SEAI RD&D 2019

SMARTeBuses

# Online Model to Detect Abnormal Wind Power Supply

## Deliverable number: WP3-D3

This Project has received funding from the SEAI RD&D Programme.
Grand Award Number: 19/RDD/519

**Project Acronym:** SMARTeBuses
**Project Full Title:** SMART electric Buses
**Grant Number:** 19/RDD/519
**Project URL:** https://smartebuses.github.io/web/

| | |
|---|---|
| Deliverable type: | Deliverable / Report |
| Dissemination level: | Public |
| Delivery Date: | August 31, 2021 |
| Number of pages: | 16 |
| Keywords: | SMARTeBuses, Artificial Intelligence, Anomaly Detection, Wind Power |
| Authors: | Dr. Alejandro Arbelaez - University College Cork |

# Contents

# List of Figures

# List of Acronyms

**SMARTeBuses** SMART electric Buses
**SEAI** Sustainable Energy Authority of Ireland
**AI** Artificial Intelligence
**LSTM** Long-Short Term Memory
**MAE** Moving Average Error
**MAS** Moving Average Standard deviation

# 1 Introduction

This document corresponds to the delivery WP3-D3 "*Online Model to Detect Abnormal Wind Power Supply*"' of the SMART electric Buses (SMARTeBuses) project, funded by the Sustainable Energy Authority of Ireland (SEAI) RD&D Programme. This project is classified as Non-economic public Good Research under the European Union (EU) State Aid regulations and will exploit, combine and improve cutting-edge Artificial Intelligence (AI) technologies to develop and implement optimization models for the operation of electric buses in Ireland with operational constraints.

In this deliverable, we describe a model to detect anomalies in the production of wind power, e.g., unexpected lower production that might impact the energy supply and the regular operation of the electric buses. We recall that the availability of wind power is intermittent and only available under specific circumstances. Therefore, the national grid might prepare contingency plans to supply the demand when the clean energy production is lower than expected. Similarly, the national must must also decide to switch off the turbines when the energy supply of electricity exceeds the demand.

Generally speaking, there are three main categories of anomalies:

- Point anomalies (or global anomaly): assumes that abnormal observations are significantly different that the entire dataset. For instance, in a credit card fraud system, a point anomaly corresponds to unexpected high payments.
- Contextual anomalies: abnormal observations deviate from the dataset in a specific context. For instance, daily expensive dinners on during the holidays is normal, but odd otherwise.
- Collective anomalies: an abnormal set of observations deviating significantly from entire dataset. However, individual values in the set might not significantly deviate to represent a point anomaly.

In this deliverable we aim at building an online system to detect potential anomalies that require further investigation. These anomalies are observations that significantly deviate from the expected pattern.

# 2 Models

In this deliverable, we assume that our dataset is a time series data, which consists of a sequence of ordered observations indicating the amount of clean energy produced in Ireland during a 15-minute time interval (in MW). Given the nature of our dataset, observations are not completely independent and it is expected that the observation at time $x_i$ will impact the value of the immediate future observations. This way, abrupt changes in the time series can be seen as abnormal observations, for instance, assuming that a wind farm generates the following amounts of power in MW every 15 minutes:

$$200, 210, 205, 208, 50, 51, 55, 48, 55 \tag{2.1}$$

A non-temporal view of the previous observations might not represent abnormal patterns as there are two independent clusters. However, the temporal aspect of the time series should identify an unusual behaviour from 208 to 50 that requires further investigation, e.g., a failure in a wind turbine. Currently, there are two stablished techniques for time series, i.e., based on predictions and based on unusual patterns [1]. The former identifies points that significantly deviate from the predictions. While the latter uses machine learning to detect anomalies using traditional supervised machine learning methods, e.g., SVM or KNN [2], or unsupervised methods based on clustering, e.g., DBSCAN or k-Means [3]. These machine learning methods are typically preferred for multi-dimensional datasets

## 2.1 Anomaly Detection

In [4] the authors provide an extensive literature review of anomaly detection techniques. These techniques can be classified as forecasting methods and distance-based approaches. The former aims at building a model (i.e., using statistics, machine learning, or deep learnign) and recognising that abnormal observations deviate from the expected pattern. The second alternative uses clustering models to evaluate the distance (or similarity) to the neighbours observations, abnormal points are far apart from their neighbours.

