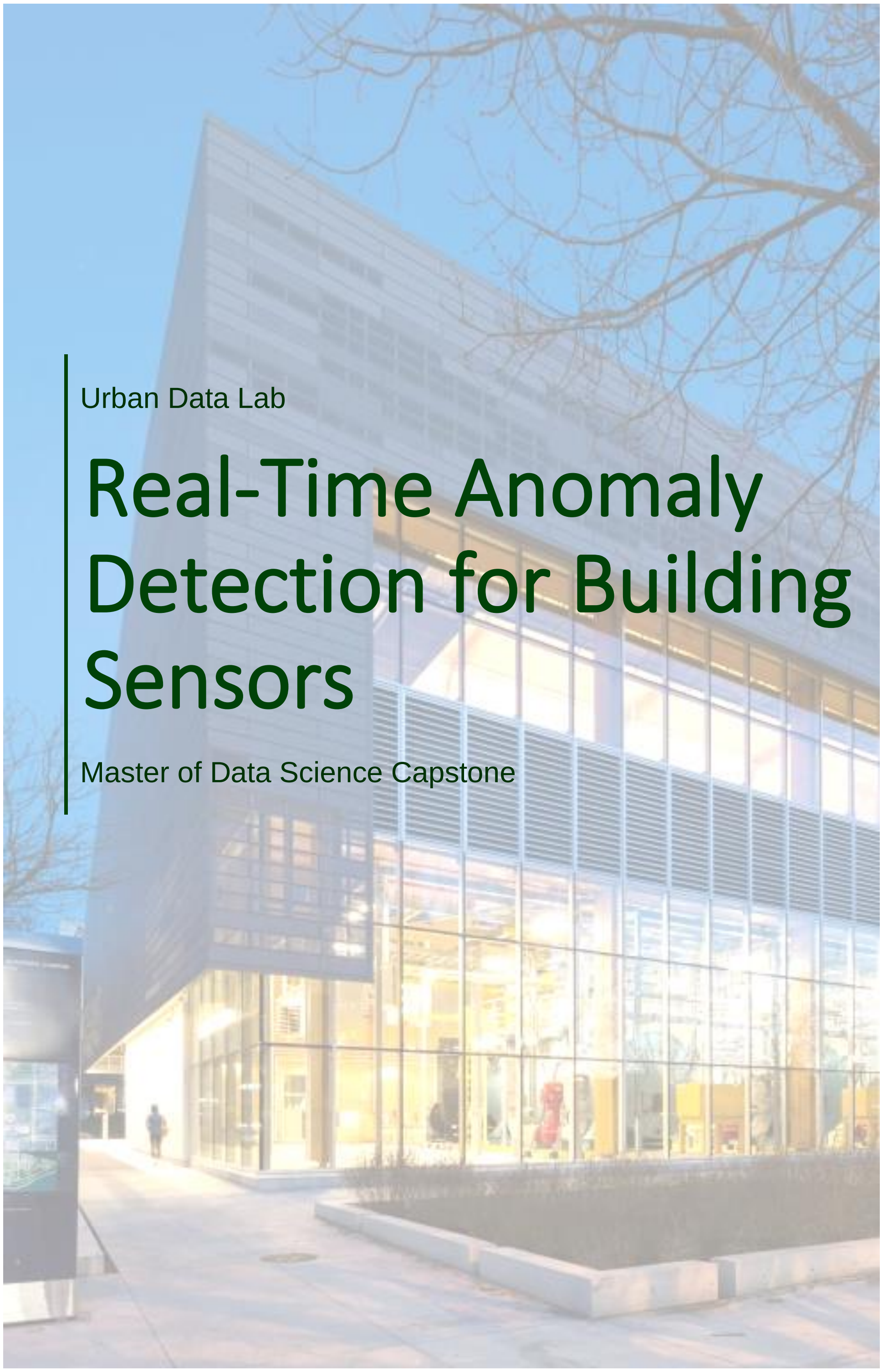


Urban Data Lab

Real-Time Anomaly Detection for Building Sensors

Master of Data Science Capstone

Nathan Smith, Mitch Harris, Ryan Koenig
6-21-2021

Executive Summary

The Urban Data Lab (UDL) advances data access, data management, and data analytics capabilities on the University of British Columbia (UBC) campus with a goal of addressing campus-wide sustainability challenges. UDL has access to the UBC Energy and Water Services (EWS) SkySpark analytics platform that collects data from buildings on campus including heating, ventilation and air conditioning (HVAC) equipment and energy data. UDL stores data from SkySpark in their own InfluxDB database. UDL have noticed potentially erroneous data reporting from SkySpark and there is currently no system in place with InfluxDB to flag these data. The project goal was to develop a real-time anomaly detection system using open-source tools that could be used with UDL's database.

The approach used in this study provides near real-time anomaly detection with InfluxDB. Model training is completed by querying sensor data on an infrequent basis (for example monthly), training, and saving the models. Anomaly detection occurs on a continuous basis by reading recent data, loading and running the previously trained models, and writing predictions to InfluxDB. A subset of Campus Energy Center (CEC) boiler sensors available in SkySpark was selected for the study to test this approach.

A long short-term memory recurrent neural network with an encoder-decoder architecture (LSTM-ED) is used for anomaly detection. This was selected as it provides a general model with good performance in recent studies. The generalizability of the LSTM-ED is important given the wide variety of sensor types available to UDL. The model is trained in an unsupervised approach using sequence reconstruction of input data. Anomaly predictions are then based on identifying data with high sequence reconstruction error. The LSTM-ED was found to have good initial performance on the selected subset of CEC sensors. A data pattern was identified that the model had trouble detecting but it is believed that performance can be improved using more sophisticated anomaly threshold identification rules.

A dashboard and notification system were also implemented with the anomaly detection model in a test InfluxDB environment. The dashboard can be built directly in InfluxDB and provides a simple display of sensor data highlighted as either normal or anomalous. The notification system also uses built-in InfluxDB functionality and can be configured to send notifications for data predicted as anomalous.

This study provides an initial open-source anomaly detection approach that can be used by UDL with InfluxDB. The approach is general and should be applicable to a variety of sensors. Additional studies that can be considered for next steps include implementing the model online for several test sensors and monitoring performance, improving the model anomaly detection threshold method, comparison of the LSTM-ED with additional models, testing additional sensors, and building a more complex dashboard and notification system as required. Ideally, the detection system could ultimately be used to provide campus and building managers with real-time or near real-time notifications of potential issues in system operations reducing operational costs, downtime, and unexpected maintenance.

Table of Contents

Figures

Tables

1 – Introduction

1.1 - Urban Data Lab

The Urban Data Lab (UDL) was founded in 2019 to advance data access, data management and data analytics capabilities on the University of British Columbia (UBC) campus with the goal of addressing campus-wide sustainability challenges. It accomplishes this by providing open access of UBC sustainability data to researchers, policymakers and operational staff. It also supports the monitoring and measurement of sustainability performance for buildings, transportation, and natural assets specifically as it relates to the policy commitments of UBC Sustainability Initiative and Campus and Community Planning (UDL, 2021).

UDL has access to the SkySpark platform managed by UBC Energy and Water Services (EWS). SkySpark is an analytics platform for smart devices and equipment systems and collects data from buildings on the UBC campus including information such as heating, ventilation and air conditioning (HVAC) equipment and energy data. UDL stores data from SkySpark in their own database using InfluxDB and the data are made available to students, researchers, and operational staff at UBC. These data are used to support UDL's collected research and operational interests including energy profiling, fault detection, spatial visualizations, and dashboard decision support systems (UDL, 2020).

1.2 – Project Scope and Objectives

The project goal was to develop a real-time anomaly detection system using open-source tools that could be used with InfluxDB. UDL have noticed potentially erroneous data reporting to InfluxDB from SkySpark and there is currently no system in place with UDL's InfluxDB instance to flag these data. The detection system would allow users to understand when data are potentially erroneous such that the anomalous data can be removed from analyses or used to understand if sensors or systems should be investigated. The anomaly detection system would also include visualization of data identified as potentially anomalous on a dashboard and provide automated notifications to users. Ideally, the detection system could be improved and used to provide campus and building managers with the real-time or near real-time notifications of potential issues in system operations reducing operational costs, downtime, and unexpected maintenance.

A high-level schematic of the system is provided below.

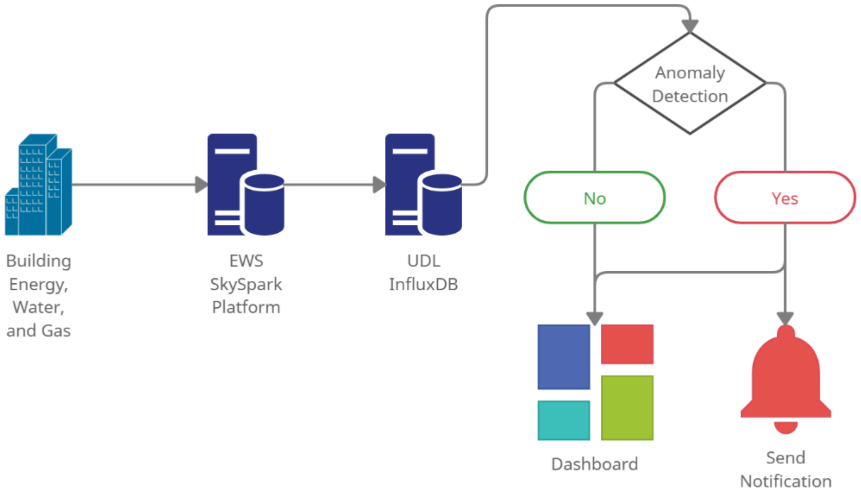


FIGURE 1: High-Level Anomaly Detection Schematic

The project used the following approach described in this report:

- Select a subset of SkySpark sensors for the study and complete an initial review of the data to understand variations in sensor patterns and data anomalies.
- Identify a real-time anomaly detection framework that could work with UDL's InfluxDB instance.
- Research and select an anomaly detection model.
- Test performance of the detection model with the selected data subset.
- Implement the real-time anomaly detection model including a dashboard and notification system.

It was also a goal of the project to identify an anomaly detection system general enough that it could be applied to sensors outside of the subset used in this study.

2 – Study Data

The Campus Energy Centre (CEC) hot water boiler facility sensor data was selected as the subset of SkySpark data for the project. The selection was based on recommendations from UDL as a dataset that would provide a variety of sensor types.

2.1 – Database

UDL uses InfluxDB to store data provided from the EWS SkySpark platform. InfluxDB is a timeseries optimized database (Influxdata, 2021a) and UDL is using the open-source implementation of InfluxDB. The database currently only contains limited historical SkySpark data from approximately June to September 2020 and live streaming of SkySpark data to InfluxDB is not functional. UDL is working on implementing live streaming with the database using Telegraf. Telegraf is a plugin driven server agent and is used to listen to http posted data provided by EWS and parse/direct the data to InfluxDB. Support was also provided to UDL during this project for Telegraf parsing and implementation. Note that UDL is currently working with version 2 of InfluxDB.

Data from InfluxDB can be accessed directly through a user interface or using command line interface (CLI) or client libraries. The InfluxDB Python client library (Influxdata, 2021b) was used as the main tool to read and write from InfluxDB for this project.

Most of the data used in the study was downloaded directly from the EWS SkySpark platform user interface due to the limited data available in InfluxDB during the project timeline. A limited amount of data had to be selected for the project as this is a time-consuming process.

2.2 – Campus Energy Centre Sensors

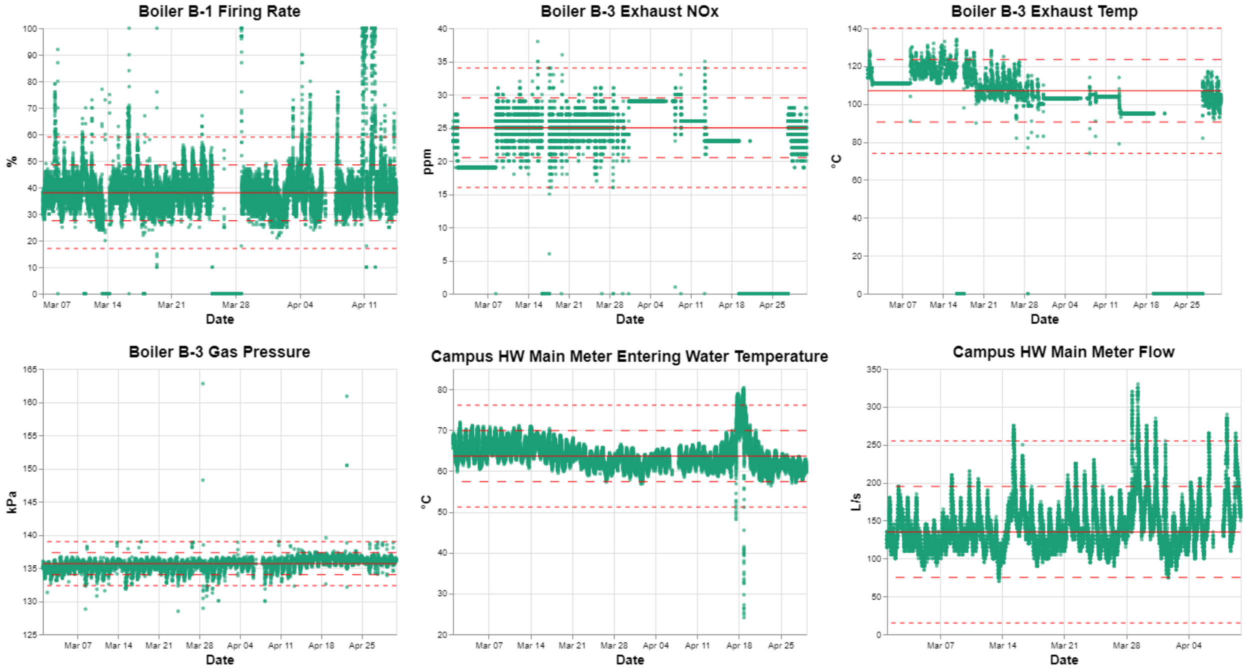
The SkySpark platform currently has 72 numerical sensors for the CEC. The sensors have different data resolutions on SkySpark varying from approximately one to 15-minute reading intervals. During the exploratory data analysis (EDA) step of the project, data were visually assessed using the SkySpark user interface and samples of data were downloaded for additional exploration (approximately 2-months for sensors with 1 minute interval data and 4 months for sensors with longer interval data).

The sensors can be generally categorized into three main groups based on the exploration:

- Group A: Sensors related to power output of the boilers. This includes power and energy as well as firing rate of the boilers and speed of the pumps. These sensors can have large fluctuations related to the operation of the boilers. This group contains 22 sensors.
- Group B: Sensors that are less likely to experience sharp fluctuations in data compared with Group A. These include temperature (water temperature, exhaust temperature), pressure, concentration, and tank levels. This group contains 41 sensors.
- Group C: Water flow rate sensors. This group contains 9 sensors.

The main purpose of the grouping was simply to provide an initial understanding of the variety of sensors available in the exploration phase. It also provided guidance for selecting sensors for anomaly detection testing recognizing the project timeline did not provide allowance to individually look at all sensors.

A selection of sensor traces for a two-month period is shown below. The plots include the median as well as 1.5x and 3x the interquartile range (IQR) of the record shown on the plots. The IQR values are shown to help provide an understanding of data variability.



Note: Lines indicate the median, 1.5xIQR and 3xIQR.

FIGURE 2: CEC SENSOR DATA - 6 SENSORS, 2-MONTH DURATION

The figure shows there is a variety of patterns between the sensors. Some observations from the subset of data shown above include:

- Campus HW Main Meter Entering Water Temperature is generally contained with 3xIQR whereas the Campus HW Main Meter Flow has high fluctuations that appear to be normal data exceeding 3xIQR.
- Boiler B-3 Gas Pressure has short duration spikes that are well out of the 3xIQR compared with a longer duration spike that occurs at the Campus HW Main Meter Entering Water Temperature.
- Boiler B-3 Exhaust Temp appears to have steps in the data compared with the Boiler B-1 Firing Rate and Boiler B-3 Gas Pressure which both follow a constant trend.
- Boiler B-3 Exhaust and Boiler B-3 Exhaust Temp have periods of flatline data that appear to be erroneous.

Figure 3 shows two sensors over a one-year duration.

