CMPT 733 Final Project, Spring 2020
Archana Subramanian, Asha Tambulker, Carmen Riddel, Shreya Shetty
Professional Master's in Computing Science Big Data- Simon Fraser University

Elevator Movement Anomaly Detection: Building a System that Works on Many Levels

Motivation and Background

Elevators are used everyday in residential, commercial and industrial settings. Elevator incidents have been on the rise in Canada for many years and it would be useful to be able to predict when elevators may require maintenance (Perkel, 2017). One way to accomplish this would be to detect anomalous acceleration patterns in elevators both in a vertical sense (up and down movement of the elevator), as well as in a horizontal sense (side-to-side). Anomalous movement along the vertical axis may represent issues such as abrupt stops, uncomfortable overspeeding, and leveling discrepancies, where the elevator surface does not properly align with the floor after stopping (Acada Consulting Inc., 2018). These issues may be caused mechanically by rope or sheave issues, or unbalanced counterweight (Lorsbach, n.d.). Anomalous horizontal movement could represent excessive side-to-side vibration, which could be caused by misaligned rails or roller/slide guides. An anomaly detection system could potentially save businesses money on elevator repairs and replacement, and more importantly, could prevent injuries and even fatalities.

A 2018 report by Acada Consulting Inc. in partner with Technical Safety BC (hereafter collectively referred to as TSBC) acted as a helpful guide for the duration of the project (Acada Consulting Inc., 2018). As the proprietors of the dataset, the authors were able to detect elevator movement anomalies using a number of different methods including Hidden Markov Models, Self-organizing Maps and wavelet transformation.

An additional study used crowdsourced acceleration data in an attempt to recognize elevator movement activity (Yang et al., 2013). The goal was to have a robust, generalized model that could detect movement in several different types of elevators.

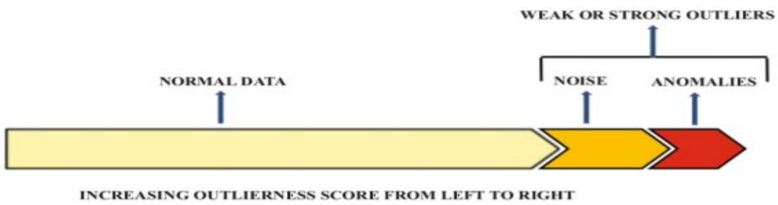
Anomaly detection in other types of time series using deep learning is a very popular topic. One study was successful in detecting anomalies in various types of sensor data including parameters from electrocardiograms, space shuttles and vehicle engines (Mahltora et al., 2016). The authors implemented a Long Short Term Memory (LSTM) Encoder/Decoder to detect anomalies based on prediction error.

Problem Statement and Challenges

The questions we set out to answer were:

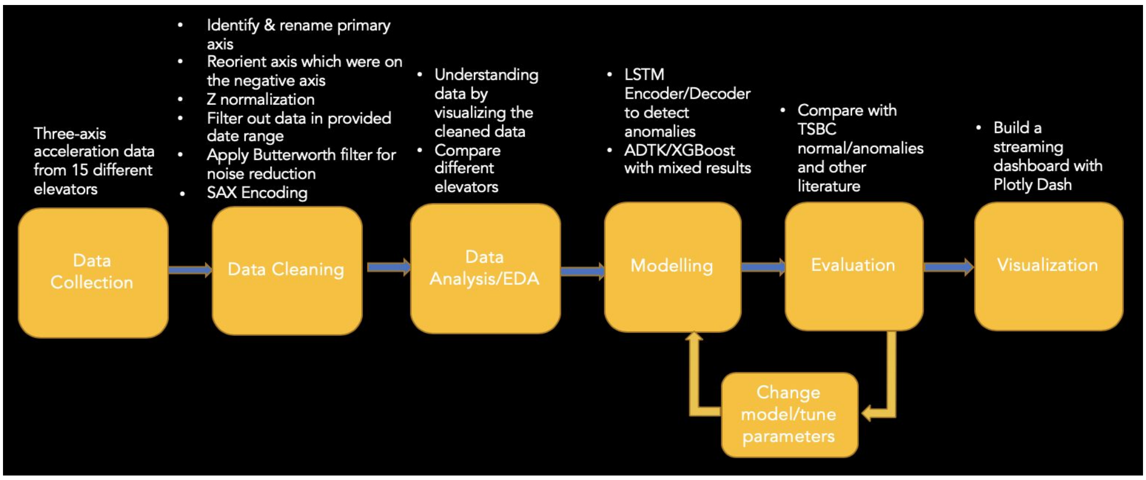
1. How can we use internet-of-things (IOT) and machine learning to predict when elevators may require maintenance?
2. How can we present these findings in near real-time?