In this deliverable, we use the follow three-step method to detect anomalies:

- forecasting the next observation in the time series;
- measuring the deviation or error between the prediction and actual observations;
- identifying a threshold $\rho$ indicating whether the error of a given observation is an anomaly.

[5] provides a general description of our forecasting methods for wind power prediction in the SMARTeBuses project. In this particular, in this deliverable, we focus our approach, in the LSTM model and the predictions currently available in the smart grid dashboard.[1] A popular approach to calculate $\rho$ assumes that the dataset is normally distributed. This way, observations that are more than 3 (resp. 2) standard deviations ($\sigma$) away from the mean can be declared as anomalies. Figure 2.1 a normal distribution with zero mean a unit $\sigma$, it can be observed that 68% of the observations are within $1\sigma$, 5% are within $2\sigma$ and 0.3% are within $3\sigma$ deviations.

[1] https://www.smartgriddashboard.com

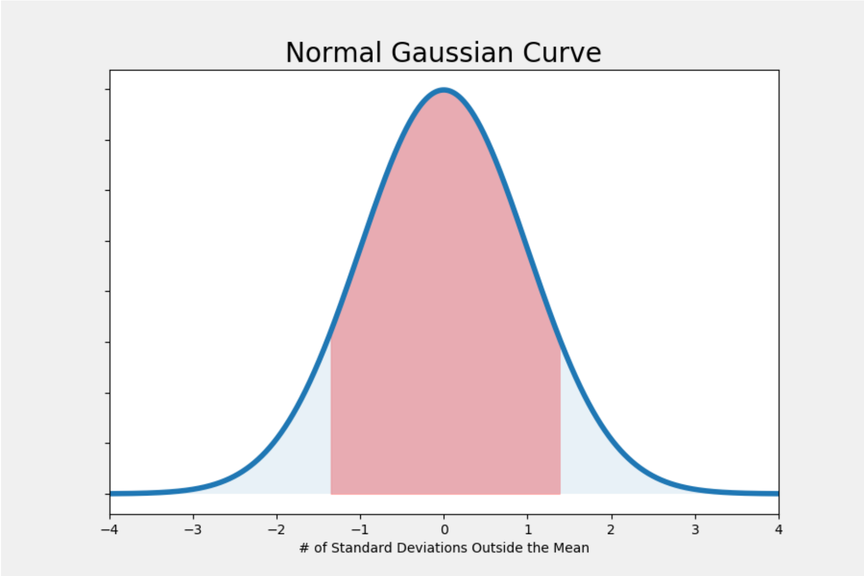



**Figure 2.1:** Normal distribution with zero mean and a unit $\sigma$.

## 2.2 Box Plot

The box plot [6] is a statistical tool to estimate the value of $\rho$ and detect anomalies in a given dataset. Figure 2.2 provides a visual representation of a box plot, here we observe the main components of the box plot, i.e., min. (non-abnormal) observation, lower quartile (Q1), median, upper quartile (Q3), and the max. (non-abnormal) observation. The interquartile range (i.e., IQR = Q3-Q1) indicates the limit between anomalies and normal observations, i.e., observations outside the region Q1 - 1.5·IQR and Q3 + 1.5·IQR.

