Proving Ground where a selection of EVs were charged using a dedicated charging point – this consisted of 4 single phase chargers on a separate three phase supply.

A GridKey LV monitoring system was installed alongside the Power Quality meters at Millbrook and set to record the voltages and current on each of the chargers.

Figure 1 – LV Monitoring Installation at Millbrook Proving Ground

The measurements were set to a 1 minute reporting period and parameters captured included mean, min and max currents and voltages, real and reactive powers and THD. These parameters were reported back to and stored on the GridKey Data Centre via a secure GPRS link and a weekly CSV file produced and emailed to TTP.

Millbrook then provided a weekly log on which identified type, date, time, start charge condition, end charge condition for all vehicles which had been charged in the previous week. This was then compared to the CSV such that a library of charge profiles has been generated.

This has extended the work done on My Electric Avenue to create a library for different vehicle types that can be used for other projects in the future.

A complete list of vehicle profiles created is as follows:

Vehicle List

Vehicle No.	Vehicle Make	Vehicle Model	Arrival Date	SoT Date	EoT Date	Departure Date
1	Mercedes	B250e	18/04/2017	19/04/2017	21/04/2017	26/04/2017
2	BMW	330e	27/02/2017	27/02/2017	02/03/2017	13/03/2017
3	VW	Passat GTE	08/05/2017	08/05/2017	09/05/2017	12/05/2017
4	BMW	i3 REX	13/03/2017	13/03/2017	16/03/2017	23/03/2017
5	BMW	i3 BEV	14/06/2017	19/06/2017	21/06/2017	21/06/2017
6	BMW	i8	29/03/2017	30/03/2017	30/03/2017	07/04/2017
7	Tesla	Model X	05/07/2017	06/07/2017	12/07/2017	14/07/2017
8	Nissan	Leaf Acenta	10/04/2017	11/04/2017	12/04/2017	24/04/2017
9	Nissan	Leaf Tekna	03/05/2017	03/05/2017	05/05/2017	17/05/2017
10	Nissan	eNV200	06/03/2017	08/03/2017	09/03/2017	20/03/2017
11	Kia	Soul	05/07/2017	06/07/2017	11/07/2017	14/07/2017
12	Peugeot	Ion	03/05/2017	17/05/2017	18/05/2017	30/05/2017
13	Renault	Kangoo Mk1	27/02/2017	27/02/2017	10/03/2017	13/03/2017
14	Renault	Kangoo Mk2	06/03/2017	07/03/2017	10/03/2017	20/03/2017
15	Renault	Zoe	10/04/2017	11/04/2017	13/04/2017	25/04/2017
17	Mitsubishi	Outlander Gx4	20/03/2017	20/03/2017	22/03/2017	03/04/2017
18	Volvo	V60	03/04/2017	04/04/2017	05/04/2017	07/04/2017
19	VW	Golf	20/03/2017	21/03/2017	23/03/2017	31/03/2017

20	Tesla	Model S	17/05/2017	18/05/2017	27/05/2017	30/05/2017
21	Hyundai	Ioniq EV	03/04/2017	05/04/2017	11/04/2017	18/04/2017
22	Kia	Soul	13/03/2017	14/03/2017	16/03/2017	27/03/2017
23	Kia	Optima	24/04/2017	25/04/2017	26/04/2017	03/05/2017

Table 1 – EV car types for which charging profiles have been captured

4.2.2 Development of Detection Algorithm

Inspection of the raw charging profiles from the different vehicles showed that from one charge to another of the same vehicle type resulted in a repeatable profile however from one vehicle type to another there were differences – some were current limited and some were power limited (so when looking at a trace of power against time the power limited ones had a flat “charging period” whereas the current limiting types had variations due to voltage changes). Also the vehicles had a range of battery cell balancing – these varied depending on the state of the battery. In summary there were a lot of variations. Examples of vehicle charge profiles are shown in section 4 of this report.

We also looked at power, current and Total Harmonic Distortion (THD) in these raw data traces - particularly at the start of the charging cycle – there was a very small reactive power component and this was fairly constant throughout the charging cycle and there was no noticeable change in THD – certainly not something that could be detected when there were other electrical background loads/noise. We also repeated this looking at the 1-second data which is available from the GridKey unit and this did not show any specific features in the profiles.