These questions were quite challenging to answer because we were presented with unlabelled data to work with and we were not able to formally evaluate our findings. As well, sensor data requires significant cleaning and processing before it is usable. Removing noise from sensor data is finicky because noise must be controlled, but with strong denoising, you run the risk of removing natural variation in the data. This variation is important when looking for outlying points and patterns (Acada Consulting Inc., 2018). The figure below illustrates noise and outliers also termed as weak and strong outliers, respectively (Hayati, 2018).



Spectrum of Data, from Normal to Anomalous from (Hayati, 2018)

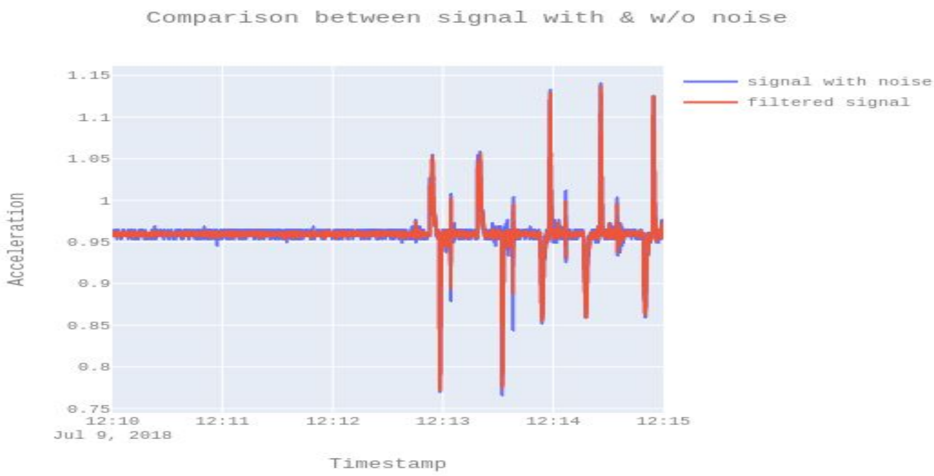
Data Science Pipeline



Data Science Pipeline

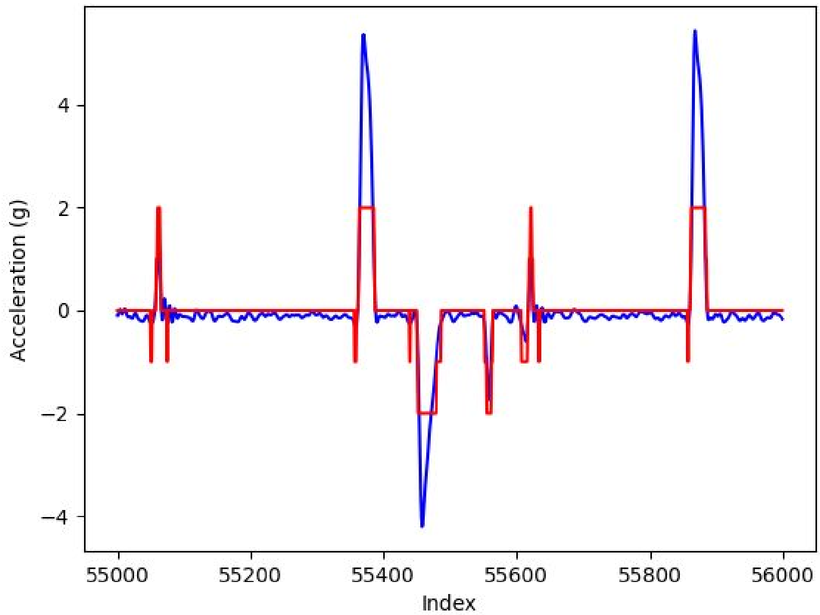
Below is a description for each component involved in our Data Science pipeline:

1. **Data collection:** We received approximately 75 GB of data from TSBC consisting of data from 15 different elevators. Unfortunately due to file corruption, only 12 of the elevators were explored as part of this project. The data consisted of 4 columns: timestamp, x-axis acceleration, y-axis acceleration and z-axis acceleration. The sampling frequency was 25 Hz which means there were 25 data points per second.
2. **Data cleaning:** This was one of the important parts of the project. Below are the list of data preprocessing done:
    - Adding elevator ID as one of the columns and renaming the primary axis (vertical, up/down movement) and secondary axes. The data related to the primary axis was provided by TSBC. The primary axis was different among the elevators and it depended how the accelerometer was placed in each instance.
    - Reoriented the values on the negative primary axis to positive. This was done when the accelerometer was placed in such a way that the primary axis was on a negative axis. This step helped us to keep all values in positive range for analysis.
    - Filtered out the data points within the date range provided by TSBC, which was one month of data: 2018-07-01 12:00:00 to 2018-08-01 12:00:00.
    - Applied Z normalization which centred the mean at 0 and set the standard deviation to 1, enabling comparison of patterns and values between the elevators.
    - Applied a Butterworth filter, a low pass filter available for signal processing which reduces noise and creates a smoother time-series graph. A low pass filter passes signals with a frequency lower than a selected frequency and attenuates signals with frequencies higher than the cutoff frequency (Wikipedia).



Noise Reduction from Low Pass Butterworth Filter

- Applied symbolic aggregate approximation (SAX) encoding which binned each data point and assigned a symmetric integer based on its value. For example, in a 3-alphabet encoding, values below -0.4307273 were set to -1, those above 0.4307273 were set to 1, and those in between were set to 0. The 0 values were classified as 'stationary', i.e., indicated when the elevator was not moving, and the 1 and -1 values were classified as 'moving'. The intent was for the model to learn the pattern of the data rather than the magnitude of acceleration; both 3 and 5-alphabet SAX encoding were explored.



SAX Encoding with an Alphabet of Size 5

3. **Exploratory Data Analysis:** This was done to compare the data patterns of the elevators. We compared the maximum positive and negative acceleration from each elevator and the percentage of time each elevator spent in movement.
4. **Modelling:** In this step, we attempted to detect anomalies using machine learning models including Long Short Term Memory (LSTM), Random Forest and Extreme Gradient Boosting Algorithm (XGBoost) for the primary axis and Generalized ESD test for vibration analysis across the horizontal axes.
5. **Evaluation:** This is one of the important steps in the data science model. Results were compared visually with anomalous and normal data found by TSBC (Acada Consulting Inc., 2018).
6. **Visualization:** Since we are dealing with sensor data, visualization is critical to understand the result. We plotted varieties of anomalies and built an elevator anomaly streaming system.

Methodology

Data Cleaning and Exploration

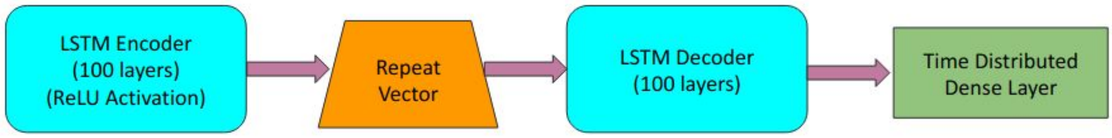
We used PySpark for initial data cleaning as it allows for parallel computing on large datasets. This helped us to obtain the correct window of data, reorient, filter and normalize while taking advantage of the columnar structure of data frames. We also used common packages such as Pandas, Numpy and Scikit-learn, as well as plotting libraries such as Matplotlib, Seaborn and Plotly Graph Objects, throughout the project.

The tools and methods used for modeling are explained more in depth.

LSTM

LSTM Encoder/Decoder frameworks are useful for time series anomaly detection. When trained on normal data, they are able to learn and reconstruct normal sequences, but struggle to predict anomalous ones, resulting in a higher prediction error (Mahltora et al., 2016). This prediction error can then be subjected to a threshold, with those points above the threshold considered to be anomalies. We used a LSTM Encoder/Decoder model on the primary (vertical) axis as anomalies in elevator trips were the main focus of this project.

We used Keras with Tensorflow to implement a LSTM Encoder/Decoder model as it is a very easy package to use and build sequential models with. The Keras/Tensorflow combination also allowed for easy GPU use for accelerated computing.



LSTM Encoder/Decoder Architecture

Our model consisted of a LSTM encoder, followed by a repeat vector which repeats the output of the encoder so it is the same dimension as its input. This was then fed into the LSTM decoder. The last component applied a fully connected layer to each temporal slice. Our input sequences consisted of 5 time steps. We also used an Adam Optimizer and mean absolute error (MAE) to determine anomalies.