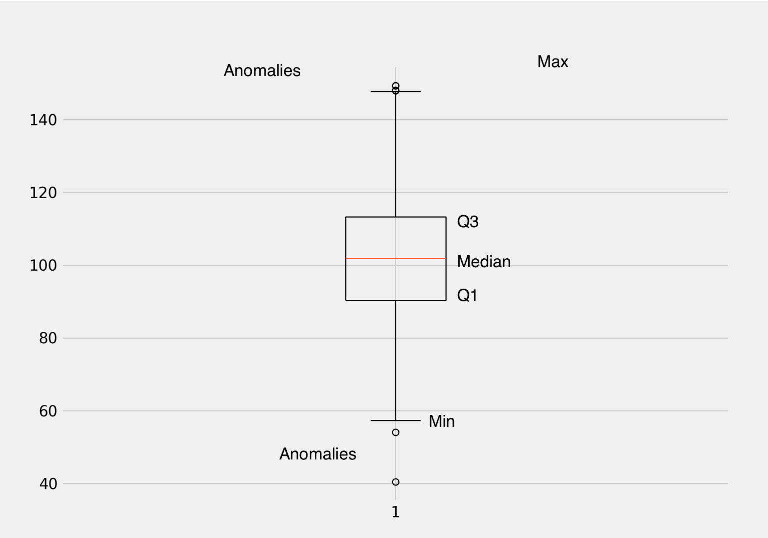



**Figure 2.2:** Box-plot example.

## 2.3 Online anomaly detection

In this deliverable, we build a system capable of alerting the user about potential anomalies. These unusual observations might require further further investigation to avoid disruptions in the grid as a result of disruptions in the production of renewable power, i.e., wind power. In this context, we incrementally calculate the Moving Average Error (MAE), i.e., difference between the prediction and the actual value, and the Moving Average Standard deviation (MAS) of a pre-defined time window.

Algorithm 2.1 outlines a pseudocode of the online algorithms using Python and Pandas [7]. We assume that Line 2 reads the dataset from a given file, here we assume that the file is in csv format and includes the actual values, and the predictions, and each line provides the amount of power in a 15-minute interval. Line 3 calculates the error, Line 4 and 5 calculate the moving average (with a time window tw) of the error and the standard deviation, and lines 6 and 7 calculate the interval of normal observations.

**Algorithm 2.1:** Online Anomaly Detection

```
1 def online_anomaly_detection(file, tw):
2   ts = pd.read_csv(file)                                  #Load dataset
3   ts["error"] = ts["actual"] - ts["prediction"]           #Error
4   ts["mean"]  = ts["error"].rolling(window=tw).mean()     #Avg. error
5   ts["std"]   = ts["error"].rolling(window=tw).std()      #Avg. std
6   ts["3S"]    = ts["mean"]+3*series1["std"]               #Upper limit
7   ts["-3S"]   = ts["mean"] - 3 * series1["std"]           #Lower limit
```

# 3 Empirical Evaluation

We start our empirical evaluation with a general overview of the dataset from October, November, and December 2020. Figure 3.1 shows our observations with 15-minute intervals for the actual wind power values, our predictions using the Long-Short Term Memory (LSTM) model, and the smart grid predictions currently available in the system. As it can be observed, our LSTM predictions are slightly better than the reference predictions, we refer the reader to [8] for a more detailed empirical evaluation of our models.