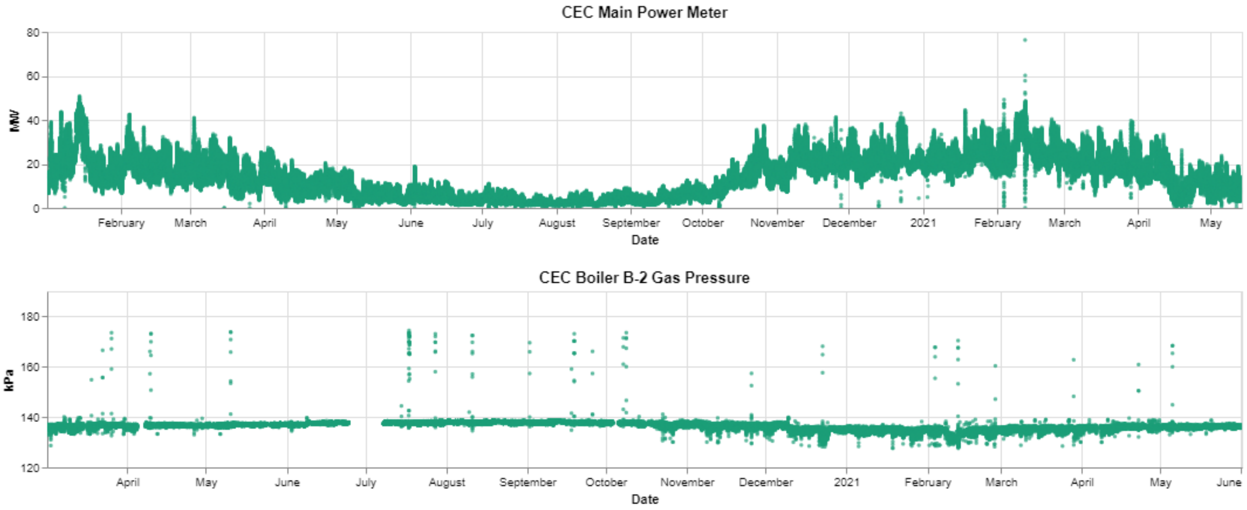


FIGURE 3: CEC SENSOR DATA - 2 SENSORS, 1-YEAR DURATION

Figure 3 shows a seasonal change in the data for CEC Main Power Meter while CEC Boiler B-2 Gas Pressure is largely constant year-round with minimal seasonal variation.

This variety in patterns indicates that a flexible anomaly detection model is likely required if a single model is to be applied to all sensors. It also suggests that model training may be required for individual sensors (or groups of very similar sensors) versus training a single model that could be applied to all sensors.

Additionally, Figures 2 and 3 indicate there appear to be different types of data anomalies between the sensors.

2.3 – Data Anomalies

Labelled data anomalies are not readily available for the CEC sensors. Accordingly, any data that did not appear to follow the typical pattern of data for a sensor was considered as anomalous for the study. These anomalies could be a function of sensor malfunction, equipment malfunction, abnormal system operations, or network related issues resulting in erroneous recordings.

Sensor anomalies could include the following and many of these are shown on Figures 2 and 3:

- Single point anomalies: one or two points that fall well above or below the trend of data.
- Repeated values (flatline) anomalies: the same value repeated for a period, not in-line with surrounding data patterns.
- Data spikes: a rapid increase/decrease in data compared with the typical data pattern and could represent an operational issue.
- Erroneous values: Sensors recording a value that does not appear correct. For example, a sensor recording zeros or values such as -999 or 999.

- Contextual anomalies: a pattern of data that does not appear standard based on the historical sensor data (Cook et al., 2020).

These would all be considered anomalous data within this study. As labelled data was not available for the sensors the project was framed as an unsupervised modelling approach. However, an exercise of manually reviewing and flagging data that appeared visually anomalous was completed to help provide an understanding of model performance. Additional details on the manual labelling and performance criteria used in the study are described in Section 5.

It should be noted that it is not expected that sensors would have labelled anomalies except in cases when detailed records have been kept (the occurrence of this is unknown). Accordingly, viewing the project as an unsupervised modelling problem is likely representative of how the anomaly detection model would be used.

3 – Anomaly Detection Framework

UDL is currently implementing live streaming from the SkySpark platform to InfluxDB using Telegraf. Telegraf is a plugin driven server agent (Influxdata, 2021c) and is used to listen to http posted JSON data provided by EWS. Telegraf parses the data into the line protocol format required by InfluxDB and writes to the database.

The project included identifying an open-source framework for real-time (or near real-time) anomaly detection that works with the InfluxDB/Telegraf. Several options were explored, and the selected framework was implemented within a test InfluxDB environment. The test environment was used because live streaming from the SkySpark platform to the UDL InfluxDB instance was not operational during the project. An overview of the test environment is provided in Appendix A.

3.1 – Options Considered

Several anomaly detection frameworks that could work with InfluxDB/Telegraf are described below. The selected framework is discussed in Section 3.2.

3.1.1 – In-line with Telegraf

Telegraf can output data to multiple sources and continuously run executables through its exec and execd plug-ins. This functionality could allow Telegraf to send parsed data directly to InfluxDB as well as to an external program used for anomaly detection. This was initially identified as the option that would provide the most real-time anomaly detection capability on incoming data from SkySpark.

However, this option was not pursued in this project as Telegraf was still being configured by UDL and testing of the system would not have been possible. It is recommended that this option be considered in the future to potentially provide faster detection.

3.1.2 – Python Client Library

The InfluxDB python client library 'influxdb-client-python' (Influxdata, 2021b) allows reading and writing directly from InfluxDB. A python script could be used on a timed interval to read data from InfluxDB, detect anomalies, and write back to InfluxDB. For example, the script could be run every 1-minute to predict the most recent data written to InfluxDB. The script would be configured to read the window of data required by the anomaly detection model.

This approach would not be as real-time as the inline Telegraf option described in Section 3.1.1 but would instead detect using a near real-time approach on a timed interval. This option was selected for this study as it does not rely on operational Telegraf streaming.

3.1.3 – Flux Tasks

InfluxDB includes functionality to set automated tasks. These tasks can be used to send data out of the database on a schedule which could be picked up by Telegraf or an API for anomaly detection. This option provides a reasonable approach and uses built-in InfluxDB functionality but was considered similar to the approach using the InfluxDB python client library. This option was not pursued in favor of using the python client library.

3.2 – Framework Selected

The python client library 'influxdb-client-python' was selected for implementing near real-time anomaly detection. A schematic of the framework is shown below, and the components are described.

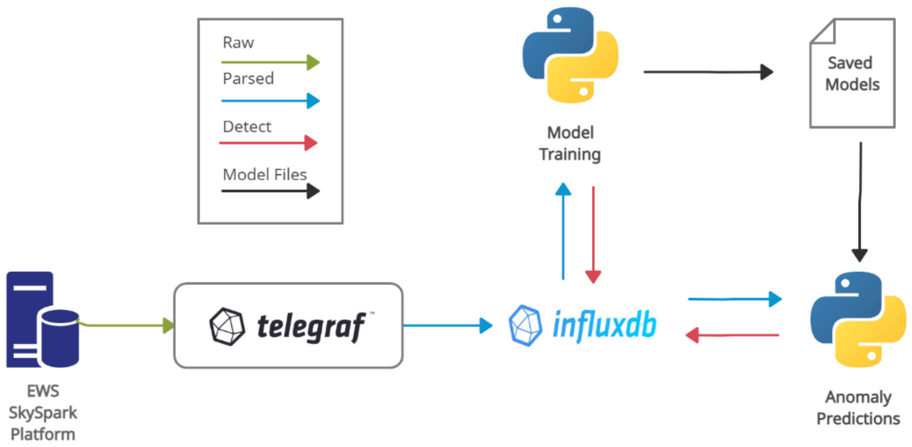


FIGURE 4: ANOMALY DETECTION FRAMEWORK

- **Telegraf:** EWS posts SkySpark data in JSON format to an http port. Telegraf listens to the port, parses the JSON data, and write the data to InfluxDB. These data are labelled as READINGS measurements in InfluxDB. Note that this is still being implemented by UDL.
- **InfluxDB:** UDL InfluxDB instance. All data are stored in multiple 'measurements' within a single InfluxDB 'bucket'. A description of the InfluxDB measurements is provided in Section 3.3.

- **Model Training:** A model training python script is run on a selected interval (for example every month) to update the anomaly detection models. A description of the model is provided in Section 4. Anomalies identified during the model training process are labelled as TRAINING_ANOMALIES measurements in InfluxDB.
- **Saved Models:** The trained models (1 model per sensor) from the model training script are stored in HDF5 files. These files are updated each time the model training script is run.
- **Anomaly Predictions:** The anomaly prediction python script is run on a selected high frequency interval (for example every 1-minute or every 5-minutes). The script reads the most recent data in InfluxDB including the window of data required to make a prediction, loads saved model files, makes anomaly predictions, and writes the results to InfluxDB. These predictions are labelled as REALTIME_ANOMALIES measurements in InfluxDB.

3.3 - InfluxDB Schema

A bucket is a named location where time series data is stored in InfluxDB. Within an InfluxDB bucket, a measurement category acts as a container for tags, fields, and timestamps. Tags provide metadata and are typically used in queries while fields represent data values. Each record in InfluxDB has a timestamp. The UDL InfluxDB instance uses a single bucket to store all SkySpark related data. The SkySpark bucket measurements used in the anomaly detection framework are described in the following sections and a schematic showing the relationship between the measurements is provided below. Note that the val_num field is currently duplicated in the TRAINING_ANOMALY and PREDICT_ANOMALY measurements. This was done as a field is required for each measurement in InfluxDB. TRAINING_ANOMALY and PREDICT_ANOMALY could be merged into a single measurement to remove one of these duplications. Alternatively, the anomaly flags could be set as fields instead of tags such that val_num is not required.

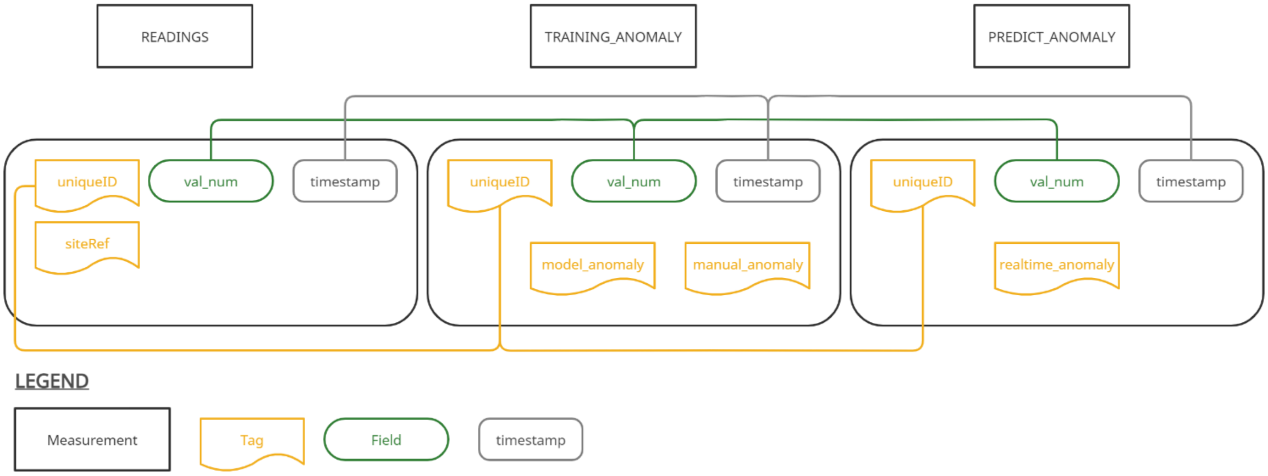


FIGURE 5: INFLUXDB SKYSPARK MEASUREMENTS FOR ANOMALY DETECTION

3.3.1 READINGS Measurement

Contains the SkySpark data streamed from InfluxDB. The READINGS measurement has the following tags and fields relevant to the anomaly detection framework:

- **siteRef:** Tag key that specifies a UBC building. This can be used to query all sensors from a specific building on campus.
- **uniqueID**: Tag key that specifies unique sensor IDs. This can be used to query a specific sensor.
- **val_num**: Field key used for numeric values with sensor data stored as the field value. This study only looks at sensors that have val_num as the field key.

3.3.2 TRAINING_ANOMALY Measurement

Contains the anomalous data predictions from the training process. Every time model training is completed for a sensor (for example monthly), the predictions are rewritten. Also contains a tag for manually labelled anomalies. The TRAINING_ANOMALY measurement contains the uniqueID tag key, val_num field key with the sensor values and:

- **model_anomaly**: Tag key used to store the predictions from model training (False = normal, True = anomalous). The field values are rewritten every time model training is completed.
- **manual_anomaly:** Tag key used to store manually labelled anomalous data (False = normal, True = anomalous). Any field values labelled as anomalous (True) are removed from the training process as described in Section 4.

3.3.3. PREDICT_ANOMALY Measurement

Contains the anomalous data predictions from the anomaly predictions script. Only contains predictions that were made in real-time (does not include any data from the training process). The PREDICT_ANOMALY measurement contains the uniqueID tag key, val_num field key with the sensor value and:

- **realtime_anomaly**: Tag key used to store the boolean predictions from the anomaly prediction script (False = normal, True = anomalous).

4 – Anomaly Detection Model

4.1 – Model Review

The model criteria for this study included the following:

- Provide anomaly detection on univariate data such that the model could be applied to individual sensors without a requirement of additional data.

- Be general/flexible enough that the model could be applied to any building sensor. While the project study used the CEC boiler sensors, the goal was to provide an anomaly detection method that could be applied to many sensors.
- Useable in a continuous predict-detect approach such that model training was not required every time a prediction on new data was made.
- Trainable in an unsupervised approach. Labelled anomalies were not readily available for the CEC sensors and it is anticipated that this will likely be the case for most sensors that use the model.

An initial review of anomaly detection models for sensor data was initially completed with a focus on the above criteria. Internet of Things (IoT) sensor studies was used for the review as they represent a wide variety of univariate data and appear to be the focus of recent sensor anomaly detection studies. General categories of anomaly detection models identified based on the review include: (Cook et al., 2020; Liu et al., 2020; Riazi et al., 2019)

Statistical: Simple statistical rules include using a set number of standard deviations from the mean or multiplication of the inter-quantile range (IQR) to identify anomalous data. More sophisticated approaches include using a model such as Autoregressive Integrated Moving Average (ARIMA) to predict expected future data and compare if actual data is sufficiently different from the predicted value to be labelled as anomalous.

Machine Learning: Machine learning models typically provide an approach that typically provides less inferential information compared with the statistical model. Models include using k-Nearest Neighbour (KNN), One-Class Support Vector Machine (OCSVM), Isolation Forest, and Gaussian Mixture Models. Many of these models are used in anomaly detection by identifying clusters of data and labelling anomalies as data that are not near these clusters.

Deep Learning: Several neural network approaches have been used successfully with IoT sensors for anomaly detection. Examples include one-dimensional convolutional neural networks (CNN) or long short-term memory recurrent neural networks (LSTM RNN). These represent very flexible models but provide more of a black box modelling approach.

The above categories represent a high-level classification and Cook et al. (2020) provides a good discussion of recent anomaly detection methods. There are also methods that use a combination of approaches or even ensembles of model. For example, Microsoft’s SR-CNN Anomaly Detector accessible through the Azure cloud platform uses a spectral residual transformation combined with a CNN (Ren et al., 2019).

There are also many variations that have been studied for individual anomaly detection models. For example, a recent study on IoT-based vertical plant wall for indoor climate control sensors (Liu et al., 2020) looked at the performance of multiple LSTM architecture variations including bi-directional LSTM (bi-LSTM) and a LSTM with an encoder-decoder (LSTM-ED). The study also discussed variations of the anomaly detection approach used with the model. For example, the LSTM model could be used to recreate a sequence of input data for comparison with the original input data (sequence reconstruction) or could be used to predict the next point (point prediction) for comparison with the actual next data point. A poor sequence reconstruction or point prediction would indicate potentially anomalous data.

In summary, there are multiple categories of anomaly prediction models, and each category has many options. Furthermore, there are approaches that include using multiple models or data transformations and there are variations on each of the models as noted in the LSTM example above.