A previous project, known as Project Galaxy, also looked at load profiling for certain types of load. This measured (on a 1 minute basis) a series of electrical parameters which were then compared to a standard profile, we had created by isolating the load from any background noise and then measuring the same parameters. In that case there were repeatable features, particularly in the switch on profile, which we were able to detect and then combine with other parameters, to create an algorithm that reliably identified this specific type of load.

Initially for the Electric Nation project we tried to use the same analytic techniques as used on Galaxy – traditional pattern correlation, probabilistic analysis etc. However, as the only real trigger was a 7kW load this increased the probability of false alarms to an unacceptable level. There are other similar sized loads of that type, for example electric showers which would trigger the analytics.

A completely new and alternative approach was therefore adopted using a neural network “self-learning” approach; this is shown in the diagram below:

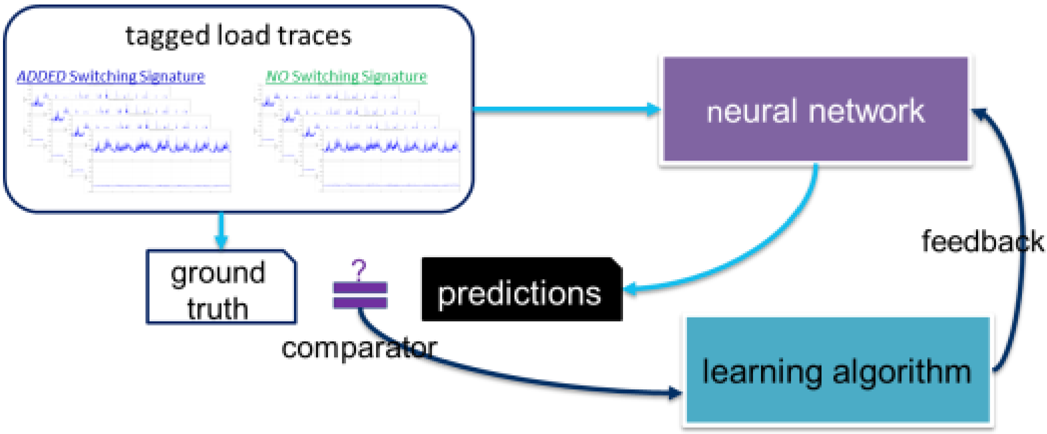


Figure 2 – Neural Network Approach

In order to further simplify the problem we limited the algorithm to try and detect a maximum of one EV charge switch on and one switch off event per hour and that both of these would be circa 7kW. This allowed the development of a multi-layered convolutional neural network algorithm which is a standard technique used for this type of analytics.

4.2.3 Combining with "Standard" load profiles

Obtaining suitable load profiles which did not have EV charging in them already was difficult. It was not known if there were any vehicle charging points on the particular feeders of the sample data or more particularly whether these charge points were being used. If there was an unknown charge event already happening this would skew the results. We were also limited to combine the EV sample data with data with the same reporting period (i.e. 1 minute).

To minimise the risk, we used some of the oldest (2013-2014) 1 minute data we had collected from other projects on the principle that there were few charging points/EVs in 2013. We also chose data from geographic areas which had low penetration of EVs to further reduce the risk. This resulted in a relatively small sub-set of suitable data however we were able to have a range of feeder loading levels which were all from residential areas. We also expanded this data set by artificially modifying the data to increase and decrease the background loads.