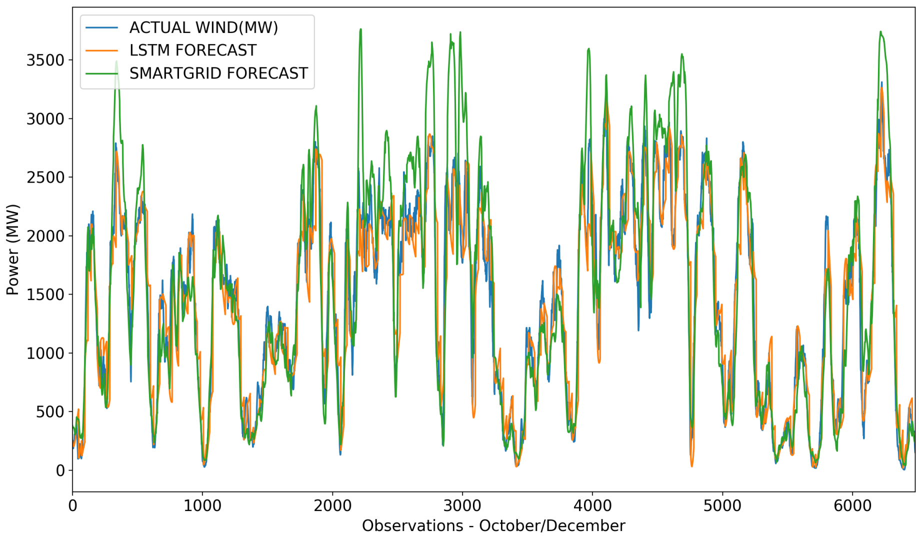
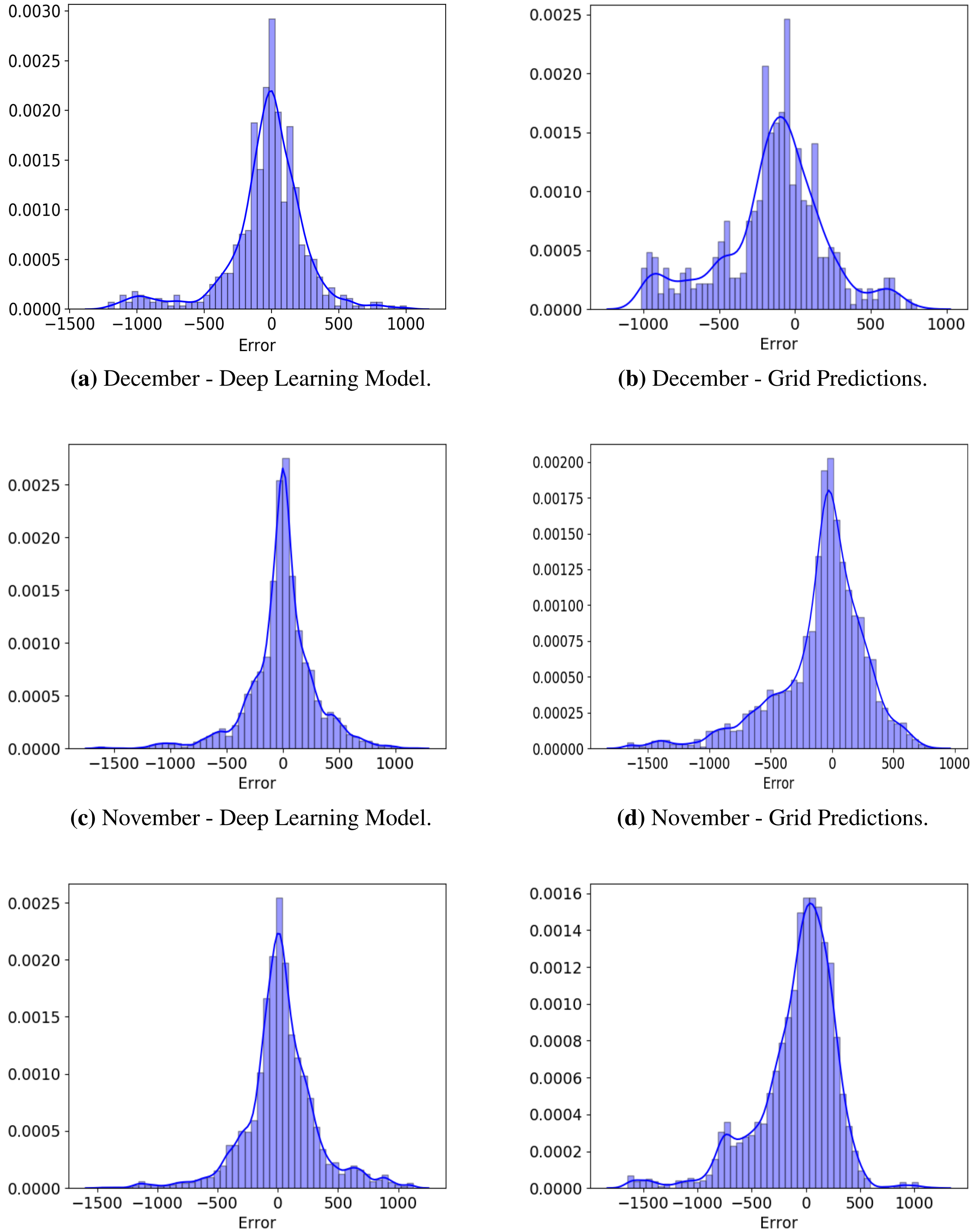