4.2 – LSTM-ED Model

The LSTM was selected for the study based on the model review completed for the following reasons:

- Good performance on recent IoT sensor studies (Cook et al., 2020; Liu et al., 2020) and provides a flexible model that should be applicable to a variety of sensor types. Minimal project data was available during the model review process and selecting a flexible model was an important consideration. The project goal was also to provide an approach/model that can be used on sensors not evaluated in this study.
- The model can be used in an unsupervised approach.
- Model parameters can be saved after training and loaded during model prediction.
- The model is sufficiently simple enough that it could be understood and built within the project timeline. Microsoft's SR-CNN anomaly detection model looks like it could provide good performance (Ren et al., 2019) but is more complex than the LSTM and the method requires anomaly injection.

Specifically, the LSTM-ED was selected for the study based on a comparison of various LSTM architectures where the LSTM-ED provided the highest performance (Liu et al., 2020).

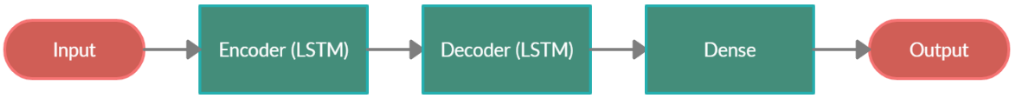


FIGURE 6: LONG SHORT-TERM MEMORY ENCODER-DECODER MODEL

The first layer of the model includes LSTM cells that act as an encoder on the model input sequence creating internal model features. The second layer also consists of LSTM cells and acts as a decoder of the internal model features to the model output sequence. A dense layer is used to connect the LSTM decoder layer and the output. The LSTM model for the project was built using Keras with 128 LSTM cells for each of the encoder and decoder layers based on example architectures (Brownlee, 2020; Geron, 2019; Keras, 2021; Keras, 2020).

Two variations on the LSTM-ED model were used in this study: next point prediction, and sequence reconstruction. These variations are described by Liu et al. (2020) and are shown in the following figure.

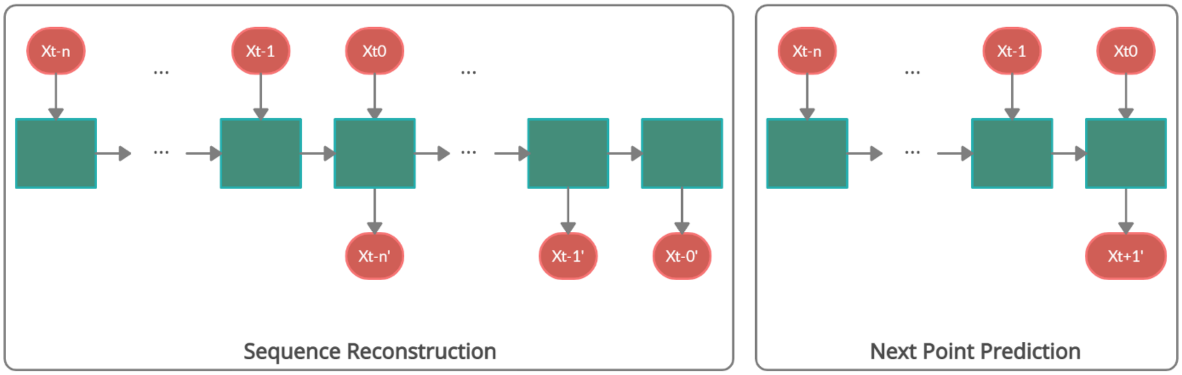


FIGURE 7: SEQUENCE RECONSTRUCTION AND NEXT POINT PREDICTION

Next Point prediction uses a window of input data (x_{t-n} to x_{t0}) and trains the LSTM-ED to predict the next point in the data (x_{t+1}’). The actual value (x_{t+1}) is then compared with the predicted point and if there is a high difference in the values above a selected threshold, the point is flagged as an anomaly. The study looked at various data window sizes (described in Section 5) and the error threshold method used is described in Section 4.3.3.

Sequence reconstruction uses a window of input data (x_{t-n} to x_{t0}) and trains the LSTM-ED to predict the same sequence of data (x_{t-n}’ to x_{t0}’). If there is a high difference in the actual and predicted sequence, the data is flagged as anomalous. There are variations on the sequence reconstruction method including training the model by sliding the window of data one step at a time (overlapping window method) or sliding the entire window by the window length (non-overlapping window) or sliding by several data points (semi-overlapping window) (Liu et al, 2020). This study uses the non-overlapping window training approach as it reduces model training time and had the highest performance in the study by Lui et al. (2020). Predictions are made by passing the data point of interest as the last point in the window to the model. If the sequence reconstruction error is high with the window error above the selected threshold, the point is flagged as anomalous. There are also variations on how to identify data as anomalous with sequence reconstruction including:

- Flagging all data in a poor reconstruction window as anomalous (not just the last point in the window).
- Using a more sophisticated method to track windows with high reconstruction error and only flagging a point as anomalous if it occurs within a certain number of high error windows (Keras, 2020).
- Using methods that consider the distribution of error within a window instead of just setting an error threshold value.

4.3 – Model Pipeline

A schematic of the LSTM-ED model pipeline is shown below, and descriptions of each component are described in the following sections.

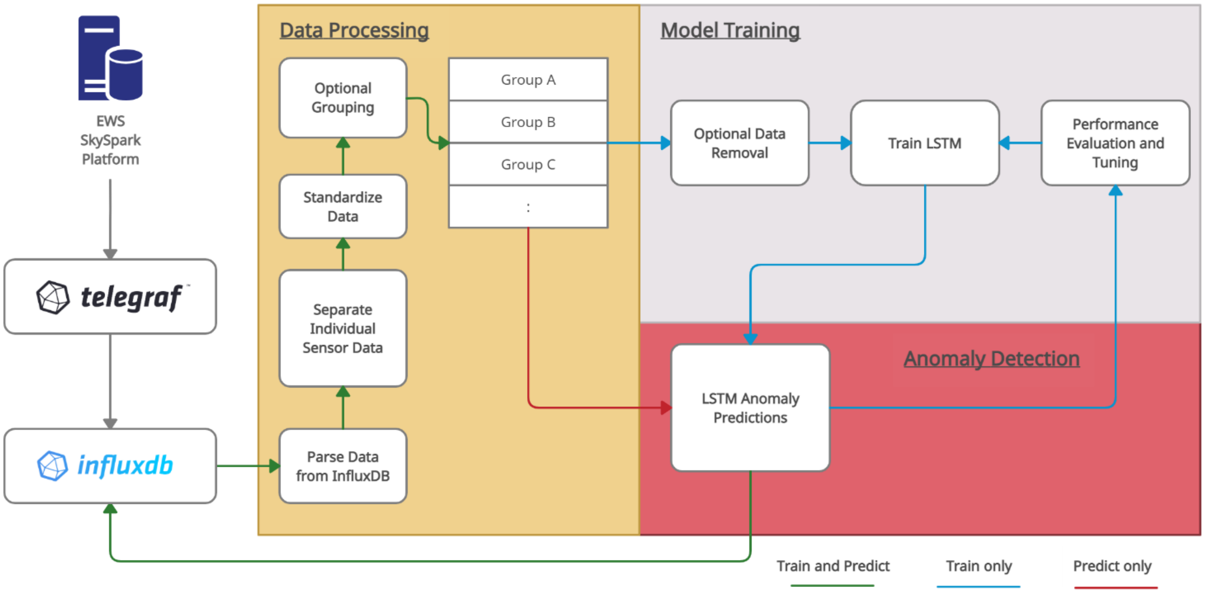


FIGURE 8: MODEL PIPELINE SCHEMATIC

4.3.1 – Data Processing

The data processing component of the model pipeline applies to both model training and model prediction. Data are queried from InfluxDB, parsed and separated into data frames for individual sensors, and each sensor data frame is standardized. Sensors can also be optionally grouped. Grouping allows multiple sensors to use a single model or allows a new sensor with minimal data (insufficient for model training) to use an existing model. Note that, while the pipeline includes functionality to allow this sensor grouping, the performance of this was not tested in the study.

4.3.2 – Model Training

The model training section of the pipeline only applies to the training script described in Section 3.2. The pipeline includes removing any data manually labelled True (anomalous) in the TRAINING_ANOMALY measurement manual_anomaly tag. This removal is unnecessary for a typical sensor with small periods of anomalous data but provides functionality if there are large periods of erroneous data at a sensor. For example, if a sensor has data from March to July but the data in March to May are erroneous, these data can be manually flagged such that they are removed from model training. This can be important as the LSTM-ED is trying to learn the historical pattern of sensor data.

After any manual_anomaly flagged data are removed, the LSTM-ED is trained on the remaining data. A default window size of 15 data points is used in training based on the model testing described in Section 5. The window size can be adjusted for a sensor as required. It is recognized that this is a low number of points for the LSTM-

ED; however, it was found to work well in this study and is similar to the window size of 10 data points used in the LSTM-ED by Liu et al (2020).

The model is automatically saved in an HDF5 file after training. The schematic in Figure 9 also shows a loop from model training to LSTM anomaly predictions, to performance evaluation and tuning, and back to model training. This evaluation loop is not an automated process and is just shown to indicate that tuning may be required if the default model values do not provide the required performance.

4.3.3 – Anomaly Detection

The anomaly detection section of the pipeline applies the trained LSTM models to the sensor data. The model results are compared with the actual data as described in Section 4.2 (using next point prediction or sequence reconstruction). Data is flagged as anomalous if the error between the model output and actual data exceeds a selected threshold.

In the model training script, anomaly detection is applied to all data on record for a sensor with the results written to the TRAINING_ANOMALY measurement model_anomaly tag. In the anomaly prediction script (used for near real-time detection) this is only applied to the period of recent data read from InfluxDB and results are written to the PREDICT_ANOMALY measurement realtime_anomaly tag.

This study used the following methods to evaluate a data point:

- Next Point Prediction: absolute error between the predicted data point and the actual data point.
- Sequence Reconstruction: maximum of the mean error and the mean absolute error between the input window data points and the reconstructed window data points. Note that while the full data window is used in the evaluation, the anomaly identification only applies to the most recent data point in the window.

This study used a default threshold value of the 99.5th percentile of training error values. This was found to work reasonably well but did require sensor specific adjustments to provide good anomaly identification performance. Adjustments were made by applying a multiplier (default = 1) to the 99.5th percentile value. Ideally a more robust anomaly detection threshold method could be identified and used as a default setting for the model. The multiplier values used for the sensors in this study are available in Appendix C and D. This threshold selection step represents the highest user effort within the framework (although note that changing the threshold does not require retraining of models).

One issue noted during model training and related to the threshold was the non-deterministic nature of the LSTM neural network. Models retrained using the same data can result in slightly different models and therefore produce different values. This results in potentially needing to set a different 99.5th percentile threshold multiplier for anomaly identification. Setting a random seed was investigated but did not resolve the issue and a quick web search indicates that making TensorFlow models deterministic does require effort and is affected by the GPU. It is also anticipated that instability could exist when retraining the model with additional data. This will need to be resolved with additional study and represents an issue for automation due to the level of effort required to set the threshold each time models are retrained. This could also potentially be resolved using a more robust threshold selection method that is less sensitive to error values. This could include using standardization of error which may reduce the sensitivity of the selected threshold value, or using the expected distribution of error within a window to flag anomalous data instead of just setting an error threshold value.

4.4 – Additional Model Comments

The LSTM-ED provides a model that should be applicable to a variety of sensors and was a goal of the project. However, it should be noted that there may be models or rules that can provide better anomaly detection performance on the specific CEC boiler sensors used in this study. The LSTM-ED model is not considered a replacement for any specific and better performing models or rules for individual sensors or systems. Instead, this model/framework represents a general approach that could be put in place parallel to existing models or rules, or useful for sensors without any existing anomaly detection system. It should also be relatively easy to integrate additional models or rules within the pipeline.

The model discussed above also uses a univariate sensor approach and does not consider data at multiple sensors. A system (as opposed to sensor) based anomaly detection approach may be more appropriate in some cases. For example, a piece of equipment with multiple sensors may require anomaly detection considering the relationship between sensors. The model in this study could potentially be used as part of a system-based detection approach.

5 – Anomaly Detection Testing

5.1 – Model Testing Approach

The LSTM-ED model was tested on the CEC sensors. These sensors are a selection from the groupings noted in Section 2.2 and include a variety of data patterns and anomalies. Streaming from the SkySpark platform to InfluxDB was not available during the project and these data had to be manually downloaded from the SkySpark platform user interface which is a time-consuming process. It is recognized that this represents a small subset of the 72 CEC boiler sensors available in SkySpark but was considered a reasonable representation for this study.

The model testing was split into two phases. Phase 1 included testing five of the ten sensors. As the model uses an unsupervised approach and anomaly labels are not available, a manual anomaly labelling process was completed on these five sensors as described in the following section. The labels were used to help assess performance of the model. This first testing phase was used to test various input data window sizes and compare the next point predictions versus sequence reconstruction methods.

A second phase of model testing was completed using the remaining five sensors. These sensors were not manually labelled. The results of Phase 1 testing were used for Phase 2 model setup. The purpose of Phase 2 was to test anomaly detection on additional sensors and assess if the best model setup identified in Phase 1 still provided good results.

5.2 – Sensors Tested

Model testing of the LSTM-ED was completed on the subset of ten CEC boiler sensors described in Table 2. These sensors were selected as they provide a selection from each sensor group described in Section 2.2, have varying data intervals, and provide a variety of data patterns and anomalies. Sensor data were divided into model train/test sets using an 60/40% split (note that the split is based on number of data points and not time period). Figures showing the sensor data are provided in Appendix B.

TABLE 1: CEC SENSORS USED IN MODEL TESTING

Sensor	Start	End	Interval (minutes)	Testing Phase	Description
HW Main Meter Power	Jan-2020	May-2021	2	1	Large seasonal changes. Most anomalies are spikes in data.
HW Main Meter Entering Water Temperature	Nov-2016	May-2021	15	1	Small seasonal changes. Periods of data in summers 2017, 2018, 2019 with different patterns. Most anomalies are spikes in data.
HW Main Meter Flow	Dec-2019	May-2021	1	1	Large seasonal change. Most anomalies are spikes in data.
Boiler B-1 Gas Pressure	Jan-2020	Apr-2021	15	1	Small seasonal changes. Most anomalies are single point values.
Boiler B-1 Exhaust O_2	Jan-2020	Apr-2021	15	1	Data are constant for long periods with jumps between periods.
Boiler B-2 Exhaust O_2	Dec-2019	June-2021	7	2	Many high point outliers. Small seasonal change. No pattern to the data, mostly noise. Long periods where the data provide flatline values.
Feeder 60L56 Real Power	May-2020	June-2021	1.5	2	Small seasonal changes. Several large event anomalies and several flatline zero value anomalies.
Boiler B-2 Gas Pressure	March-2020	June-2021	15	2	Small seasonal changes. Most anomalies are spikes in data.
Boiler B-1 Power	Dec-2019	June-2021	2	2	Small seasonal changes. Many flatline zero value anomalies. Several high point outliers.
Process System Pump Speed	March-2020	June-2021	15	2	Larger seasonal changes. Most anomalies are spikes in data.

Manual anomaly labelling was completed on five sensors to support the performance assessment for Phase 1 testing. Records of anomalies for the sensors were not available and the labelling process was based on a visual assessment of data that appeared abnormal. This could include large spikes, flatline data, or irregular patterns based on surrounding data. This process was subjective but still useful for model assessment without the availability of anomalous data records that could provide a quantitative evaluation. Examples of manually identified anomalies are shown in the figure below. Appendix B provides a table describing typical anomalies labelled for each sensor and includes figures showing the sensor data with labels.