Only 'moving' data points were used as input to prevent the model from learning 'stationary' points as normal and 'moving' ones as anomalous. Normally, training data for a LSTM encoder/decoder would consist entirely of normal points. That way, the model would know normal data very well and would have a hard time predicting anomalous points, resulting in a higher error. As we did not have access to labelled data, we used all 'moving' data as input.

Random Forest

We used Scikit-learn's Random Forest Regressor as a naive algorithm to identify the anomalies in our data. Random forest is one of the best algorithms that uses a bagging technique (an approach by which computes the end result by aggregating the results from subsets of the data) to produce results with high accuracy (Yiu, 2019).

XGBoost Regressor

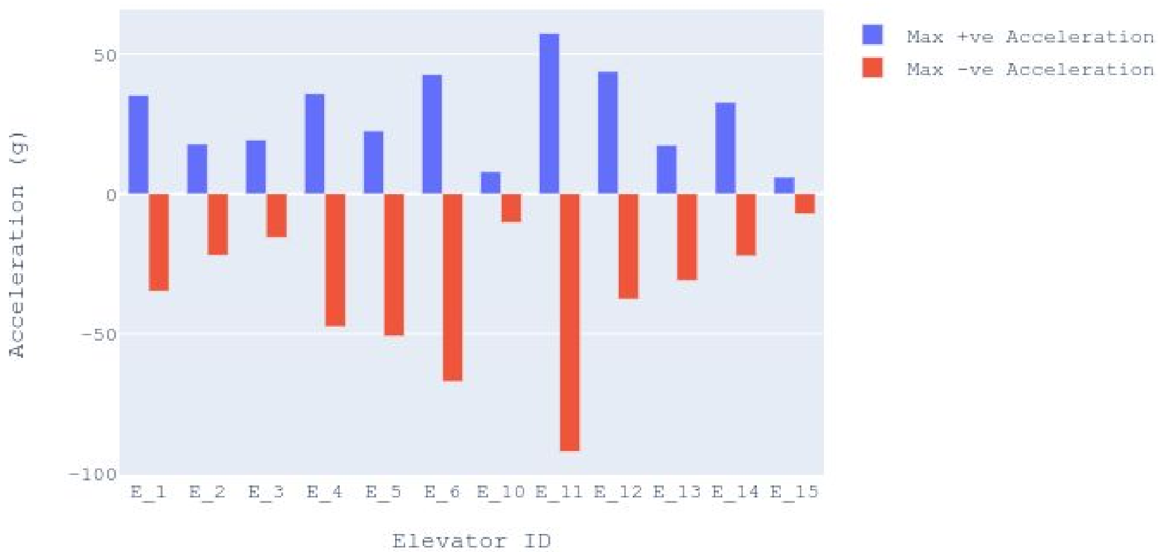
The XGBoost algorithm is an ensemble learning model which also follows a tree based approach to produce results. Unlike the Random forest algorithm that uses a bagging technique, XGBoost uses boosting/gradient boosting where the algorithm produces the prediction results based on an ensemble of weak predictors - most likely decision trees. Boosting techniques significantly increases the overall accuracy of prediction by re-modelling weak learners (Brownlee, 2016). Another major advantage of XGBoost as opposed to the general gradient boosted algorithms is that it uses an advanced regularization term which makes the model highly generalized. It was also researched that an XGBoost model produced higher accuracy for time series forecasting as compared to other machine learning models (Nikulski, 2020). We used the preprocessed data - normalized, low pass filtered, SAX encoded data as the input to our XGBoost model.