**Figure 3.1:** Time Series data for October to December.

Figure 3.2 depicts the histogram of the observations for each month. As it can be seen our observations (i.e., error = predictions - actual values) resemble a normal distribution, excluding the November/December dataset where the histograms with the smart grid errors seem a bit skewed. However, we would like to remark these observations fail the statistical test for normality. Nevertheless we use the $3\sigma$ threshold for anomaly detection.




**(a)** December - Deep Learning Model.

**(b)** December - Grid Predictions.

**(c)** November - Deep Learning Model.

**(d)** November - Grid Predictions.

**(e)** October - Deep Learning Model.

**(f)** October - Grid Predictions.

**Figure 3.2:** Histograms with the error of the Deep Learning model and the Smart grid predictions.

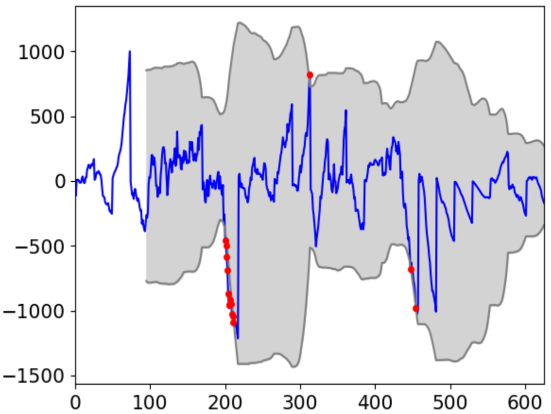
In the following, we analyse our static anomaly detection method (i.e., with prior knowledge the monthly data) and our online one. The main difference between the two alternatives lies in the fact that the static methods needs to wait until the end of the month to identify anomalies, while while the online method is capable of alerting the user of unexpected patters after receiving a new observation.

| Month | Model | Static Method | Online Method |
|---|---|---|---|
| October | LSTM | 20 | 15 |
| October | Grid | 0 | 9 |
| November | LSTM | 67 | 77 |
| November | Grid | 60 | 61 |
| December | LSTM | 51 | 55 |
| December | Grid | 51 | 50 |

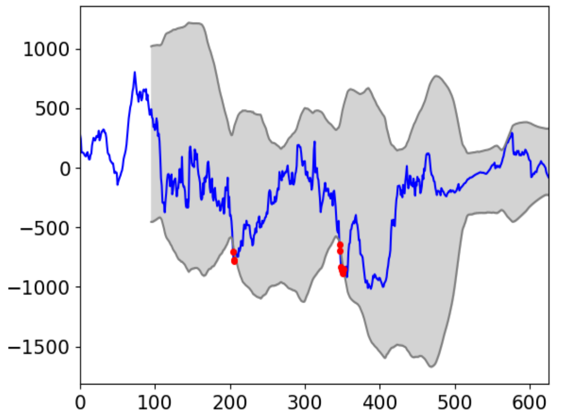
**Table 3.1:** Total anomalies for the static and online methods.

Table 3.1 indicates the number anomalies identified for each forecasting model. It can be observed that the Grid model is more conservative and we anticipate that this model might ignore abnormal observations due to it provides higher errors than the LSTM. For instance, for October 2020 this method was unable to find anomalies in the production of wind power, a highly unexpected scenario due to the high variability of wind production. Both models identify November as a critical month with more wind power potential anomalies. We attribute this phenomenon to the weather variability in the country before the official start of the winter in Ireland.