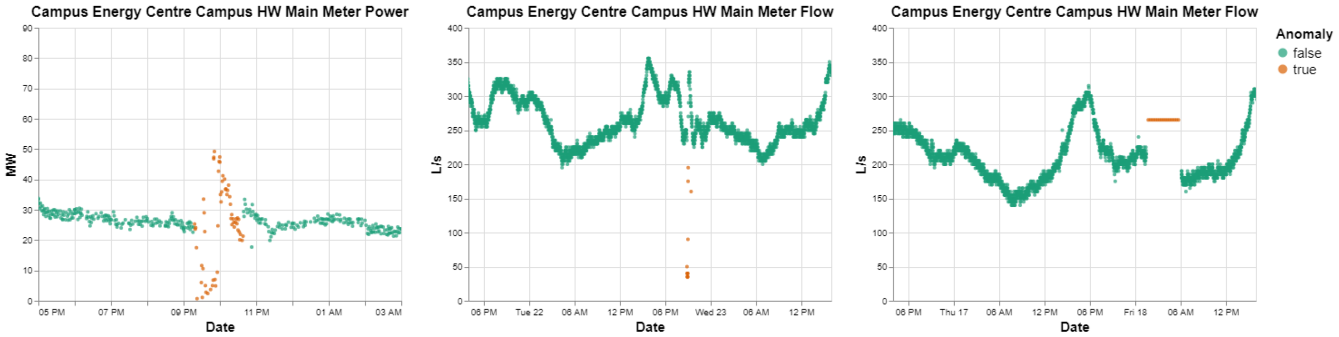


FIGURE 9: PHASE 1 SENSOR DATA - ANOMALY EXAMPLES

5.3 – Performance Criteria

Anomalous data labels are not available for the sensors and the LSTM-ED model uses an unsupervised approach. The initial purpose for manually labelling anomalous data was to provide some level of quantitative evaluation for the project study. However, it was recognized that labelling process was subjective, and it would be difficult to assess model performance quantitatively using the manual labels. It was also observed that the model identified data that while not initially identified and manually labelled as anomalous, did appear to be potentially abnormal data.

Accordingly, it was decided to focus on a qualitative assessment instead of trying to use quantitative measures. This was considered reasonable given the unsupervised modelling approach and the uncertainty in the manual labelling process. The following qualitative criteria was used to assess model performance and allow comparison between model tests:

- **Event Identification**: If the anomaly detection model is identifying events that are considered anomalous. This does not include consideration for the proportion of data points within the event flagged, but just whether some data within the event is identified or not. A poor anomaly detection model would miss many events.
- **Event Coverage**: This includes consideration for the proportion of data identified in an anomalous event Note that it is sometimes difficult to determine the duration of an anomalous event even visually. This performance measure is to provide a general understanding of whether the model is flagging what appears to be the majority of the event, or just one or two points within the event.
- **Initial Event Detection:** How early the model identifies anomalous data point within the event. Ideally the model would label the first point as anomalous.
- **False Anomaly Identification:** If the model is incorrectly identifying normal data as anomalous.

In summary, the qualitative measures are aimed at assessing if the model is identifying events that appear to be anomalous, how much of the event is identified, how early it is identifying the event, and if normal data is being incorrectly labelled as anomalous. These measures were assessed using interactive visualizations of the data. An example of a qualitative assessment is provided in the following table.

TABLE 2: QUALITATIVE MODEL ASSESSMENT EXAMPLE

Test No.	Sensor	Event Identification	Event Coverage	Initial Even Detection	False Anomaly Identification
3	HW Main Meter Power	Most manual labelled events identified	Identifies majority of event but sometimes identifies normal values after event	Can be delayed by a couple points if event is less obvious	Minimal on smaller windows, moderate on larger windows

It is recognized that this model assessment method is subjective, but it was found to provide a reasonable understanding of model performance and comparison of models for this study.

5.4 – Phase 1 Test Results

Phase 1 testing used the sensors with manual labelled anomalies to asses he following model variations:

- Next Point Prediction vs Sequence Reconstruction
- Window Size: this varied between 4 and 120 data points (corresponding to 15 minutes to 30 hours depending on the sensor recording interval)
- Anomalous Data Removal: removal of data identified as anomalous to understand the impact on model performance

The model tests and qualitative results are summarized in Tables C1 and C2 in Appendix C. The appendix also provides figures showing model results. The best performing tests for each sensor are summarized in Table 3.

TABLE 3: PHASE 1 MODEL TESTS

Sensor	Type	Best Window Size (Data Points)	Window Size Approx. Duration	Data Removed from Training	Qualitative Evaluation Summary
HW Main Meter Power	Sequence Reconstruction	30	30 min	No	Identifies most manual labelled events and the majority of data in the event. Sometimes labels points after event. Performs slightly better than 15-point window.
HW Main Meter Entering Water Temperature	Sequence Reconstruction	15	3 hr 45 min	No	Identifies most manual labelled events and the majority of data in the event. Sometimes labels several points after event. Smaller duration windows do not perform as well.

HW Main Meter Flow	Sequence Reconstruction	15	15 min	No	Identifies most manual labelled events and the majority of data in the event. Sometimes labels points after event.
Boiler B-1 Gas Pressure	Sequence Reconstruction	15	3 hr 45 min	No	Identifies most manual labelled events and the majority of data in the event. Sometimes labels points after event.
Boiler B-1 Exhaust O_2	Sequence Reconstruction	4	1 hr	No	This sensor has a very different pattern from other sensors (constant data with infrequent jumps in data). The shorter window works best as it still identifies events but minimizes the amount of data labelled anomalous after the event.

The results of Phase 1 testing indicate that:

- Sequence reconstruction performs better than next point prediction for all sensors. Both methods are good at identification of anomalous events, but next point prediction typically only identifies a subset of data points within an event. However, it should be noted that sequence reconstruction sometimes identifies normal data after an event as anomalous.
- A shorter data input window size appears to result in good event identification performance while having less false positives compared with longer windows. A window of 15 points appears to provide a good default value. Note that this may be due to the anomaly detection threshold method used as discussed in Section 4.3.3. A more sophisticated threshold method may provide better identification and the use of longer data windows.
- Removal of anomalous data from model training is not necessary. Only long period of erroneous data should be removed from training.

Examples of the difference in performance is shown in the following figure.

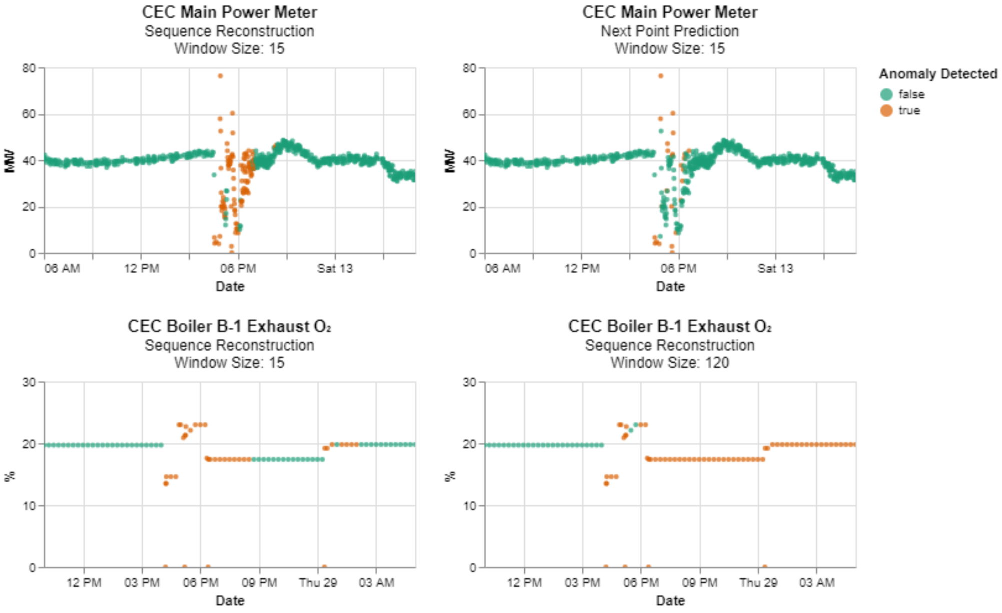


FIGURE 10: PHASE 1 MODEL TEST EXAMPLES

Overall, the Phase 1 testing results indicated that the model performed well in identifying anomalies for these five sensors. A 15-point input data window with sequence reconstruction provides a good default model setting. Sequence reconstruction can result in identifying normal data as anomalous at the end of an event but this was considered better than next point prediction which typically only identifies the start of an event.

5.5 – Phase 2 Test Results

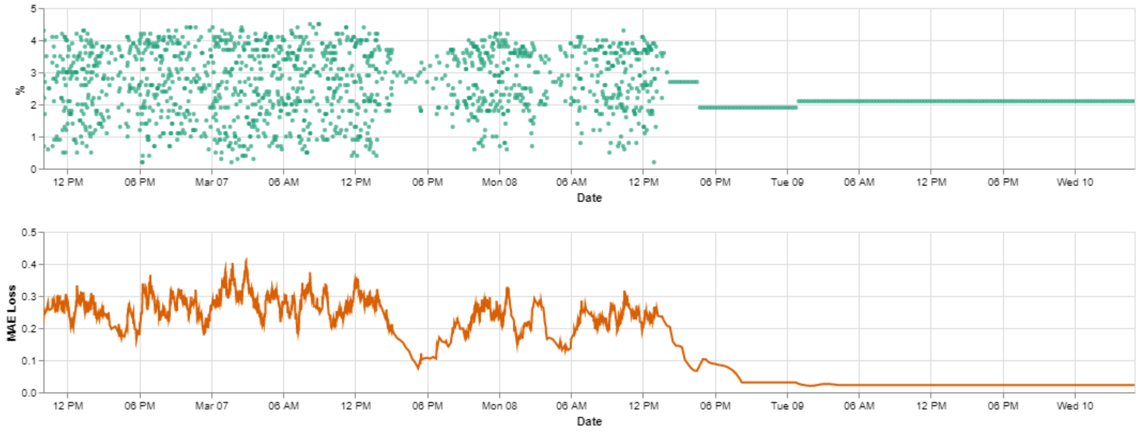
Phase 2 testing used sequence reconstruction and a 15-point data input window size on the five remaining sensors that were not manually labelled. A summary of the tests is provided below.

TABLE 4: PHASE 2 MODEL TESTS

Sensor	Type	Best Window Size (Data Points)	Window Size Approx. Duration	Data Removed from Training	Qualitative Evaluation Summary
Boiler B-2 Exhaust O_2	Sequence Reconstruction	15	1 hr	No	Identifies outlier point anomalies. Does not identify flatline data.
Feeder 60L56 Real Power	Sequence Reconstruction	15	15 min	No	Identifies majority of anomalous events. Only identifies start of flatline zero data.
Boiler B-2 Gas Pressure	Sequence Reconstruction	15	3 hr 45 min	No	Identifies majority of anomalies.
Boiler B-1 Power	Sequence Reconstruction	15	30 min	No	Identifies majority of anomalies. This sensory turns on and off frequently making data hard to predict. Only identifies start of flatline zero data.
Process System Pump Speed	Sequence Reconstruction	15	3 hr 45 min	No	Identifies majority of anomalies.

The Phase 2 qualitative test results are provided in Appendix D. The tests indicate the model achieved good event identification performance on four of the five sensors using the model parameters from Phase 1. However, flatline zero value events that occur for Feeder 60L56 Real Power and Boiler B-1 Power did not have good event coverage. The model was able to label the start of these events but not subsequent data points. Examples of anomaly detection events for the sensors are provided in Appendix D.

The model did not perform well for sensor Boiler B-2 Exhaust O_2 which has periods of flatline values that were not detected. Review of the sensor indicates that there is minimal pattern within the data and a large amount of noise. Inspection of the LMST-ED shows high reconstruction error for this noisy data and the flatline periods have better reconstruction. This behavior was not observed for other sensors where data had patterns/trends and flatline periods were identifiable from the reconstruction error. A more sophisticated threshold method may be able to provide better identification for this type of data (for example assessing the distribution of error within a window of data and flagging data outside of expected distribution bounds, instead of using a maximum error threshold).



Top figures shows sensor data, bottom figure shows LSTM reconstruction error.

Figure 11: Boiler B-2 Exhaust O_2 LSTM-ED Reconstruction Error Example

5.6 – Testing Summary

The Phase 1 tests indicate that the LSTM-ED model with sequence reconstruction using a window of 15 data points has good performance using the qualitative assessment. However, it was observed that the default threshold based on the 99.5th percentile required modification for each sensor and represents the highest user effort within the anomaly detection framework. Ideally a more robust anomaly detection threshold method could be identified and used as a default setting for the model.

The Phase 2 tests indicate the model performed reasonably well but had poor complete event coverage performance on two sensors for flatline zero value data and was unable to detect flatline data for sensor Boiler B-2 Exhaust O_2. It is believed that model performance can be improved with a more sophisticated threshold identification rule for these types of anomalies.

The testing completed in this study provides a qualitative assessment of the model as a good initial step in understanding model performance. A more rigorous model performance assessment could be completed using a quantitative approach. This would require identifying sensors with labelled anomalous data or selecting sensors where a simple model of rule could be used to identify anomalous data for comparison.

6 – Dashboard and Notification System

A dashboard and notification system were implemented in the test InfluxDB environment described in Appendix A. This should be considered preliminary and additional work on investigating/implementing functionality should be completed.

6.1 – Dashboard

InfluxDB version 2 provides built-in dashboard functionality. The dashboard allows visualization of data stored in the database and templates can be saved for future use. Grafana provides an alternate option to InfluxDB and appears to currently have more functionality with many available plug-ins. Grafana can easily be connected to InfluxDB but is a separate program. UDL indicated they would prefer to use the built-in InfluxDB dashboard to store and view data within a single platform. Accordingly, the InfluxDB dashboard was explored for this project.

The dashboard discussed with UDL and created for the project provides a simple interface to view sensor data with anomalous data highlighted. The user can select if they want to view the manual labelled (TRAIN_ANOMALY measurement manual_anomaly field), model labelled (TRAIN_ANOMALY measurement model_anomaly field), or real-time labelled (PREDICT_ANOMALY measurement realtime_anomaly field) anomalies on the graphs. Figure A4 in Appendix A provides an example figure of the dashboard with the Phase 1 test sensors.

The dashboard implemented in this project provides a simple solution directly integrated with InfluxDB. More complex options that provide dropdowns or use of multiple tabs for sensor group views could be investigated. There were also limitations encountered with the InfluxDB dashboard. InfluxDB dashboards do not include the ability to set point colors and sizes and only the scatter plot format was useable with the data. Grafana should be considered in the future if additional functionality or formatting is required.

6.2 – Notification System

A simple notification system was also setup with the test InfluxDB environment using built-in InfluxDB functionality. Notifications can be configured through the InfluxDB user interface although it is understood that notifications can be set directly using flux tasks. This study tested InfluxDB notifications sent to Slack. Notifications were setup to provide messages when any of the five CEC sensors used in the Phase 1 tests had True (anomalous) values written to the PREDICT_ANOMALY measurement realtime_anomaly tag (see Figure A5 in Appendix A for an example message).

This component of the real-time detection framework requires additional testing and configuration before it could be used in an operational setting. This study was limited to testing the capability of InfluxDB to send notifications on anomalous data and was found to be successful. More sophisticated notification systems could include suppressing notifications when equipment is not operational or only providing notifications when multiple sensors detected anomalous data. The capability of the InfluxDB notification system to do this was not assessed.

It should also be noted that the anomaly detection framework in this study does not identify or provide notifications on missing data or long duration intervals when data are not provided to InfluxDB. These may represent operational or sensor issues, but the model applied in this study currently does not detect this. It should be possible to implement this functionality within the anomaly detection framework.

7 – Conclusion and Recommendations

7.1 – Conclusion

This project resulted in a proposed approach for a near real-time anomaly detection system with UDL's database. The approach includes a framework for anomaly detection model training and prediction with InfluxDB using open-source software. Model training is completed by querying sensor data from InfluxDB on an infrequent basis (for example monthly), training, and saving the models. Anomaly detection occurs on a continuous basis by reading recent data from InfluxDB, loading and running previously trained models, and writing results back to InfluxDB. This framework was implemented in a test InfluxDB environment.

The anomaly detection model selected is a LSTM-ED with a short-duration window using sequence reconstruction in an unsupervised approach. The LSTM-ED provides a general model that should be applicable to a variety of sensors. The model was tested on ten CEC boiler sensors available from the EWS SkySpark platform. A qualitative performance assessment was completed as anomaly labels for these sensors were not available during the study. The results indicate good initial performance on the subset of sensors. There were several data patterns identified that the model had trouble detecting but it is believed performance can be improved using more sophisticated threshold identification rules. One issue identified was the non-deterministic nature of the LSTM and the resulting impact on setting the threshold value (discussed in Section 4.3.3). This will need to be investigated and may also benefit from a more sophisticated threshold identification rule that is less sensitive to the error values.

A simple dashboard and notification system were also implemented with the anomaly detection framework in the test InfluxDB environment. The dashboard is built directly in InfluxDB and provides a simple display of sensor data with anomalous flagged data highlighted. The notification system also uses built-in InfluxDB functionality and was configured to send notifications to a Slack channel for data predicted as anomalous.