Generalized ESD Test for Horizontal Movement

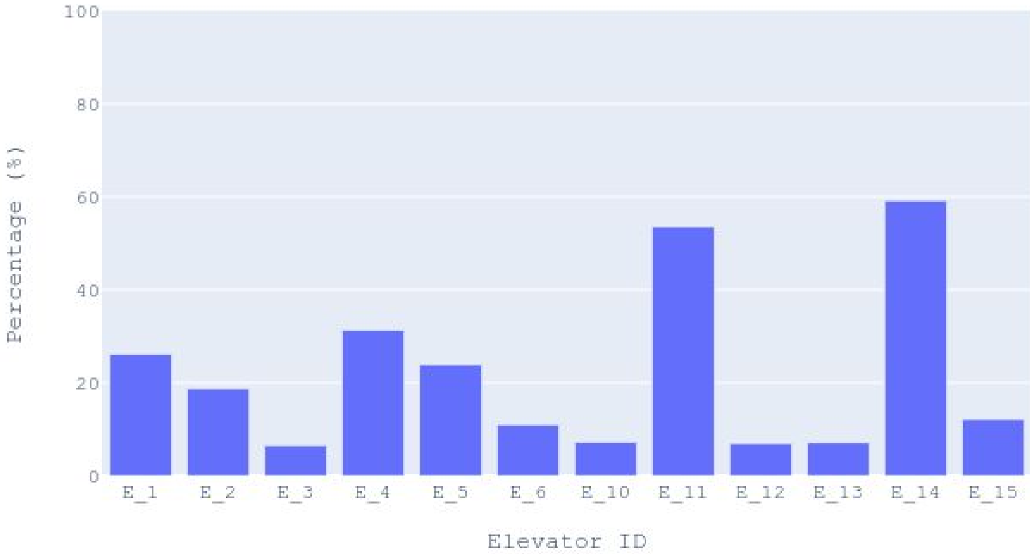
For the horizontal axes, representing side-to-side vibration of the elevator, we used the Generalized ESD test detector from Arundo's Anomaly Detection Toolkit (ADTK) package. A Generalized ESD test is for univariate data, and follows the assumption that the normal data follows a roughly normal distribution (Kuppusamy and Kaliyaperumal, 2013). We believed this to be a reasonable assumption for each horizontal axis as the ideal amount of vibration is relatively close to zero, with points of higher magnitude in either direction more likely to be outliers. Arundo's ADTK is an easy to use, high level package with built in plotting functionality.

Exploratory Data Analysis Results

The plots below show the results of our exploratory data analysis. We were not able to confirm, but speculated that elevators such as 3, 10 and 15 were those of smaller residences and/or businesses as they did not reach particularly high accelerations and were not used very frequently. Elevator 11, a fast and busy elevator, was likely that of a high rise condo or bustling place of business.



Maximum Positive and Negative Acceleration for Each Elevator



Percentage of Time each Elevator spent in Movement

Evaluation

LSTM

As we were working with unlabelled data, we relied on visual differentiation between normal and anomalous elevator acceleration for the vertical primary axis. This is not as difficult as it may sound, as elevators follow fairly universal patterns and we were able to confirm with multiple sources what those patterns look like. Any sequences differing from those were likely anomalous and possibly indicative of an elevator needing repair.

Certain methods we attempted were not visibly successful at detecting anomalies. This was clear as in these cases, essentially all points higher than an absolute, near zero threshold were classified as anomalies. This is obviously incorrect as this would indicate that all movement of the elevator was anomalous; an elevator's sole purpose is to move, afterall.

The LSTM Encoder/Decoder shared this issue at first. Once we implemented SAX encoding and excluded the near zero values from training, the model was able to detect the patterns in the data and some visually obvious anomalies were correctly classified as such. There did not seem to be much difference between 3 and 5-alphabet SAX encoding.

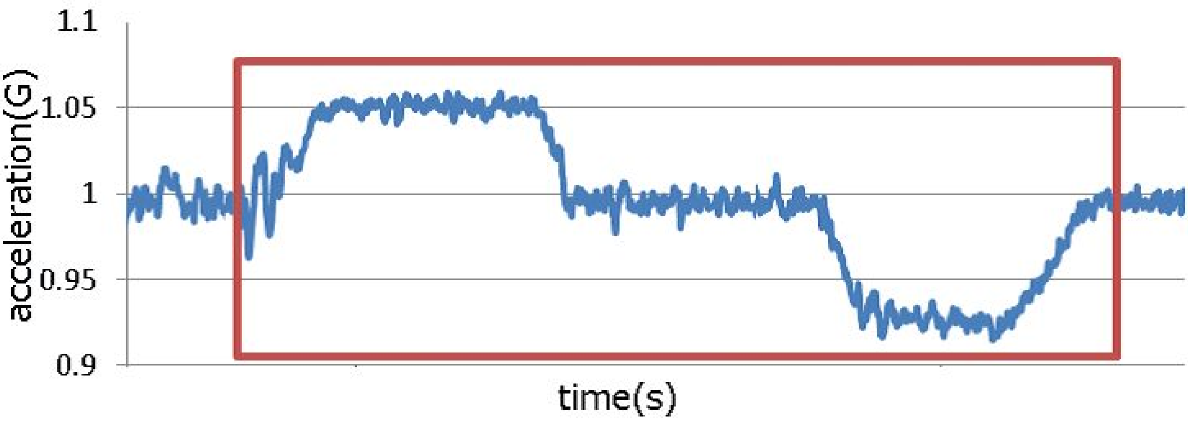
We could have visually identified anomalies and manually labelled a subset of the data to train with, but there was a trade-off between training on a large, unlabelled dataset versus a small, but labelled dataset. It would not have been feasible to label all of the data.