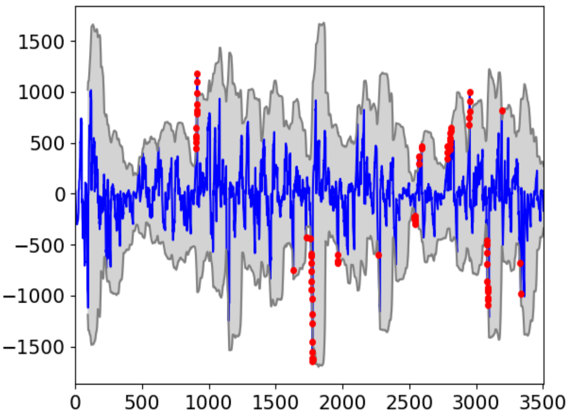
Figures 3.3 and 3.4 provide a visual representation of the anomalies for each reference month. Figure 3.3 provides empirical results for the online system, the blue line represents the error of the observations, red dots represent anomalies, and the grey area represents the interval of normal observations (i.e., within $3\sigma$ of the mean error). Figure 3.4 provides our empirical results for the static monthly method, anomalies are placed above (resp. below) the dashed red area. Similarly to the previous figure, the grid model reports less anomalies than the LSTM model.



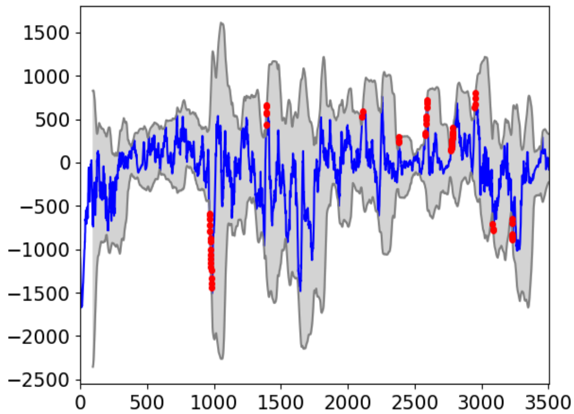

**(a)** December - Deep Learning Model.



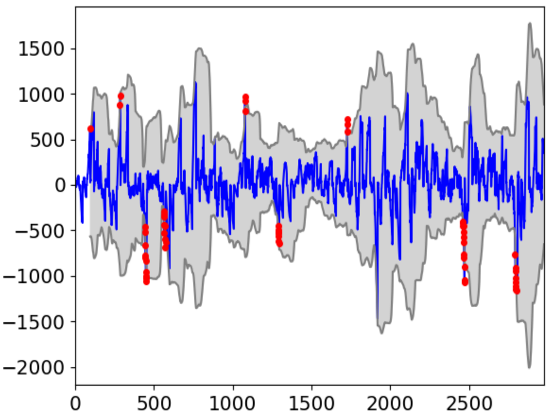

**(b)** December - Grid Predictions.



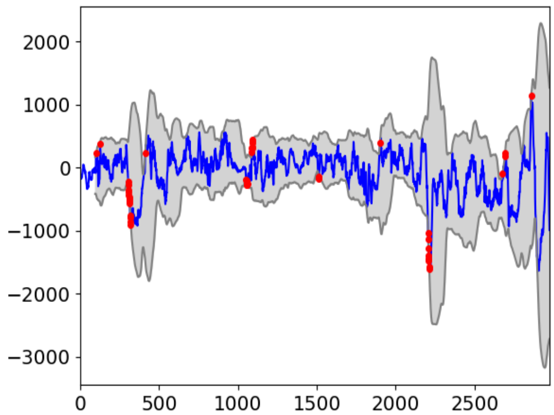

**(c)** November - Deep Learning Model.




**(d)** November - Grid Predictions.

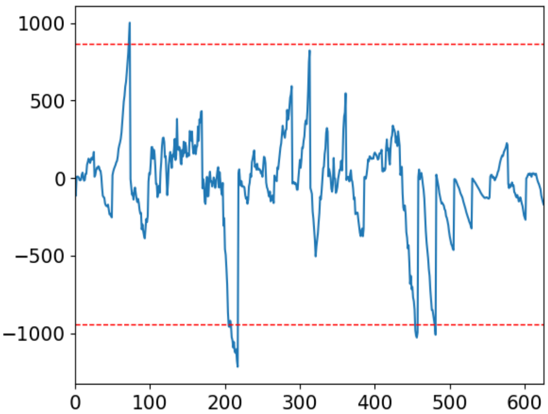



**(e)** October - Deep Learning Model.

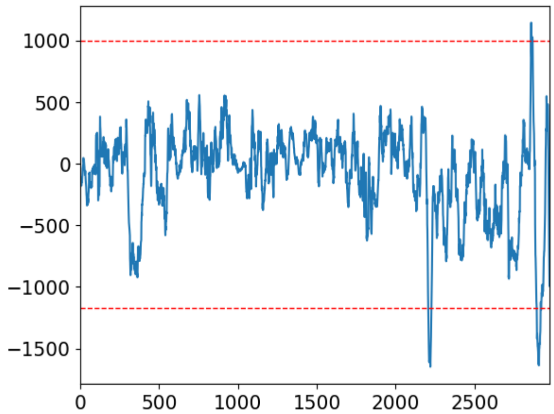



**(f)** October - Grid Predictions.

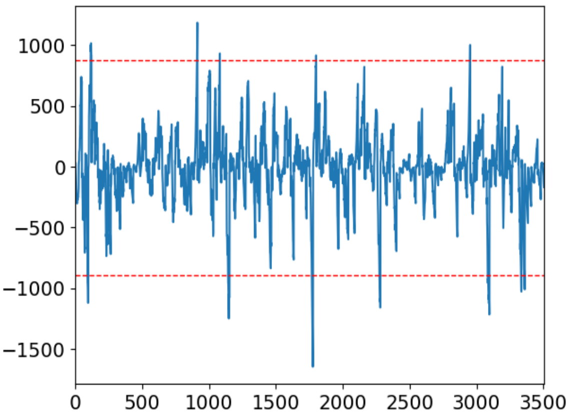
**Figure 3.3:** Online Anomaly detection - Red points denote anomalies.