7.2 – Recommendations for Future Study

This study resulted in a proposed open-source approach for near real-time anomaly detection that can be used with InfluxDB. The following sections discuss improvements that should be considered to understand the capability for potential operational use. It is also expected that modifications to the framework/code may be required when implementing with the online version of InfluxDB as testing was only completed in the environment described in Appendix A.

Ideally, the detection system could ultimately be implemented and used to provide campus and building managers with real-time or near real-time notifications of potential issues in systems operations reducing operational costs, downtime, and unexpected maintenance.

7.2.1 – Anomaly Detection Framework

The anomaly detection framework in this study operates by reading recent data from InfluxDB, providing predictions, and writing results back to InfluxDB. This can be implemented on a near real-time basis (for example, every 1 minute or every 5 minutes). Improvements to the current approach include providing a method to avoid re-reading the same data from InfluxDB each time the prediction script is executed. This could either be done by writing data to a temporary file or running the program continuously and keeping the data in memory.

A more real-time approach includes providing predictions in-line with Telegraf as discussed in Section 3.1.1. There are several challenges associated with this that would need to be considered including configuration of Telegraf to send the data to an external program, implementing streaming error handling, and keeping the required window of data for predictions in memory.

7.2.2 – Anomaly Detection Model

The LSTM-ED model was selected as a model believed to provide good generalization to a variety of systems and sensors based on recent studies. The results indicate good initial performance on a subset of CEC boiler sensors, but the following additional studies should be considered:

- Improvements on the LSTM-ED anomaly identification threshold rule described in Section 4.3.3. The method used in this study is simple and a more sophisticated method could be used. The threshold selection also represents the highest amount of user effort within the framework. A better default threshold selection method would provide a more automated framework. Improvement of the threshold method could improve anomaly detection with minimal changes to the pipeline.
- Assessing the stability of model anomaly detection when retraining with additional data as discussed in Section 4.3.3.
- Different types of models (for example, bi-LSTM or ARIMA) were not tested in this study based on the limited project timeline. Testing additional models may result in identification of models with higher performance or provide a simpler approach but with similar performance.

- This study focused on open-source anomaly detection models. The use of service-based anomaly detection models can also be considered (such as Microsoft's anomaly detector available through Azure). This could potentially provide an easier implementation than the framework used in this study.
- The study did not have time to assess the use of additional models or anomaly detection rules that could be used in conjunction with the LSTEM-ED. It is expected there are rules and models that have high anomaly detection performance with specific systems and sensors. It would make sense to apply these (where they are known) with this more general model.
- The model pipeline presented in Section 4.3 has the capability to group multiple sensors for use with a single trained model. This could allow a sensor with minimal data to use a model trained on other sensors. This functionality could also allow fewer models to be trained instead of creating an individual model for each sensor. This grouping should be viable on sensors that are similar but testing of this was not completed.
- The LSTM-ED in this study only considers univariate data. Modification of the LSTM-ED to use multi-variate data could be considered.

7.2.3 – Model Performance

A qualitative performance assessment was completed on ten CEC boiler sensors in this study. A more rigorous model performance assessment could be completed using a quantitative approach. This would require identifying sensors with labelled anomalous data or selecting sensors where a simple model or rule could be used to identify anomalous data.

The study also only looked at a subset of boiler system sensors from a single building. Additional testing should be completed on additional sensors/systems to understand the generalizability of the approach. This could be done in an online approach by implementing the anomaly detection model on additional sensors and monitoring performance.

7.2.4 – Dashboard and Notification System

The dashboard and notification system assessed in this study are simple implementations using built-in InfluxDB functionality. The purpose of the assessment was to understand viability and it is expected that additional effort would be required to improve the dashboard and provide a useful notification system.

If it is found that the built-in InfluxDB dashboard does not provide the functionality required for a useable interface, it is recommended that Grafana is further explored. Grafana provides easy integration with InfluxDB and currently appears to have more functionality.

The notification system tested in this study provides notifications to a Slack channel when anomalous data are identified in near real-time anomaly. To use the notification system for an operational system likely requires bringing the anomaly detection system online and monitoring performance before the notification system is turned on for a sensor to ensure the detection system is providing meaningful results. As discussed in Section 4.3, there are several adjustments that can be made to the default model parameters on an as-needed basis including the detection threshold and input data window size.

More sophisticated settings may also be required such as only triggering a notification if multiple sensors detect an anomaly or turning off notifications when equipment is offline. These were not explored in this study but may be required to improve the operational usefulness of a notification system.

7.2.5 – System Anomaly Detection

The anomaly detection framework in this study uses a univariate sensor approach and does not consider the relationships between sensors. The LSTM-ED model tries to learn the historical pattern of data for a sensor and identify data that does not follow the normal pattern. There may also be interest in anomaly detection on a system instead of sensor basis. An example could be identification of potential boiler malfunction versus identification of anomalous data at individual boiler related sensors. The LSTM-ED anomaly detection model could help support identification of system malfunctions/anomalies but was not considered in this study. This task is likely system specific requiring an understanding of the relationships between sensors and input from domain experts. For example, a sequence of sensor anomalies may be meaningful and indicate a potential malfunction, while identification of an anomaly at a single sensor may not indicate an issue depending of the system.

8 – Acknowledgements

We would also like to thank and acknowledge support and contributions from Urban Data Lab (Mike Kennedy, Jiachen Wei, Ibrahim El-chami), UBC Energy and Water Services, Technical Safety BC, and UBC (Dr. Scott Fazackerley, Dr. Irene Vrbik, Debangsha Sarkar).

9 - References

Brownlee, H., 2020. *Long Short-Term Memory Networks with Python.* v1.7.

Cook, A.A., Misirh, G., Fan, Z., 2020. *Anomaly Detection for IoT Time-Series Data: A Survey.* IEEE Internet of Things Journal. Vol. 7, No. 7. July 2020. 6481-6494.

Geron, A., 2019. *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow.* Second Edition. Published by O’Reilly Media Inc.

Influxdata, 2021a. *InfluxDB.* Retrieved from https://www.influxdata.com/products/influxdb/ (accessed June 8, 2021).

Influxdata, 2021b. *influx-client-python*. Retrieved from https://github.com/influxdata/influxdb-client-python/ (accessed June 8, 2021).

Influxdata, 2021c. *Telegraf.* Retrieved from https://www.influxdata.com/time-series-platform/telegraf/ (accessed June 8, 2021).

Liu, Y., Pang, Z., Karlsson, M., Gong, S., 2020. *Anomaly Detection Based on Machine Learning in IoT-based Vertical Plant Wall for Indoor Climate Control.* The International Journal of Building Science and its Applications. Vol. 183. October 2020. 107212.

Ren, H., Xu, B., Wang, Y., Huang, C., Kou, X., Ming, T., Yang, M., Tong, J., Zhang, Q., 2019. *Time-Series Anomaly Detection Service at Microsoft.* arXiv: 1906.03821. Retrieved from https://arxiv.org/abs/1906.03821/ (accessed June 8, 2021).

Riazi, M., Zaiane, O., Takeuchi, T., Maltais, A., Gunther, J., Lipsett, M., 2019. *Detecting the Onset of Machine Failures Using Anomaly Detection Methods.* Big Data Analytics and Knowledge Discovery. Springer International Publishing. 2019. 3-12.

Urban Data Lab (UDL), 2021. *About*. Retrieved from https://urbandatalab.io/about (accessed June 8, 2021).

Urban Data Lab (UDL), 2020. *UBC Building Data.* January 9, 2020. Retrieved from https://urbandatalab.io/news/building-energy-data (accessed June 8, 2021).

Keras, 2021. *Keras API Reference.* Retrieved from https://keras.io/api/ (accessed June 8, 2021).

Keras, 2020. *Timeseries Anomaly Detection using an Autoencoder.* May 31, 2021. Retrieved from https://keras.io/examples/timeseries/timeseries_anomaly_detection/ (accessed June 8, 2021).

Appendix A – Anomaly Detection Test Environment

An anomaly detection test environment was used for the study as UDL's InfluxDB instance was still being setup with SkySpark data during the project timeline. The test environment was setup using a local instance of InfluxDB created with Docker. The anomaly detection framework discussed in Section 3.2 was then implemented in this environment. This included model training, saving model files, and simulating anomaly predictions. A dashboard and notification system were also setup in the test environment. A detailed walk-through is provided in a Jupyter Notebook (test_env_demo.ipynb) within the project code repository. Comments are provided in the notebook where modifications have been made to the code that are specific to the test environment.

The following sections briefly discuss the steps completed in the test environment. It is recommended to review the notebook file to provide a more detailed understanding.

Steps 1 and 2 – Create Local InfluxDB and Populate with Data

The test environment was setup using a local instance of InfluxDB created with Docker. The docker-compose.yml file used for setup is available in the code repository. The test environment was then populated using sensor data manually downloaded from SkySpark. The data include the five CEC sensors used in Phase 1 testing (Section 5.4). The InfluxDB READINGS measurement tags and fields required for the anomaly detection framework noted in Section 3.3 were populated by uploading the data in csv format.

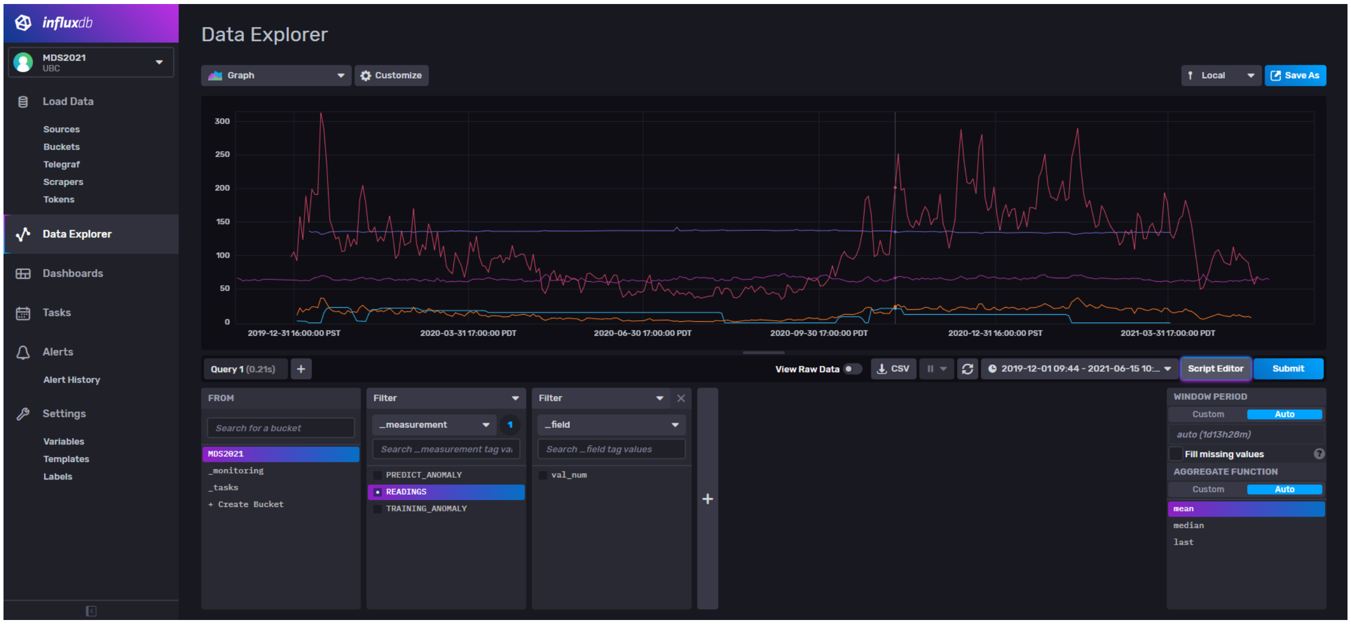


FIGURE A1: TEST ENVIRONMENT SKYSPARK BUCKET READINGS MEASUREMENT

Step 3 – Anomaly Detection Model Training

The Python script for anomaly detection model training was implemented in the test environment. This script would typically be run on an infrequent basis (such as monthly). Several modifications specific to the test environment were made to the code. Comments in the notebook were provided where this was done. The general steps include:

1. Data is read from InfluxDB and data frames are created for each sensor.
2. Any values labelled as True in the TRAINING_ANOMALY measurement manual_anomaly tag is removed from the data frames.
3. Sensor values are standardized and the standardizers are saved.
4. Data windows required for the LSTM-ED are created and the models are fit and saved.
5. Anomaly predictions from the LSTM-ED are written to InfuxDB to the TRAINING_ANOMALY measurement model_anomaly tag.

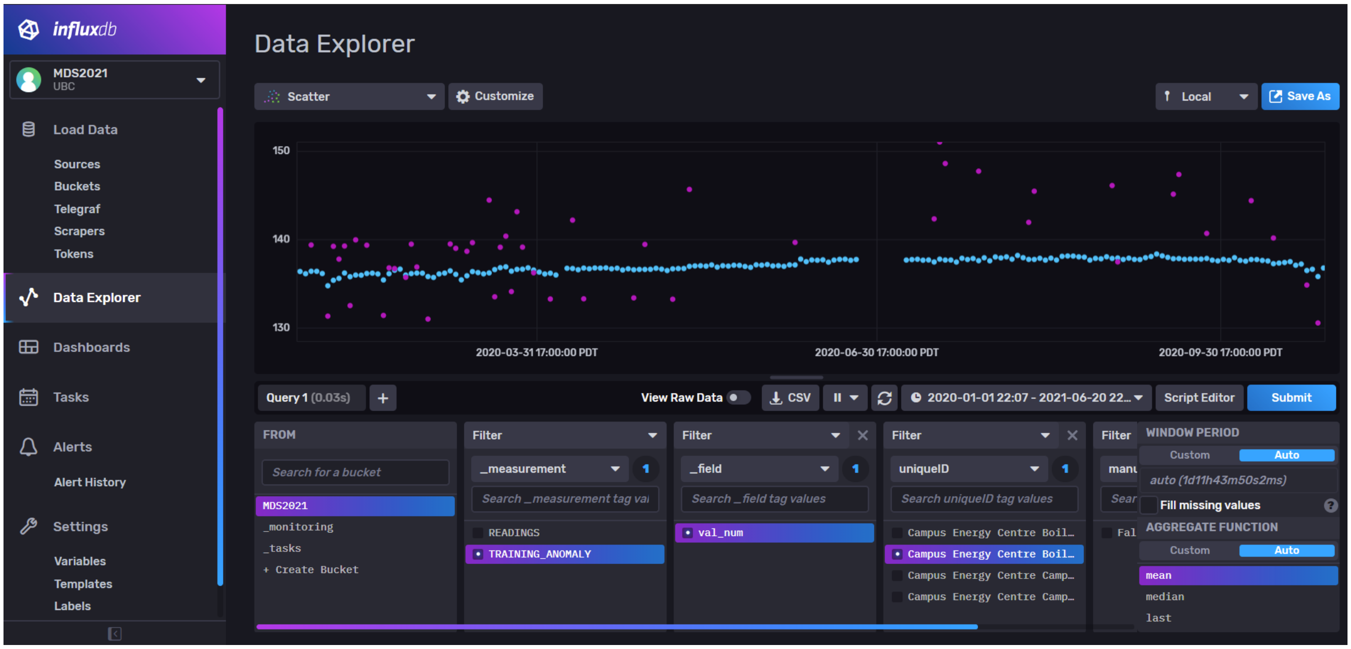
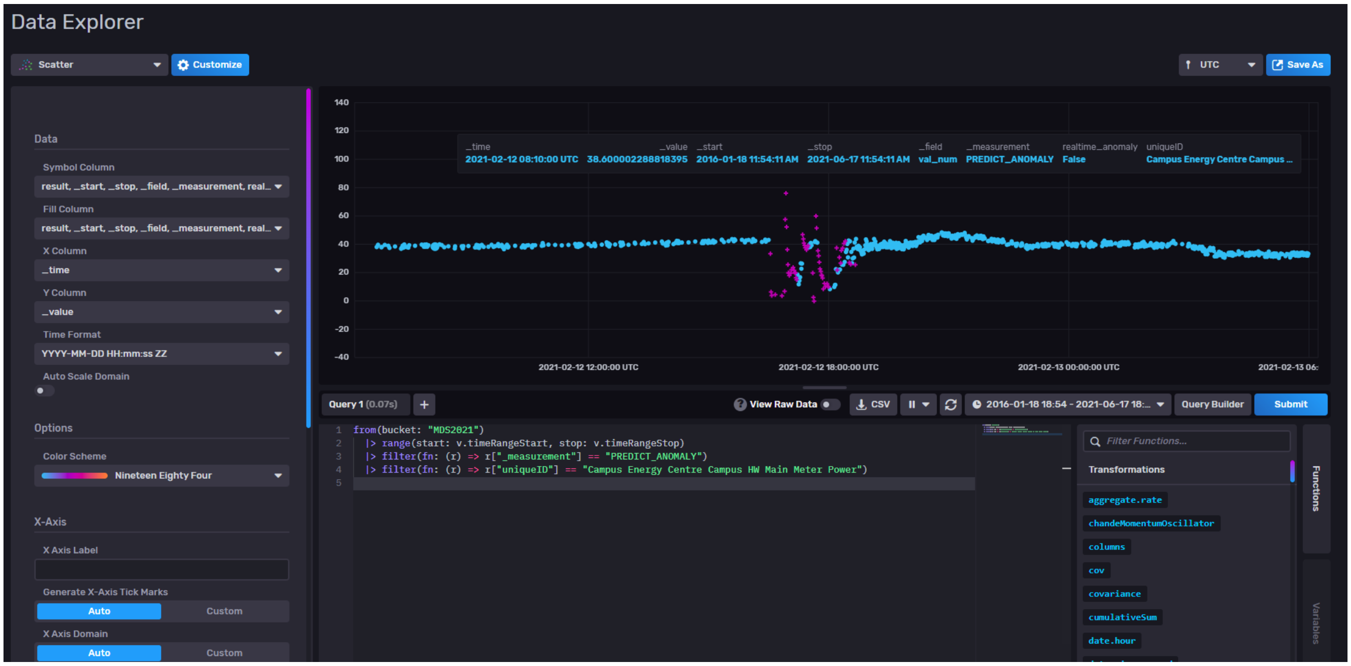