4.2.4 Algorithm Testing

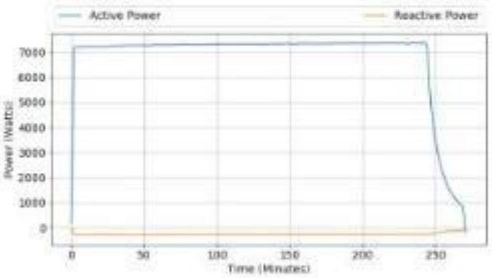
The algorithm was allowed to "learn" using the background data and then testing was carried out to both look for:

- **positive detections** (i.e. when there was a vehicle present)
- **false-positives** (i.e. when there was not a vehicle present but there was other electrical background noise which the algorithm mistook for a vehicle charging).
- **false negatives** (i.e. when a vehicle was present but the algorithm mistook it for background noise)

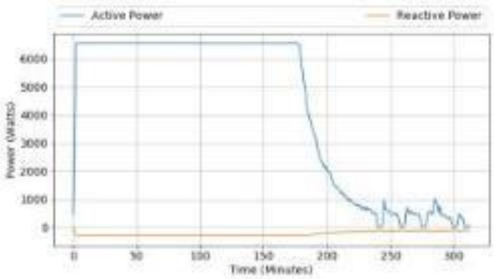
5. Results and Learning Outcomes

During the initial data gathering exercise at Millbrook a number of different vehicle profiles were collected and some examples are shown below:

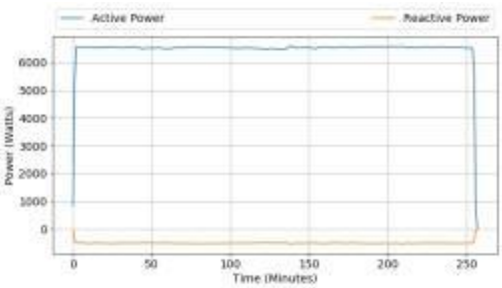
Vehicle #5 – BMW i3 REX



Vehicle #10 – Nissan Leaf Tekna



Vehicle #23 – Kia Soul



Vehicle #16 – Renault Zoe

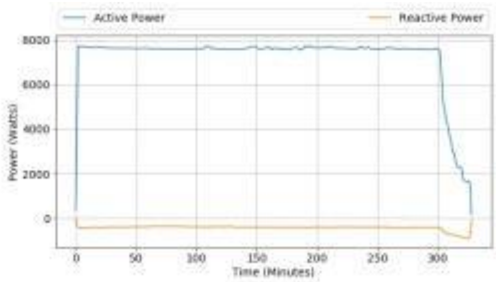


Figure 3 – Sample EV charging curves (Power vs Time)

As can be seen from these examples, the start-up ramp is very similar in each case but the end of the charging is very different depending on the battery cell balancing carried out. One vehicle that seemed to have different profile was the Porsche Cayenne, vehicle #22 which seemed to exhibit some variation from the "top hat" shape seen on the other vehicle types:

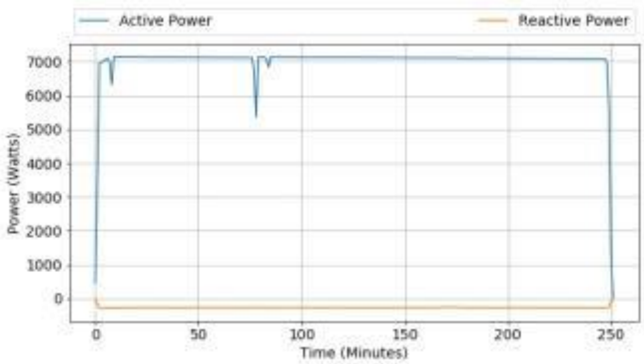


Figure 4 – Porsche Cayenne typical charging profile

5.1 Algorithm optimisation

The algorithm as implemented in this demonstrator project took 1 hour blocks of data with the aim to determine whether an EV had started or finished charging in that timeslot. The human factors of EV charging mean that any domestic EV charge point is very unlikely to be used more than once per day (so in practice there is only 1 chance per day to miss an EV charging event). By contrast, there are 24 one hour slots where it is possible to assert the presence of an EV being charged when there is no car present. Consequently, the algorithm must be pessimistic in order to avoid reporting many false positive events – which could lead to overestimation of the use of that particular charge point and the consequent risk of inaccurate analysis of the load profiles when considering future network planning.

The figures below show the truth tables for each 1 hour detection window and the consequent charge point reporting accuracy taken over a 1 week period for the same data. It can be seen that with the balance set for a 95% correct determination of absence of charging vs. a 75% accuracy for presence of charging the errors in overall charge point usage over a week gives an approximately even error spread either side of the correct answer (which was 7 charges a week for this dataset).