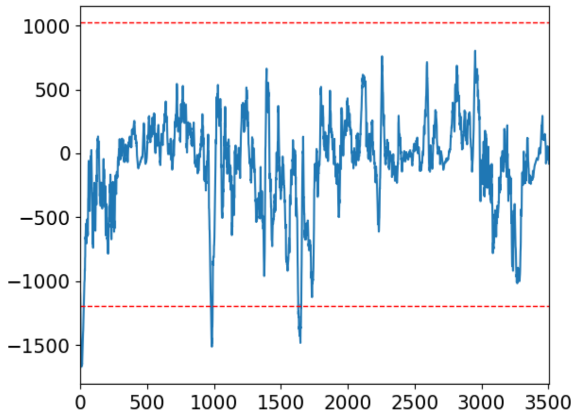

**(a)** December - Deep Learning Model.



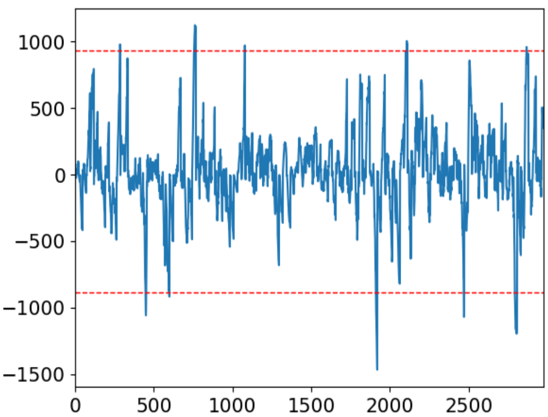

**(b)** December - Grid Predictions.



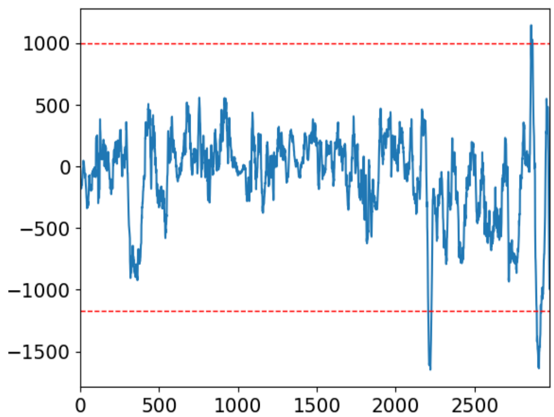

**(c)** November - Deep Learning Model.




**(d)** November - Grid Predictions.




**(e)** October - Deep Learning Model.




**(f)** October - Grid Predictions.

**Figure 3.4:** Online Anomaly detection - Red points denote anomalies.

Finally, Figure 3.5 shows anomalies for each model using the box plot. This method reports more anomalies than the previous one as the interval of normal observations is about $\mu \pm 2.6\sigma$. Interestingly the grid method reports no anomalies above the first quartile (Q1), we attribute this phenomenon to the distribution of the dataset as observed in Figure 3.2.

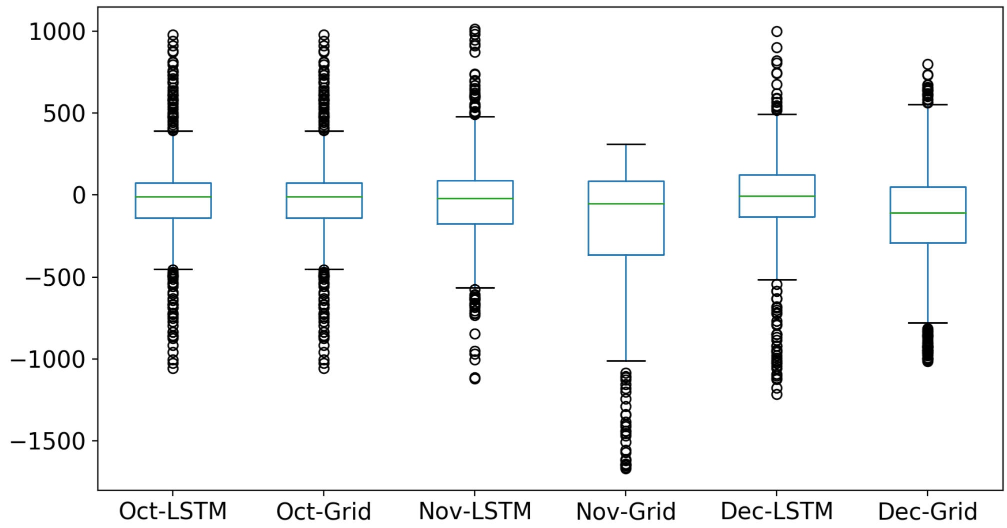



**Figure 3.5:** Box Plot for both models.

# 4 Conclusions

In this deliverable we have described a popular anomaly detection method for time series data. This method assumes that the dataset is normally distributed and observations with an error outside a predefined area (e.g., $\mu \pm 3\sigma$ ) are considered anomalies. Furthermore, we evaluate two forecasting methods, based on our own predictions using LSTM [8], and the predictions currently available in the smart grid system. Our empirical results suggest that our LSTM model is capable of alerting the users of more potential anomalies than the reference method.

# Bibliography


[1] F. Giannoni, M. Mancini, and F. Marinelli, “Anomaly detection models for iot time series data,” *CoRR*, vol. abs/1812.00890, 2018.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995.

[3] S. I., P.-B. A., and W. G, “Unsupervised clustering approach for network anomaly detection,” *Communications in Computer and Information Science*, vol. 293, 2012.

[4] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, 2009.

[5] P. Jarvis and A. Arbelaez, “Deliverable WP3-D1: Forecasting models for power prediction,” Jan. 2021, tech. Rep. WP3-D1 SMARTeBuses project.

[6] J. w. Tukey, *Exploratory Data Analysis. Addison-Wesley*. Addison-Wesley, 1977.

[7] M. W., *Pandas: a foundational python library for data analysis and statistics*. O’Reilly Media, 2012.

[8] P. Jarvis and A. Arbelaez, “Deliverable WP3-D2: Validation of wind power models,” April 2021, tech. Rep. WP3-D2 SMARTeBuses project.

SMARTeBuses

SMART electric Buses

June 17, 2021