FIGURE A2: TEST ENVIRONMENT SKYSPARK BUCKET TRAINING_ANOMALY MEASUREMENT

Step 4 – Anomaly Detection Predictions

The Python script for anomaly detection prediction was implemented in the test environment. This script would typically be run on a frequent internal (such as every minute or every five minutes). Several modifications specific to the test environment were made to the code. Comments in the notebook were provided where this was done. The general steps include:

1. Data is read from InfluxDB and data frames are created for each sensor.
2. The standardizers from the model training step are loaded and sensor values are standardized.
3. The LSTM-ED models from the model training step are loaded, data windows required for the LSTM-ED are created, and the models are applied to the data to provide predictions.
4. Anomaly predictions from the LSTM-ED are written to InfuxDB to the PREDICT_ANOMALY measurement realtime_anomaly tag.



Data

Symbol Column: result, _start, _stop, _field, _measurement, real...

Fill Column: result, _start, _stop, _field, _measurement, real...

X Column: _time

Y Column: _value

Time Format: YYYY-MM-DD HH:mm:ss ZZ

Auto Scale Domain

Options

Color Scheme: Nineteen Eighty Four

X-Axis

X Axis Label

Generate X-Axis Tick Marks: Auto Custom

X Axis Domain: Auto Custom

_time	_value	_start	_stop	_field	_measurement	realtime_anomaly	uniqueID
2021-02-12 08:10:00 UTC	38.600002288818395	2016-01-18 11:54:11 AM	2021-06-17 11:54:11 AM	val_num	PREDICT_ANOMALY	False	Campus Energy Centre Campus ...

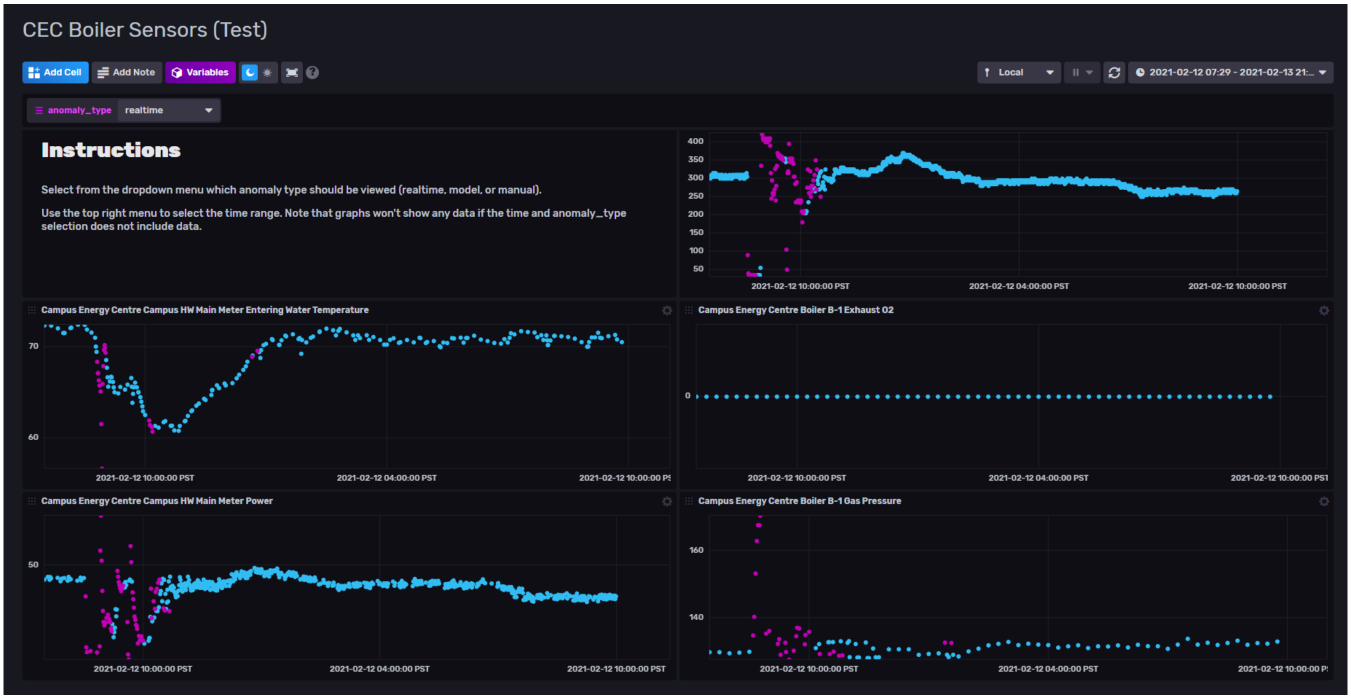
2016-01-18 18:54 - 2021-06-17 18:...

```
from(bucket: "MDS2021")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "PREDICT_ANOMALY")
  |> filter(fn: (r) => r["uniqueID"] == "Campus Energy Centre Campus HW Main Meter Power")
```

Figure A3: Test Environment SkySpark Bucket PREDICT_ANOMALY Measurement

Steps 5 and 6 – Dashboard and Notification System

The simple dashboard and notification system described in Section 6 of this report was created using the data available in the test environment from Steps 1 to 4. The dashboard was created by uploading a template (available in the code repository) to InfluxDB while the notification system was setup manually and configured to send notifications to a Slack channel for any data predicted as anomalous.



CEC Boiler Sensors (Test)
Add Cell
Add Note
Variables
Local
2021-02-12 07:29 - 2021-02-13 21:...
anomaly_type
realtime
400
350
300
250
200
150
100
50
2021-02-12 10:00:00 PST
2021-02-12 04:00:00 PST
2021-02-12 10:00:00 PST
Campus Energy Centre Campus HW Main Meter Entering Water Temperature
70
60
Campus Energy Centre Boiler B-1 Exhaust 02
0
Campus Energy Centre Campus HW Main Meter Power
50
Campus Energy Centre Boiler B-1 Gas Pressure
160
140

Instructions

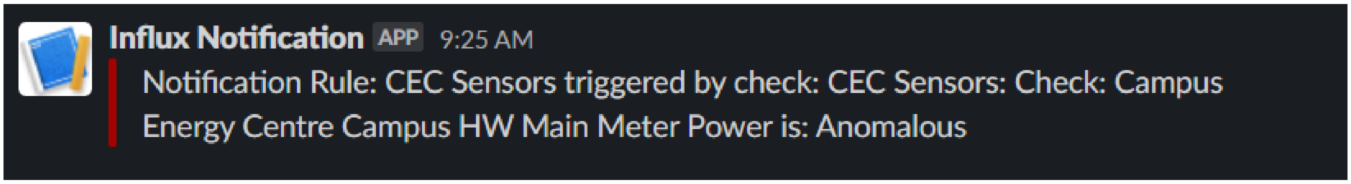
Select from the dropdown menu which anomaly type should be viewed (realtime, model, or manual).

Use the top right menu to select the time range. Note that graphs won't show any data if the time and anomaly_type selection does not include data.

FIGURE A4: TEST ENVIRONMENT DASHBOARD

Step 7 – Test Notification System

The notification system was tested by uploading anomalous tags for sensor Campus Energy Centre Campus HW Main Meter Power to InfluxDB in the PREDICT_ANOMALY measurement realtime_predict tag. This resulted in the following notification sent to the Slack channel.



Influx Notification APP 9:25 AM

Notification Rule: CEC Sensors triggered by check: CEC Sensors: Check: Campus Energy Centre Campus HW Main Meter Power is: Anomalous

FIGURE A5: TEST ENVIRONMENT NOTIFICATION

Appendix B – Additional Sensor Information

Figures B1 and B2 show the sensor records used in Phase 1 and Phase 2 testing, respectively. Both figures show the train/test splits used in model testing as vertical dashed lines and Figure B1 shows the manual labelled anomalies.

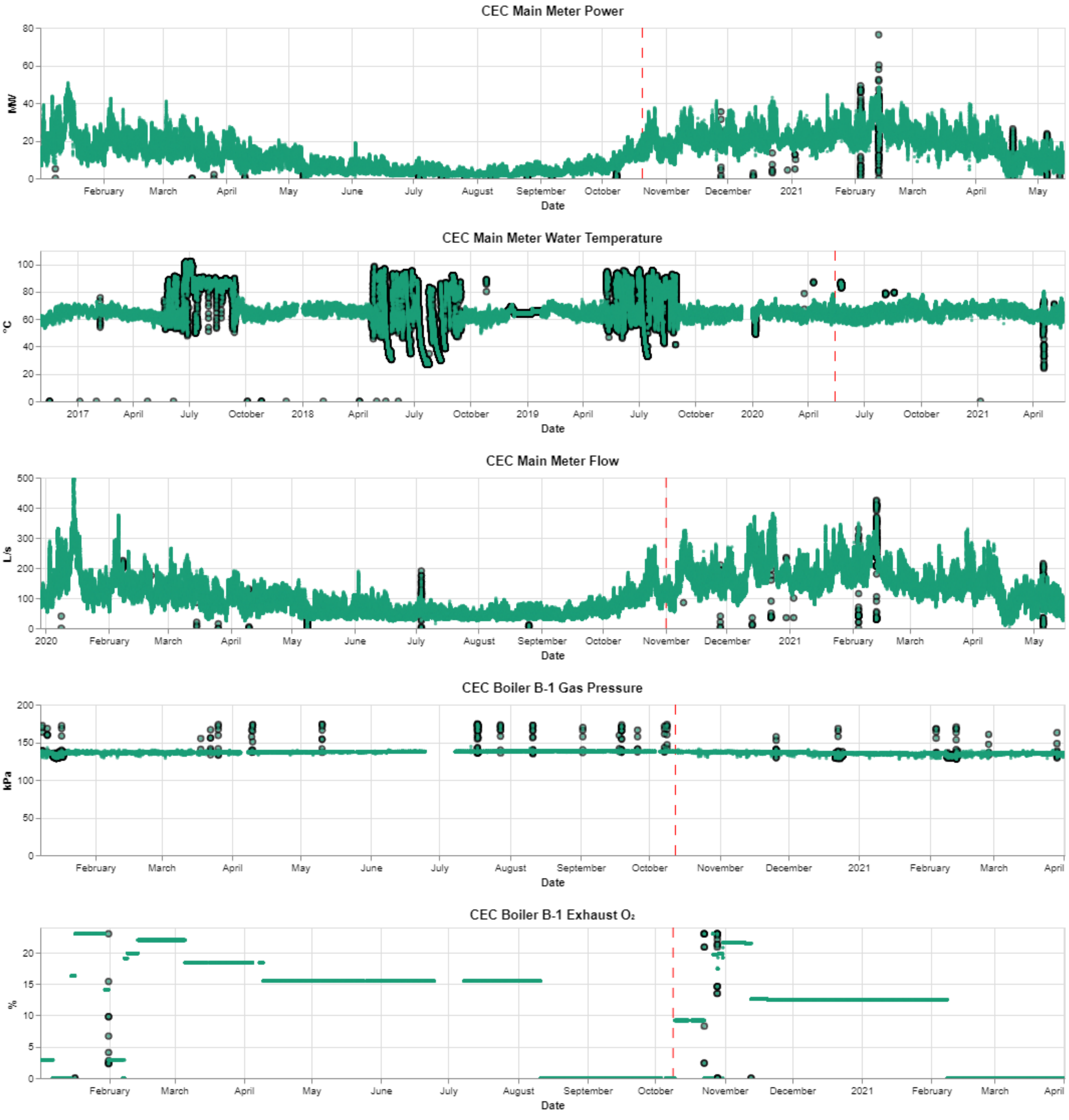


FIGURE B1: CEC SENSORS USED IN PHASE 1 TESTING

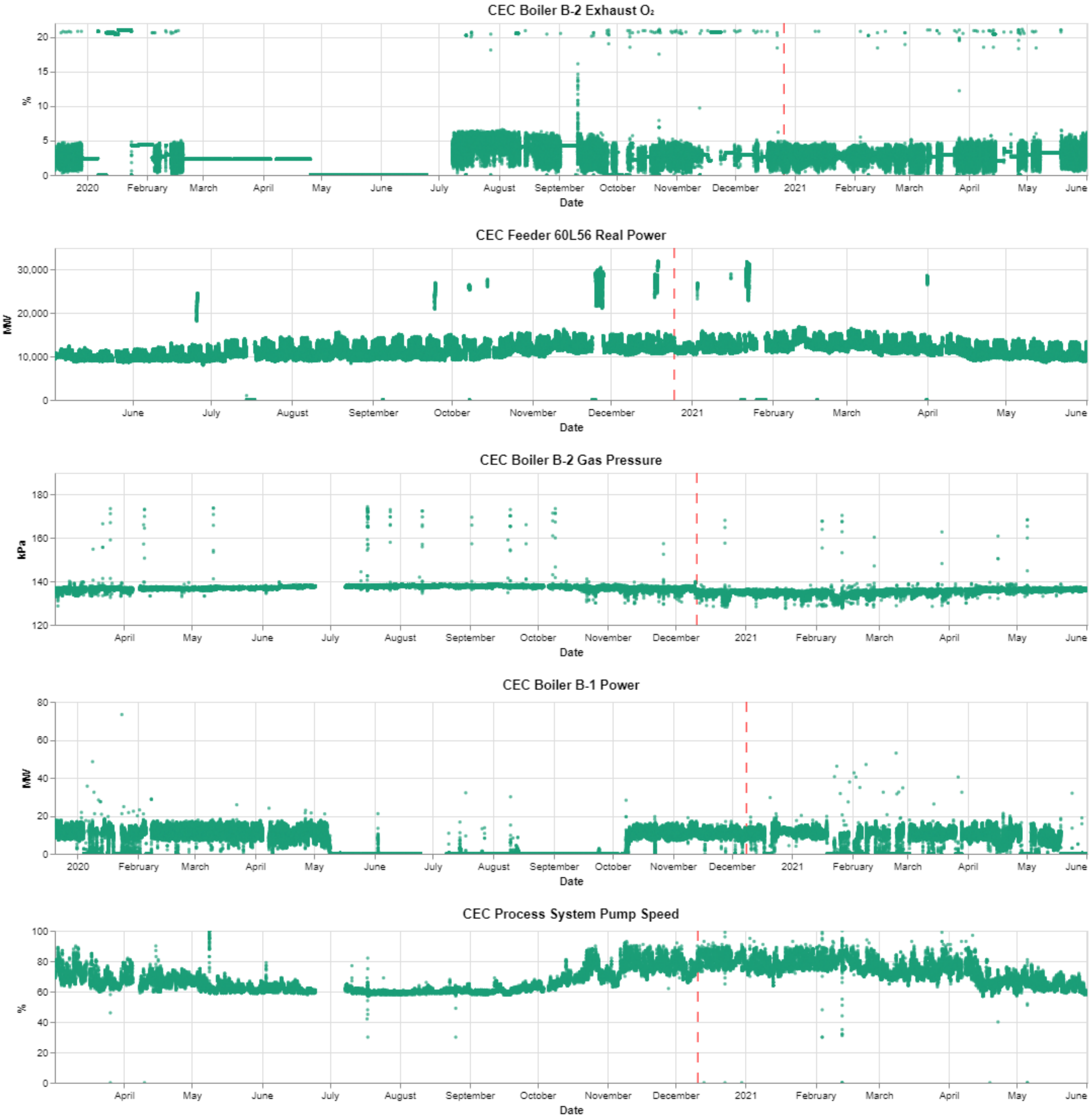


FIGURE B2: CEC SENSORS USED IN PHASE 2 TESTING

Manual anomaly labelling was completed on the five sensors used in Phase 1 testing to support model performance assessments. Records of anomalies for the sensors were not available and the labelling process was based on a visual assessment of data that appeared abnormal. This could include large spikes, flatline data, or irregular patterns based on surrounding data. A general description of the anomalies identified at the sensors is provided in Table B1.