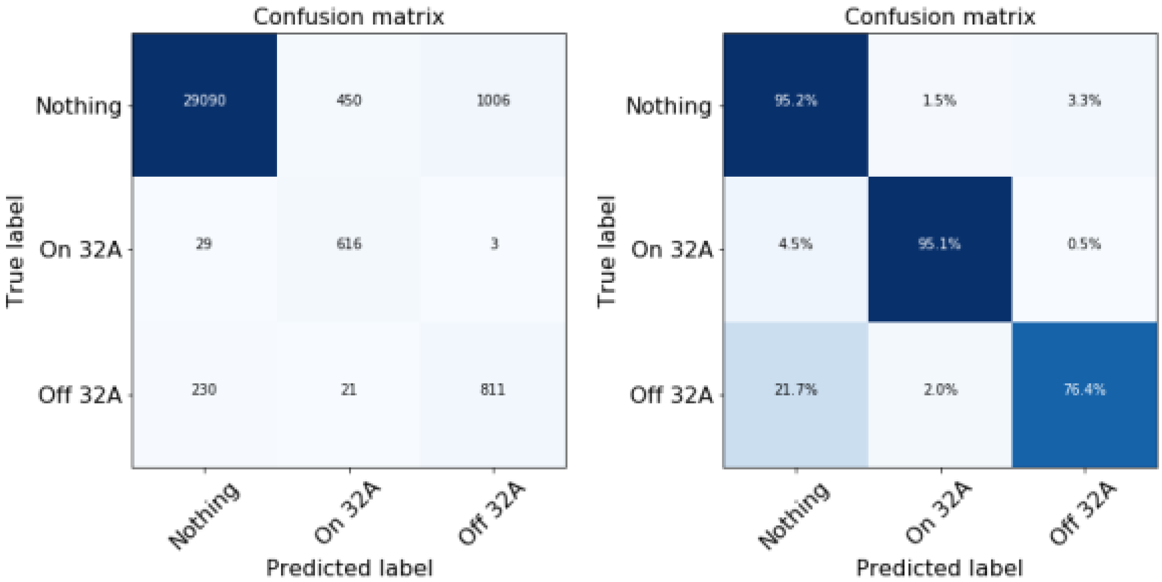


Figure 5 – EV Charging Confusion Matrices

The ‘Confusion Matrices’ above (Figure 5) shows how often the algorithm makes errors for a large number of 1 hour reporting periods. The Y axis shows the “truth” (whether a charging event was present or not) and the X axis shows what the algorithm predicted. Tests falling in the boxes on the diagonal from top left to bottom right are correct answers. Tests falling outside that diagonal represent errors. The left hand box shows the absolute number of tests and the right hand box is the same data expressed as a percentage.