The model likely would have performed much better if it were trained only on normal data (if we had access to labelled anomalies). Nonetheless, the model seemed to be able to identify anomalies similar in appearance to those detected by TSBC.

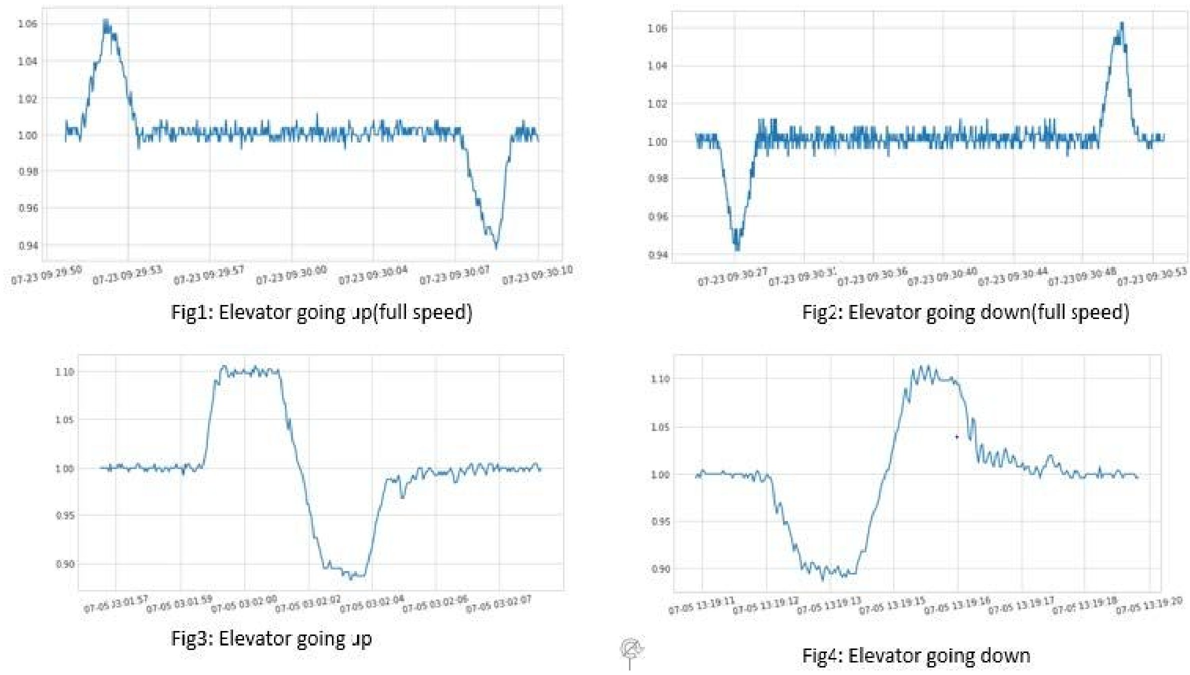
Below are some diagrams of normal elevator trips. The magnitude, shape and length of time vary by elevator, but generally you will see the following:

1. A period of constant positive or negative acceleration
2. The first phase may be followed by a period of flat acceleration, depending on the speed of the elevator
3. The trip will end with a period of acceleration opposite to the first phase before returning to flat acceleration

A trip up will have positive acceleration to start, while a trip down will have negative acceleration to start. Any movement that does not follow this pattern should be considered anomalous.