TABLE B1: CEC SENSOR MANUAL ANOMALY LABELS

Sensor Name	Description of Anomalies
HW Main Meter Power	• Most identified anomalies are several data points that rapidly drop to zero or near zero. These may be due to operations but are infrequent. • Several rapid spikes that increase in value. These may be operational but are infrequent. • A period of data that sits well below the general trend of surrounding data.
HW Main Meter Entering Water Temperature	• Points with zero values that do not make sense for a temperature sensor. • Periods of data sitting high above the trend of surrounding data. • Flatline periods of data with identical values. • Several rapid drops in data. There are periods in the summer of 2017, 2018, and 2019 that have very distinct patterns. This was noted by Energy and Water Services to be due to CEC boiler shutdowns. These periods are not labelled as anomalies. These periods do not overlap the testing portion of the train/test split.
HW Main Meter Flow	• Single points or groups of data points sitting well above surrounding trends of data. • Rapid drops in data points to zero or near zero. These may be due to operations but are infrequent. • Flatline period of data. • Several rapid spikes up and down in values. These may be operational but are infrequent.
Boiler B-1 Gas Pressure	• All anomalies labelled for this sensor are rapid spikes in pressure well above the general trend.
Boiler B-1 Exhaust O_2	• This sensor appears to have constant values for long period of time and undergoes changes in values infrequently. These infrequent changes in values have been labelled. • Zero value data points.

Appendix C – Phase 1 Test Results

A list of the tests completed in Phase 1 are provided in Table C1 and the qualitative performance results are provided in Table C2. The best performing test for each sensor is highlighted in blue and figures showing results for these corresponding tests are provided in Sections C1 to C5.

TABLE C1: PHASE 1 TEST PARAMETERS

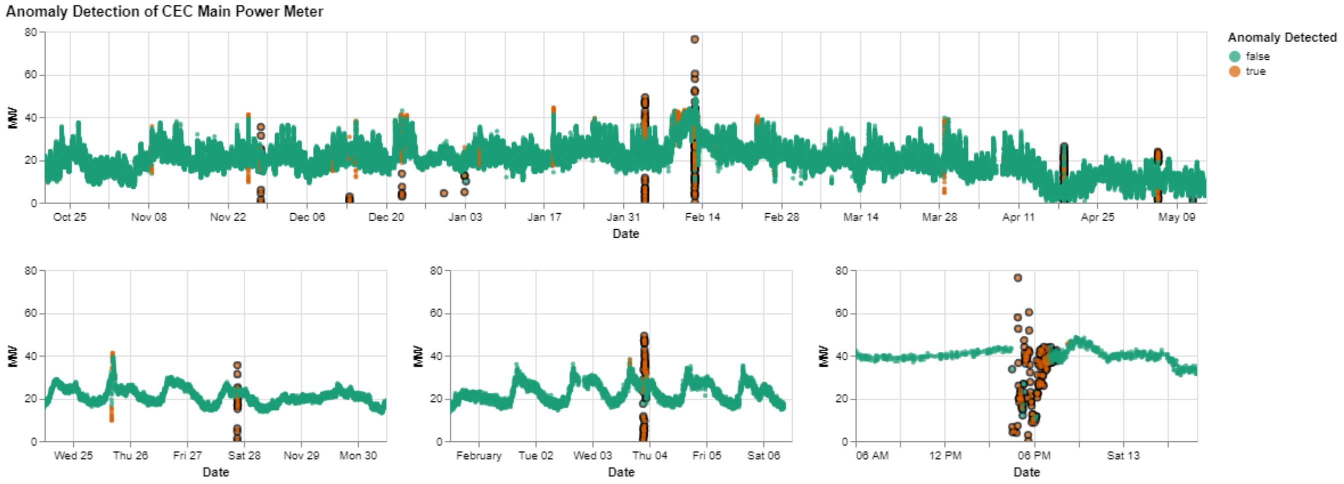
Test No.	Sensor	Type	Window Size (Data Points)	Window Size Approx. Duration	Best Window Size	Data Removed from Training	Threshold Multiplication Factor Used
1	HW Main Meter Power	Next Point Prediction	15, 30, 60, 120	30 min, 1 hr, 2 hr, 4 hr	15	No	1.6
2	HW Main Meter Power	Next Point Prediction	15, 120	30 min, 4hr	15	Yes	1.6
3	HW Main Meter Power	Sequence Reconstruction	15, 30, 60, 120	30 min, 1hr, 2 hr, 4hr	15-30	No	1.8
4	HW Main Meter Entering Water Temperature	Next Point Prediction	15	3 hr 45 min	15	No	1.4
5	HW Main Meter Entering Water Temperature	Sequence Reconstruction	15, 30, 60, 120	3 hr 45 min, 7 hr 30 min, 15 hr, 30 hr	15	No	0.3
6	HW Main Meter Flow	Next Point Prediction	15, 30, 60, 120	15 min, 30 min, 1 hr, 2 hr	15	No	2
7	HW Main Meter Flow	Next Point Prediction	15, 60	15 min, 1hr	15	Yes	2
8	HW Main Meter Flow	Sequence Reconstruction	15, 60, 120	15 min, 1 hr, 2 hr	15	No	1.72
9	Boiler B-1 Gas Pressure	Next Point Prediction	15, 30, 60, 120	3 hr 45 min, 7 hr 30 min, 15 hr, 30 hr	15	No	1.5
10	Boiler B-1 Gas Pressure	Next Point Prediction	15, 60	3 hr 45 min, 15 hr	15	Yes	2
11	Boiler B-1 Gas Pressure	Sequence Reconstruction	4, 15, 30, 60, 120	1 hr, 3 hr 45 min, 7 hr 30 min, 15 hr, 30 hr	15	No	0.23
12	Boiler B-1 Exhaust O_2	Next Point Prediction	4, 8, 48	1 hr, 2 hr, 12 hr	4	No	1.25
13	Boiler B-1 Exhaust O_2	Next Point Prediction	4, 8	1 hr, 2hr	4	Yes	1.25
14	Boiler B-1 Exhaust O_2	Sequence Reconstruction	4, 8, 15, 48	1 hr, 2 hr, 3.75 hr, 12 hr	4	No	1

TABLE C2: PHASE 1 TEST RESULTS

Test No.	Sensor	Event Identification	Event Coverage	Initial Even Detection	False Anomaly Identification
1	HW Main Meter Power	Most manual labelled events identified	Identifies most long duration events but if event contains many points close together only identifies less than half	Typically identifies first point	Moderate
2	HW Main Meter Power	Most manual labelled events identified	Identifies most long duration events but if event contains many points close together only identifies less than half	Typically identifies first point	Moderate
3	HW Main Meter Power	Most manual labelled events identified	Identifies majority of event but sometimes identifies values after event	Can be delayed by a couple points if event is less obvious	Minimal on smaller windows, moderate on larger windows
4	HW Main Meter Entering Water Temperature	Most manual labelled events identified	Identifies initial anomalous jumps, but very little of entire event	Typically identifies first point	Minimal
5	HW Main Meter Entering Water Temperature	Most manual labelled events identified	Identifies majority of event but sometimes identifies values after event	Can be delayed by a couple points if event is less obvious	Minimal on smaller windows, moderate on larger windows
6	HW Main Meter Flow	Most manual labelled events identified	Identifies most long duration events but if event contains many points close together only identifies less than half	Typically identifies first point	Minimal
7	HW Main Meter Flow	Most manual labelled events identified	Identifies most long duration events but if event contains many points close together only identifies less than half	Typically identifies first point	Minimal
8	HW Main Meter Flow	Most manual labelled events identified	Identifies majority of event but sometimes identifies values after event	Can be delayed by a couple points if event is less obvious	Minimal on smaller windows, moderate on larger windows
9	Boiler B-1 Gas Pressure	Most manual labelled events identified	Does not identify majority of event.	Typically identifies first point	Moderate
10	Boiler B-1 Gas Pressure	Most manual labelled events identified	Identifies most high outlier events. Does not identify events close to normal data	Typically identifies first point	Minimal
11	Boiler B-1 Gas Pressure	Most manual labelled events identified	Identifies majority of event but sometimes identifies values after event	Can be delayed by a couple points if event is less obvious	Minimal
12	Boiler B-1 Exhaust O_2	Most manual labelled events identified	Identifies most long duration events but if event contains many points close together only identifies less than half	Typically identifies first point	Minimal
13	Boiler B-1 Exhaust O_2	Some manual labelled data identified	Identifies very little of event	Typically identifies first point	Moderate
14	Boiler B-1 Exhaust O_2	Most manual labelled events identified	Identifies majority of event but sometimes identifies values after event	Can be delayed by a couple points if event is less obvious	Minimal on smaller windows, moderate on larger windows

C1 – HW Main Meter Power

All manual events identified. Several additional potentially anomalous events identified. Majority of points in an event are identified.

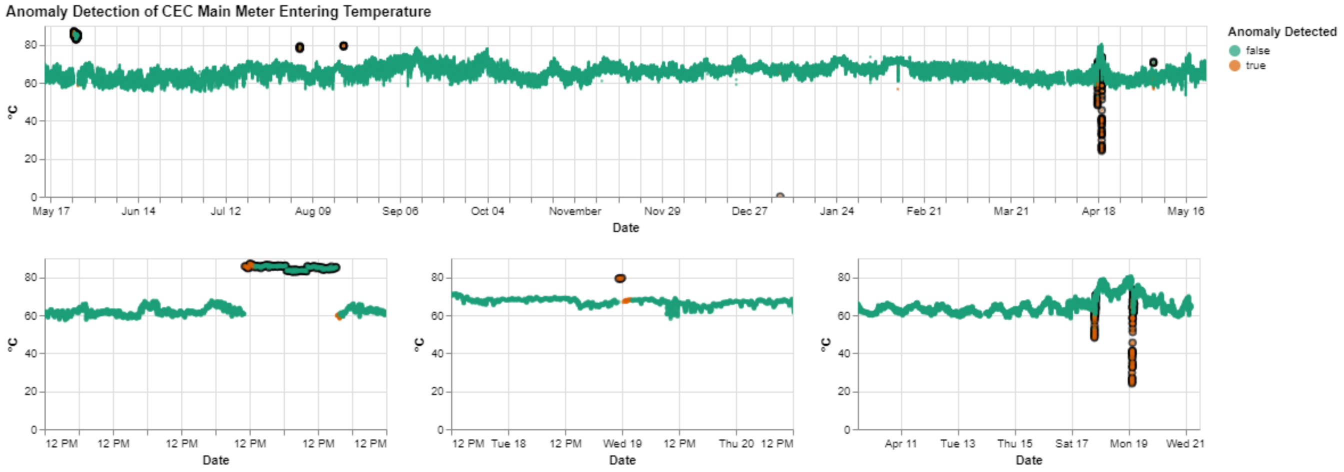


Manual Labels Outlined in Black

FIGURE C1: HW MAIN METER POWER MODEL RESULTS

C2 – HW Main Meter Entering Water Temperature

All manual events identified. The first event (far left on upper graph) only had the first few points in the event identified as anomalous (as shown in the lower left graph). Other events had good coverage.

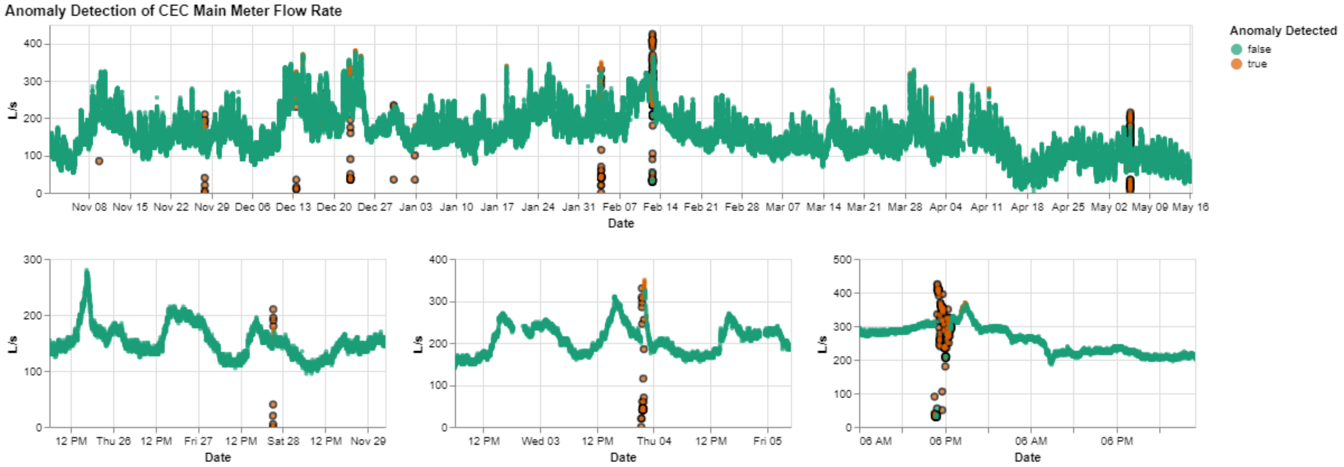


Manual Labels Outlined in Black

FIGURE C2: HW MAIN METER ENTERING WATER TEMPERATURE MODEL RESULTS

C3 – HW Main Meter Flow

All manual events identified. Majority of points in an event are identified.

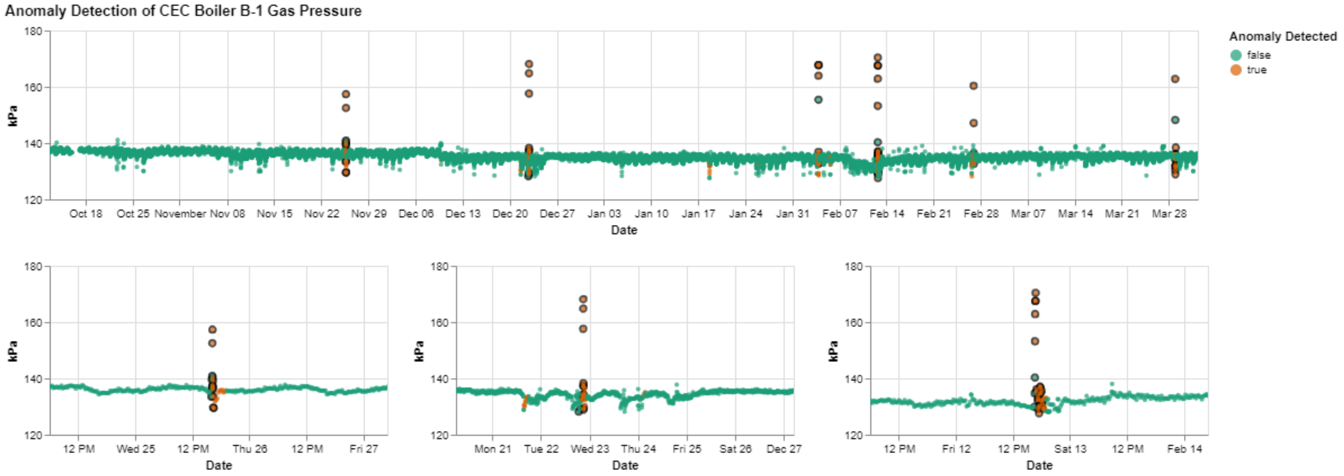


Manual Labels Outlined in Black

FIGURE C3: HW MAIN METER FLOW MODEL RESULTS

C4 – Boiler B-1 Gas Pressure

All manual events identified. Several normal points after the event are sometimes labelled as anomalous.