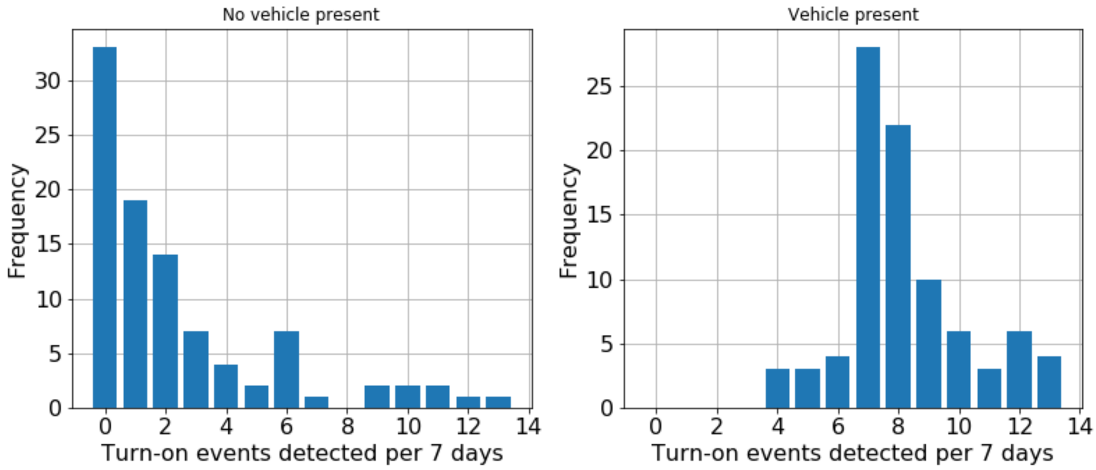


Figure 6 –Data from the confusion matrices shown in Figure 5 replotted over one week (i.e. 168 hourly slots) for the case where no charging actually took place (left) and where there was 1 charge each day

Overall, this seems to the developers to offer a fairly clear reflection of charge point usage given that usage will vary over any given week so while perfect accuracy would be desirable, data informing any decision to uprate a given circuit will have noise from many sources. It is possible to bias the algorithms further in order change the trade-off between false positives and negatives depending on the commercial requirements for the information.

The algorithm has been developed to detect the characteristic signatures of the charging of electric vehicles. It uses a deep neural network, comprising of convolutional, pooling and long short-term memory (LSTM) layers that have been trained to respond to the switching on and off signatures of electric vehicle chargers, as observed in historical data.

6. Implementation and Software Requirements

The algorithm has been developed within the following Python environment:

- Python 3.5
- Numpy (v1.12.1)
- Tensorflow (v1.1.0)
- Keras (v2.0.2)

The following key files are included in the source code package:

- DeepConvLSTM.py – Python package containing the neural network setup, construction and main configuration parameters.
- ElectricNation_001_DeepConvLSTM_32A.h5 – Keras model file containing the trained neural network
- keras_tsc.py – Python scripts for training the neural network, evaluating, and plotting
- performance metrics.
- Generate_training_dataset.ipynb – iPython notebook for loading in CSV data from GridKey MCUs, electric vehicle charging logs, and preparing data for training the neural network. Corrects class imbalance (removes some of the negative cases) to ensure quality of training.
- Generate_validation_dataset.ipynb – iPython notebook (similar to above) but for preparing validation data. Preserves the characteristics of the data, including class imbalance.
- ml_metrics.py - Python package containing metrics relating to the performance of the
- neural network.
- sliding_window.py - Python package containing data pre-processing support functions
- zoe_preprocess.py - Python package containing data pre-processing support functions
- zoe_power_switching_model.py - Python package containing data pre-processing support
- functions
- power_switching_model.py - Python package containing data pre-processing support functions, from Project Galaxy.
- dnn_infer.py – Python package containing the data pre-processing and neural network inference engine within a single function call.
- anaconda_environment.yml – Anaconda configuration file, that can be used to create a Python environment as used for development and testing

7. Data Pre-processing

Two IPython notebooks have been developed for handling data: Generate_training_dataset.ipynb and Generate_validation_dataset.ipynb. Both files load GridKey MCU data from CSV files, as background power data, and superimpose the vehicle 'reference exemplars' – charging signatures recorded separately for each vehicle type during testing at Millbrook Proving Ground. There is the option within these files (on by default to generate training data) to combine data samples in random permutations to create a larger number of aggregated test samples. This was necessary to generate the volume of training data required without reusing samples.

For the training dataset, 10,000 synthetic test samples of length 4 days (with 1-minute resolution) were generated using background data from circa. 2014 (Project Galaxy), believed to contain negligible electric vehicle activity. Electric vehicle charge exemplars were superimposed at a random time, once per day with up to ±2 hours random movement either side per day. Only 32A vehicle chargers were used, resulting in 31 vehicle charge exemplars, from 5 unique vehicle types. From these, 6 exemplars were kept for use in the validation dataset, leaving 25 for training purposes. Data was restructured into an array format suitable for training the neural network, with four columns:

Active Power (Mean) Delta	Reactive Power (Mean) Delta	Active Power (Mean)	Reactive Power (Mean)
Delta (change since previous time point) for the Active Power (Mean)	Delta (change since previous time point) for the Reactive Power (Mean)	Average (mean) *active* power during the 1-minute sampling window	Average (mean) *reactive* power during the 1-minute sampling window

Array data was scaled to bring the power values approximately into the range (-1:1), split into many smaller sections using a sliding window, and the mean was subtracted from each section to remove the effect of varying peak power levels. A class label was assigned to each section to denote, to identify if it contained: just background noise (0), a vehicle charger turning on in the 32A category (3), or a vehicle charger turning off in the 32A category (4). (The remaining two classes have been disabled, but supporting code remains for: a vehicle charger turning on in the 16A category (1), and a vehicle charger turning off in the 16A category (2).)