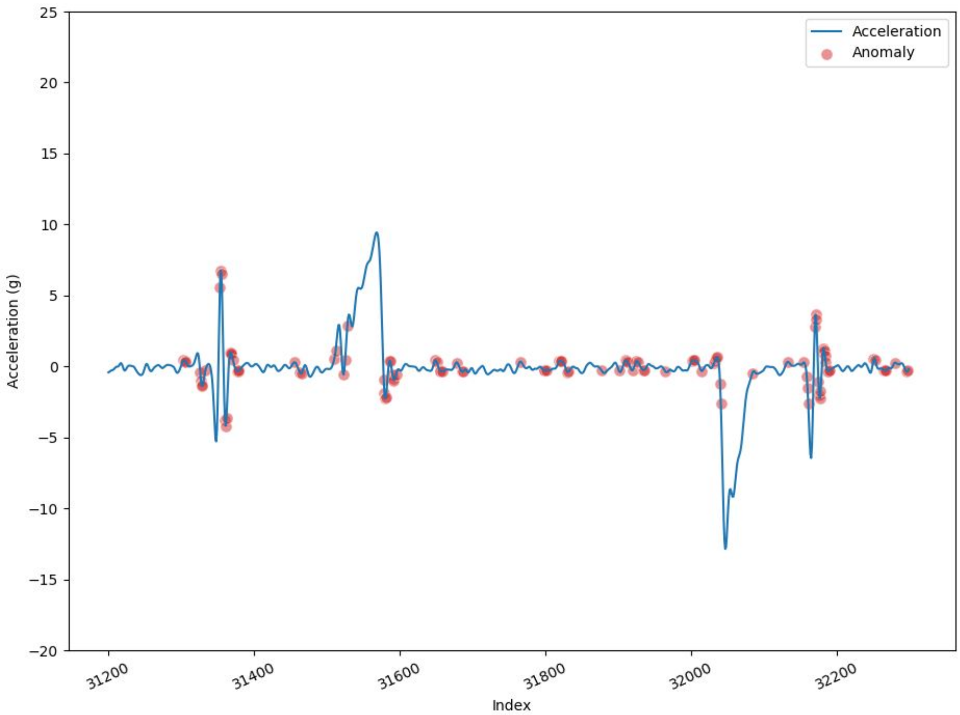


Normal Elevator Trip Up, from Yang et al., 2013

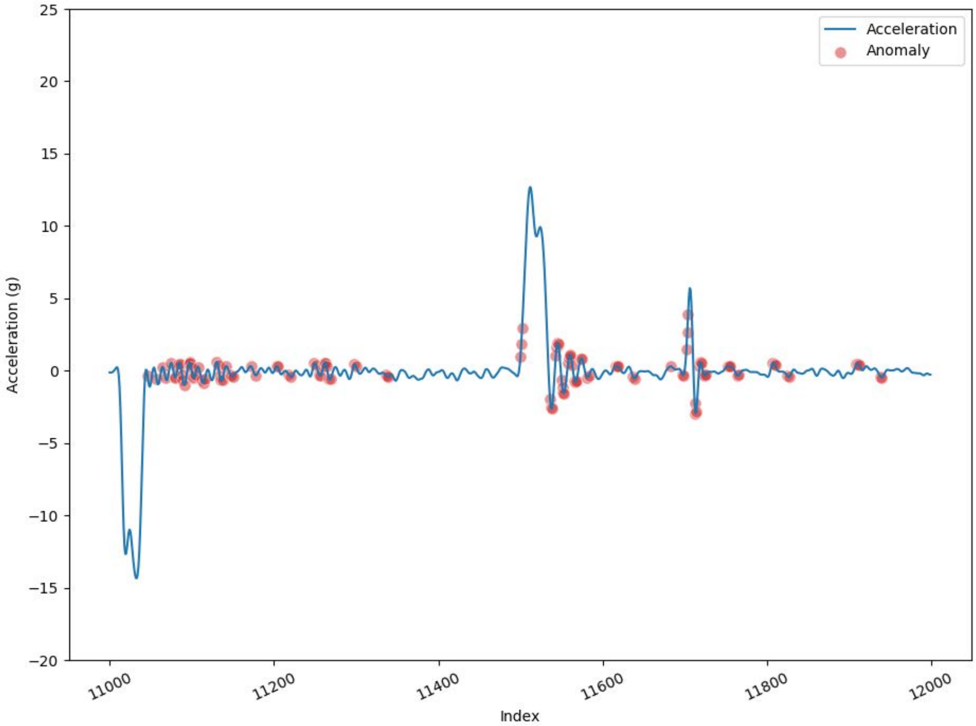


Normal Trips Detected by TSBC, from Acada Consulting Inc., 2018

The following plots show trips we were able to identify in the data, along with some anomalous points classified by our LSTM Encoder/Decoder model.