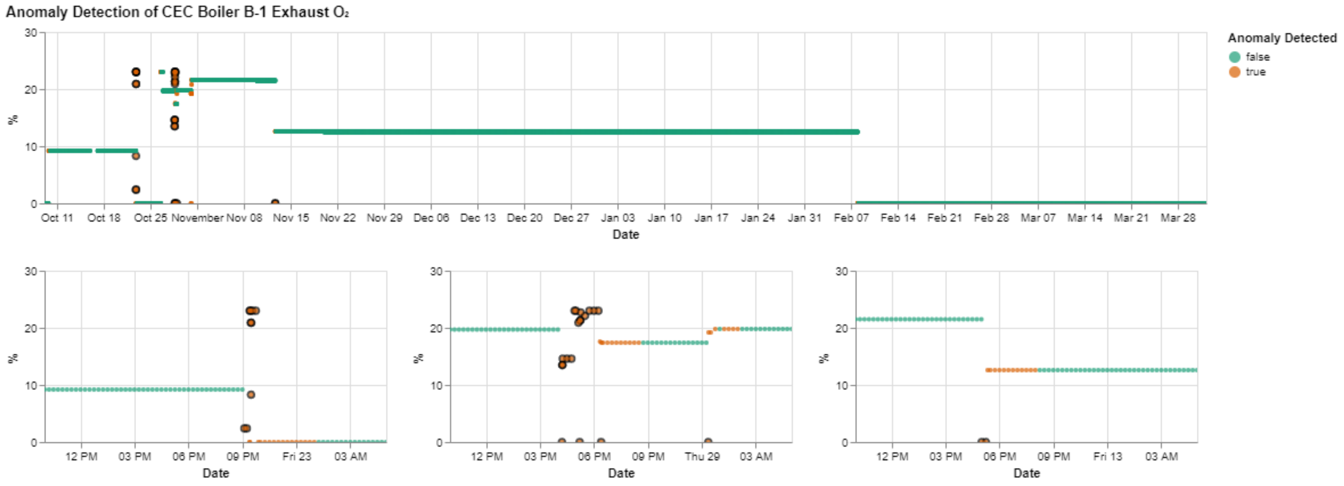


Manual Labels Outlined in Black

FIGURE C4: BOILER B-1 GAS PRESSURE MODEL RESULTS

C5 – Boiler B-1 Exhaust O_2

Anomalies were manually labelled at any jumps between the flatline values and all jumps were identified. Several normal points after the event are sometimes labelled as anomalous. Note that the last jump shown in the upper graph was identified but the label is not visible on the graph.



Manual Labels Outlined in Black

FIGURE C5: BOILER B-1 EXHAUST O_2 MODEL RESULTS

Appendix D – Phase 2 Test Results

The qualitative performance results for Phase 2 testing are provided in Table D1. Figures showing test results are provided in Sections D1 to D5.

TABLE D1: PHASE 2 TEST PARAMETERS

Test No.	Sensor	Type	Window Size (Data Points)	Window Size Approx. Duration	Data Removed from Training	Threshold Multiplication Factor Used
1	Boiler B-2 Exhaust O_2	Sequence Reconstruction	15	1 hr	No	0.9
2	Feeder 60L56 Real Power	Sequence Reconstruction	15	15 min	No	0.4
3	Boiler B-2 Gas Pressure	Sequence Reconstruction	15	3 hr 45 min	No	0.2
4	Boiler B-1 Power	Sequence Reconstruction	15	30 min	No	1
5	Process System Pump Speed	Sequence Reconstruction	15	3 hr 45 min	No	1.75

TABLE D2: PHASE 2 TEST RESULTS

Test No.	Sensor	Event Identification	Event Coverage	Initial Even Detection	False Anomaly Identification
1	Boiler B-2 Exhaust O_2	Misses flatline identification, but identifies other events	Identifies majority of event but sometimes identifies values after event.	Mostly identified immediately	Minimal
2	Feeder 60L56 Real Power	Most events identified	Identifies majority of event but sometimes identifies values after event. Doesn't identify entire flatline event, just the start of the event.	Mostly identified immediately	Minimal
3	Boiler B-2 Gas Pressure	Most events identified	Identifies majority of event but sometimes identifies values after event.	Mostly identified immediately	Minimal
4	Boiler B-1 Power	Most events identified	Identifies majority of event but sometimes identifies values after event. Doesn't identify entire flatline event, just the start of the event.	Mostly identified immediately	Minimal
5	Process System Pump Speed	Most events identified	Identifies majority of event but sometimes identifies values after event.	Mostly identified immediately	Minimal

D1 – Boiler B-2 Exhaust O_2

The model was able to easily identify single point anomalies but unable to identify flatline value anomalies. Review of the sensor indicates that there is minimal pattern within the data and a large amount of noise. Inspection of the LMST-ED shows high reconstruction error for this noisy data and the flatline periods have better reconstruction. A more sophisticated threshold method may be able to provide better identification for this type of data (for example assessing the distribution of error within a window of data and flagging data outside of expected distribution bounds, instead of using a maximum error threshold).

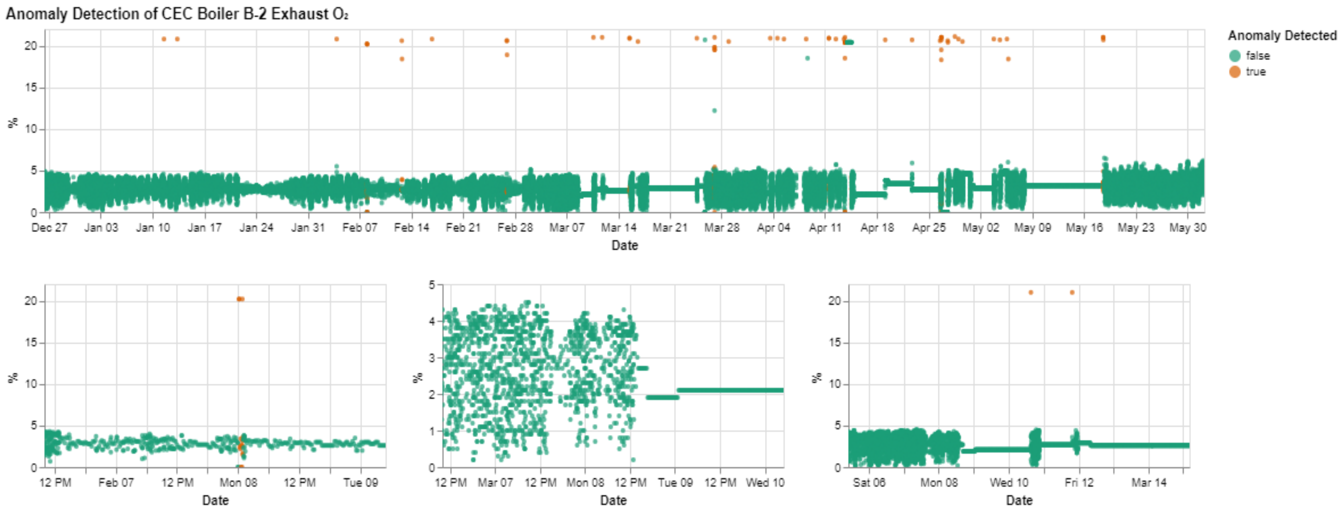


FIGURE D1: BOILER B-2 EXHAUST O_2 MODEL RESULTS

D2 – Feeder 60L56 Real Power

Appears to provide good performance and identifies majority of anomalous looking data. However, only identifies start of flatline zero value anomalies.

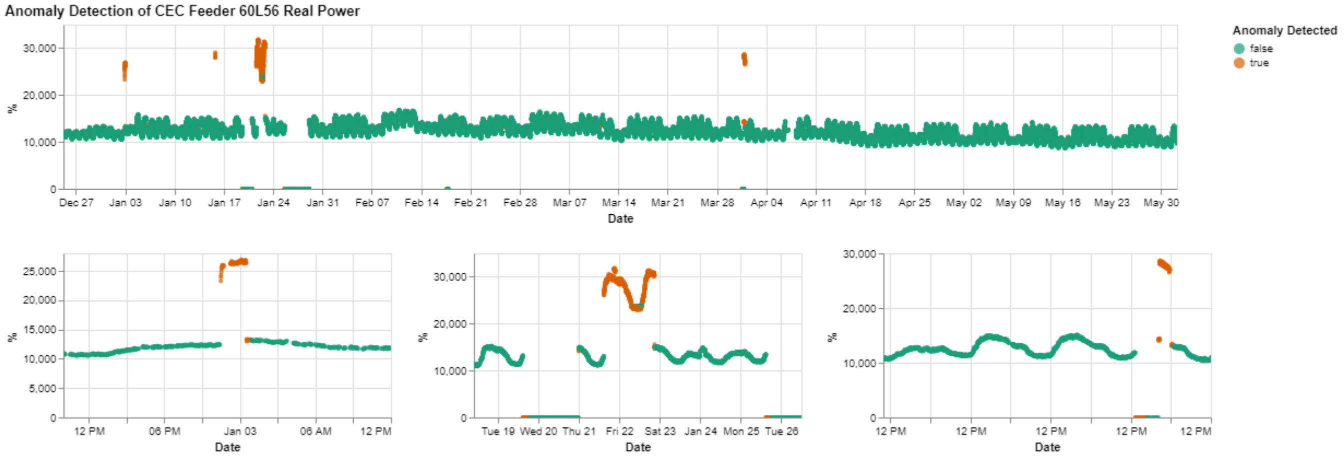


FIGURE D2: FEEDER 60L56 MODEL RESULTS

D3 – Boiler B-2 Gas Pressure

Appears to provide good performance and identifies majority of anomalous looking data.

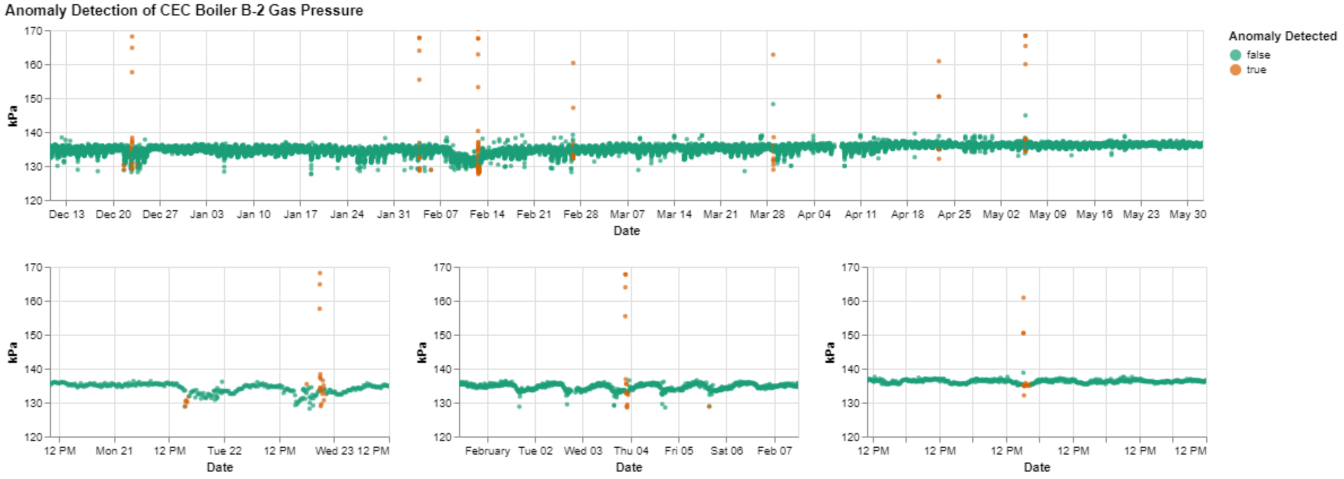


FIGURE D3: BOILER B-2 GAS PRESSURE MODEL RESULTS

D4 – Boiler B-1 Power

Appears to provide good performance and identifies majority of anomalous looking data. However, only identifies start of flatline zero value anomalies.

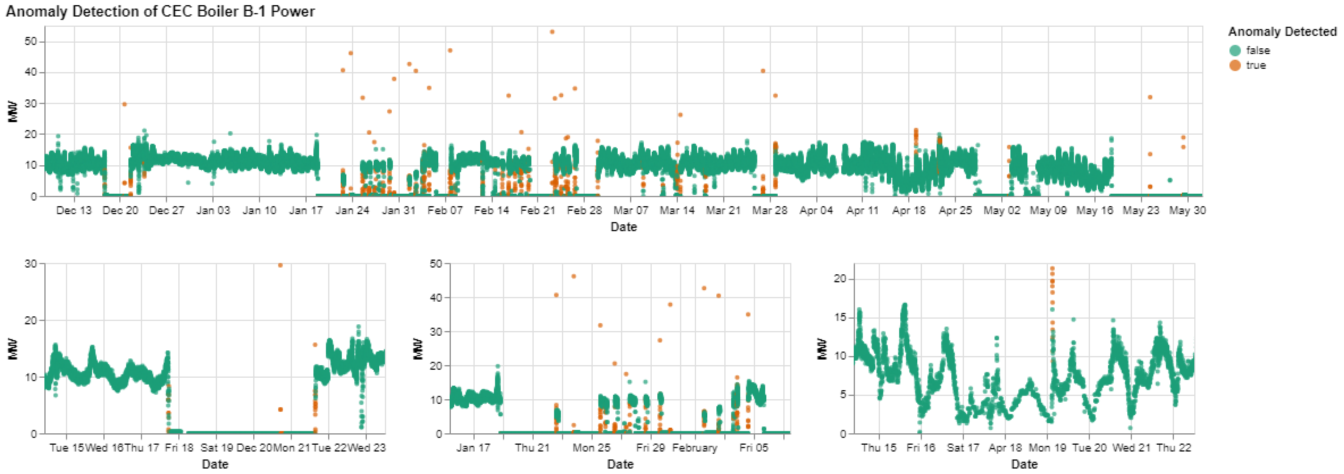


Figure D4: Boiler B-1 Power Model Results

D5 – Process System Pump Speed

Appears to provide good performance and identifies majority of anomalous looking data.

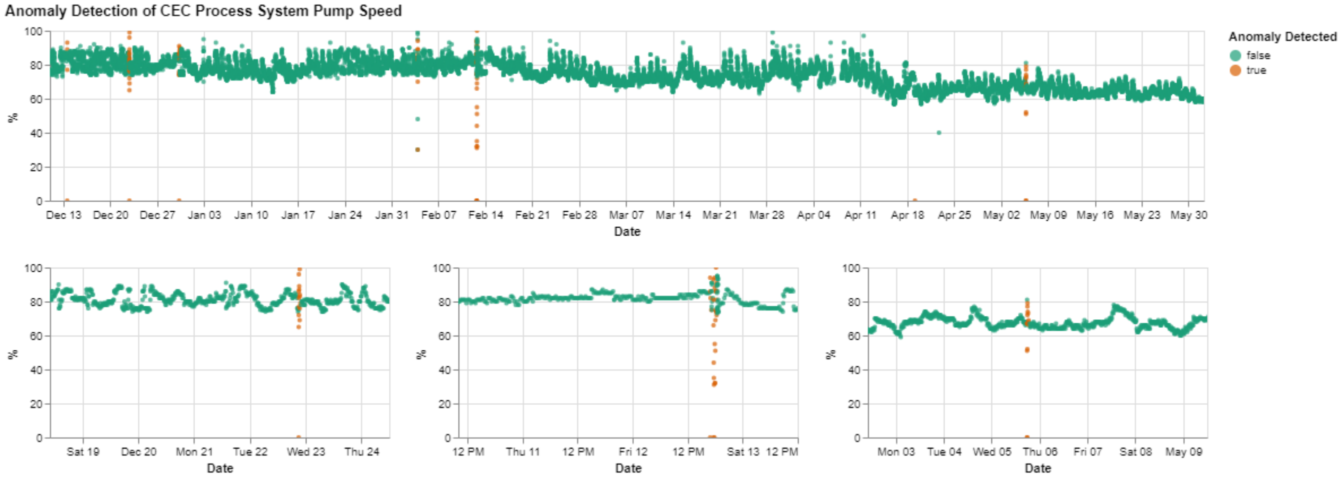


Figure D5: Process System Pump Speed Model Results