After sectioning, 1.9 million sections of 60 minutes were available from the training dataset. A 10% fraction of the dataset was separated (random sections) and reserved for evaluation of the neural network model performance during training (the test dataset).

The dataset was naturally class imbalanced: there are a lot more samples that are just background noise than there are samples containing vehicle charge signatures. This can lead to poor training performance. The ratio of classes within the training and test datasets was adjusted, by removing 75% of the background noise samples. This reduced the total number of samples (training and test) to approximately 550,000, with class distribution as shown below in Figure 1.Mean)

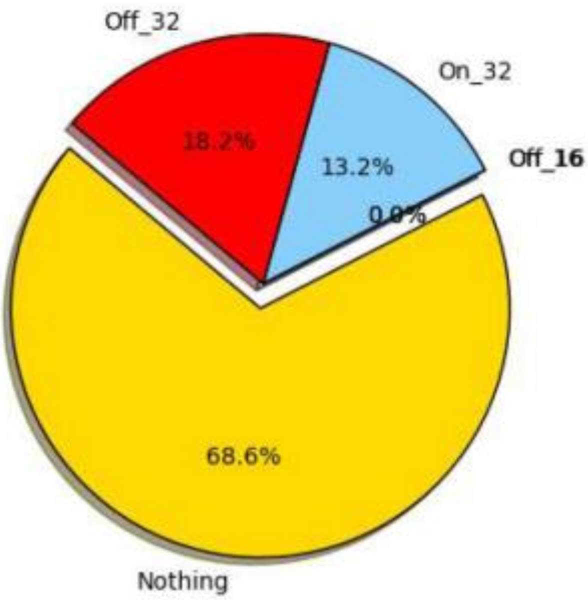


Figure 1 - Distribution of classes in the training dataset, after rebalancing

The validation dataset was initially produced using 'legacy' data (circa. 2013), although the dataset load characteristics were believed unsuitable (possibly industrial loads), so fresh synthetic test samples were generated from the Project Galaxy data. The dataset was generated in a similar manner to the training dataset (above), although the validation vehicle reference exemplars were used instead, and the distribution of classes was not adjusted. Each test sample was duplicated and vehicle charging reference exemplars were selected at random and superimposed on to every other test sample. This meant that every test sample was in the dataset twice – once with a vehicle charge exemplar and once without.

8. Neural Network Design

The neural network is a classifier that operates on multi-channel time series data. Seven variant designs of network were considered, comprising of convolutional and LSTM layers. Network performance was evaluated using a hold-out test dataset after 50 epochs of training. The preferred design is of structure:

Layer ID	Type	Notes
Input		Dimensions: 60 x 4
Layer 1	Convolutional 1D	Kernel size: 5, filters: 64
Layer 2	RELU	
Layer 3	Convolutional 1D	Kernel size: 5, filters: 128
Layer 4	RELU	
Layer 5	Max. Pooling 1D	Pool size: 2
Layer 6	Dropout	Rate: 0.25 (Training only)
Layer 7	Convolutional 1D	Kernel size: 5, filters: 256
Layer 8	RELU	
Layer 9	Max. Pooling 1D	Pool size: 2
Layer 10	Convolutional 1D	Kernel size: 5, filters: 512
Layer 11	RELU	Pool size: 2
Layer 12	Dropout	Rate: 0.25 (Training only)
Layer 13	LSTM	Units: 120
Layer 14	Dropout	Rate: 0.10 (Training only)
Layer 15	Dense (fully connected)	Units: 50
Layer 16	Dropout	Rate: 0.25 (Training only)
Layer 17	Dense (fully connected)	Units: 5
Output	SoftMax	Dimensions: 5 x 1

The neural network was developed using the Keras package, which operates on the Tensorflow framework within Python.

9. Algorithm Training

The deep convolutional LSTM neural network was trained over 500 epochs (full passes of the training dataset), to minimise the categorical cross-entropy loss function, using the Root Mean Square Propagation (‘RMSprop’) variant of stochastic gradient descent optimisation. The trained model was saved to a Keras HDF5 file (supplied). On a GPU-based workstation the training took approximately 8 hours.

10. Performance Validation

Algorithm performance was evaluated using a single pass of the validation dataset, by comparison with the known ‘ground truth’ class labels. The accuracy is assessed as the proportion of predictions that identified the correct category, with note of the number of false positives (vehicles identified when not present) and false negatives (vehicles not identified). Performance metrics are contained within the attached slides.

11. Installation

The recommended method of operation is via the Anaconda distribution of Python (www.anaconda.com). The included environment file can be used to create a suitable Conda environment using the following command:

```
conda env create -f anaconda_environment.yml
```

Note that if it is wished to operate Keras and Tensorflow in conjunction with a GPU, the above command is not suitable and the developer's installation instructions should be followed instead. The Python script files included in this distribution do not require installation, and can be run as soon as the compressed archive file has been decompressed.

12. Execution

An end-to-end example of training the neural network and evaluating test data is included in the Python script file keras_tsc.py. Alternatively, the included Python package dnn_infer.py includes a single function that can be imported and called to analyse a single MCU data file in CSV format.

For example:

```
from dnn_infer import analyse_MCU_file

MCU_filename = ‘MCU 001143233008 from 2017-01-30 to 2017-02-06.csv’

results = analyse_MCU_file(MCU_filename)
```

13. Limitations of the current algorithm

As discussed above the, risk of false positives vs false negatives has been balanced by considering the known usage patterns of EV chargepoints however the current state of development has one key weakness which we were not able to address within the current project. This concerns the use of 1 hour time segments as the search unit. The neural network algorithm developed for this project has been trained to look for just 1 charging event; two charging events in the same 1 hour slot will be recorded as only a single event. This is likely to be acceptable if the cable being monitored was feeding just 1 house since charging events are typically much longer than 1 hour and the chance of 2 cars being topped up from a 7kW charger within a 1 hour slot will be very low, perhaps even when many households have 2 EVs.

This algorithm is designed to operate at the substation however, and the feeders here supply 80 or more houses. This makes the likelihood of more than 1 EV connecting in any 1 hour slot very high, particularly given the known early evening connection peak identified by Electric Nation. Given this limitation we have considered several options to mitigate the risk but have not developed or tested them at this point.

- **Shorter unit of search**. It is thought that cutting the unit of search down to 15 min would not seriously affect the false positive/negative rate and would reduce the likelihood of multiple connections in a single time slot by a factor of 4. However, on a busy feeder this is unlikely to be enough. Narrowing the window beyond this would make it difficult to exclude many of the other loads leading to a higher false positive rate.

- **Train algorithm for multiple charge signatures.** A better possibility would be to train the algorithm to recognise multiple events in a given 1 hour slot. This is expected to be viable but will require much more training data. In practice it's difficult to predict how much though, and it's often possible to create "synthetic data" from a smaller real dataset (as we have in the existing work) and still get good results. Testing the existing algorithm on real live data is an important first step however.
- **More rapid sampling**. Sampling much more often than 1 per min is expected to reveal more characteristic structures within the charging profile. If a good fingerprint can be found in the on/off events related to charging and/or greater confidence can be obtained for other loads on the network which can then be reliably excluded, charging events can then be counted much more reliably. The following sections recommend this as the preferred route forward but it does represent a significant development effort.

Overall it is likely that a combination of the latter 2 suggestions will yield the best results going forward

14. Milestone and Deliverables Performance

Project Milestones	Quarter	Target Date	Completed Date	Milestones
	0	01/05/2016	01/06/2016	Project Initiation
	1	29/07/2016	01/02/2017	Millbrook Site Survey with WPD Installation of EV Charging points Install 1 off GridKey systems at Millbrook testing ground Determine LV sites of interest (WPD)
	2	31/10/2016	25/04/2017	Collect raw data from a variety of vehicle types Initial analysis of data by TTP Install GridKey systems on WPD network where there are known PIV clusters
	3	31/01/2017	30/04/2017	Project Q3 Completion
	4	28/04/2017	01/10/2017	Initial analysis of data by TTP Creation of Functional Requirements Document Creation of initial algorithm Electronically combine raw data with existing substation data Test algorithm Update algorithm as appropriate
	5	31/07/2017	01/12/2017	Collect real substation data Test algorithm with real data Update NIA Project Progress Report
	6	31/10/2017	25/06/2018	Project Completion Completion of Closedown Report Delivery of Functional Specification for Algorithm

15. Project Closure Recommendations

The problem to be solved turned out to be more complex than expected – other than a typical "top hat" shape there was little that (electrically) determined it to be an EV rather than some other load.

Although we were able to get a reasonably high probability of detection (>95% for individual hourly samples), this was partially as a result of limiting the problem (so only looking for circa 7kW vehicles and also only looking for one switch on event per hour) and partly by optimising the algorithm for accurate positives (at the expense of a higher (~75%) negatives accuracy). In other words, the algorithm seldom reported a car charging when one was not there but more often missed a car that was in fact charging.

There is an alternative of seeing the 7kW rising ramp and then trying to eliminate other things it could be however this only works if you know all the other things it could be so this is not really a practical solution.

Although we were able to get a reasonably high probability of detection (>95% for individual hourly samples), this was partially as a result of limiting the problem (so only looking for circa 7kW vehicles and also only looking for one switch on event per hour) and partly by optimising the algorithm for accurate positives (at the expense of a higher (~75%) negatives accuracy). In other words, the algorithm seldom reported a car charging when one was not there but more often missed a car that was in fact charging.

The decision described in section 3.1 to carry out the calculations at the Data Centre end was absolutely correct for this project – there was a lot of processing and more particularly storage required for the neural network and the GridKey MCU520 has only very limited processing capability. The downside of this decision is that the current algorithms are restricted to operating on data at 1 minute intervals. In the context of identifying EV charging this is far from ideal since the chargers typically change state over a much shorter time period so a lot of potentially useful information is lost. Moving to more rapid time reporting, 1 second data or even data at 5Hz, is expected to offer a lot of scope for improvement. The concern however is the cost of backhauling this much greater volume of data.

Since the start of the project, GridKey has introduced a newer MCU – known as the MCU318. Although it still only has limited processing capability in order to minimise the price of the unit, it does contain a Cortex M4 which is considerably more powerful than the PIC processor in the MCU520 and does have spare processing capacity and some available memory.

The MCU318 is sampling the current and voltage sensor data at 6.4kHz and is calculating parameters at 5Hz. The system is already carrying out waveform captures for use both on distance to fault analytics and also for future PQ software upgrades.

So a potential solution could be to carry out some of the processing locally on this new generation of GridKey devices. This will require some innovation in numerical computing as well as algorithm development because of the very limited processing and memory resources on those cost focussed units but we think there is good scope for performance improvements using this approach.

Sensing at the higher frequencies the GridKey is now capable of also offers potential to resolve EV chargers from other domestic loads, and even different chargers from each other.

It is therefore proposed to swap one of the MCU520s currently being used on the Electric Nation project with an MCU318 and using the information from the smart chargers to carry out data capture so the results can be analysed and determine whether a software algorithm could be incorporated into the MCU itself which detects when an EV charge is started and sends an alarm. This could also be something an LV-Cap container running in an OpenLV architecture could run. (OpenLV is a Network Innovation Competition project to provide a software architecture which allows data from multiple sources to be combined and shared with the Network Operators and third party companies, more details are ar https://openlv.net/). Due to the difference in switch off profiles, this is something that may require the Data Centre still to detect.

Overall the project has demonstrated that detecting EV charging profiles is difficult but possible however it requires better than 1 minute data resolution to be successful.