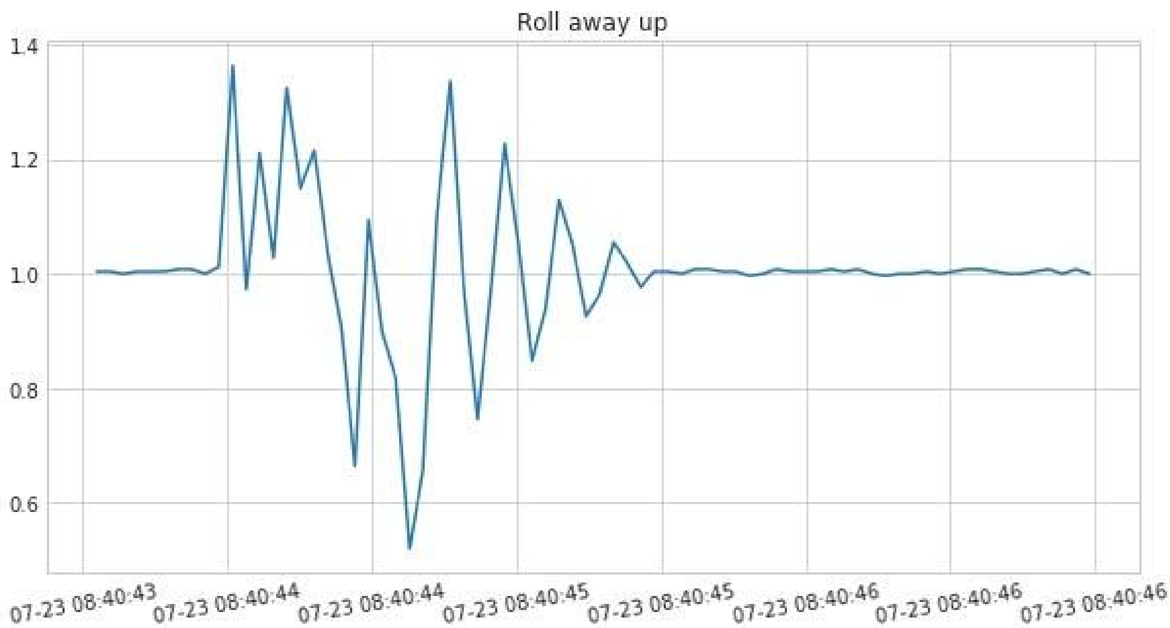


Trip Up among Anomalous Points Detected by our LSTM Model, with some possible Swamping

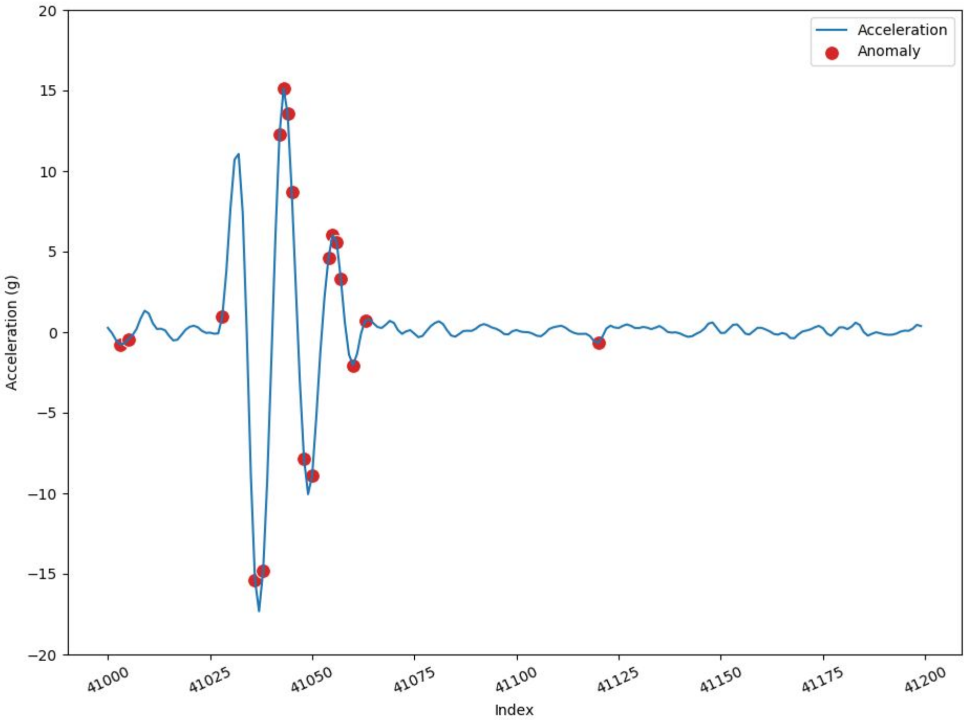


Trip Down among Anomalous Points Detected by our LSTM Model

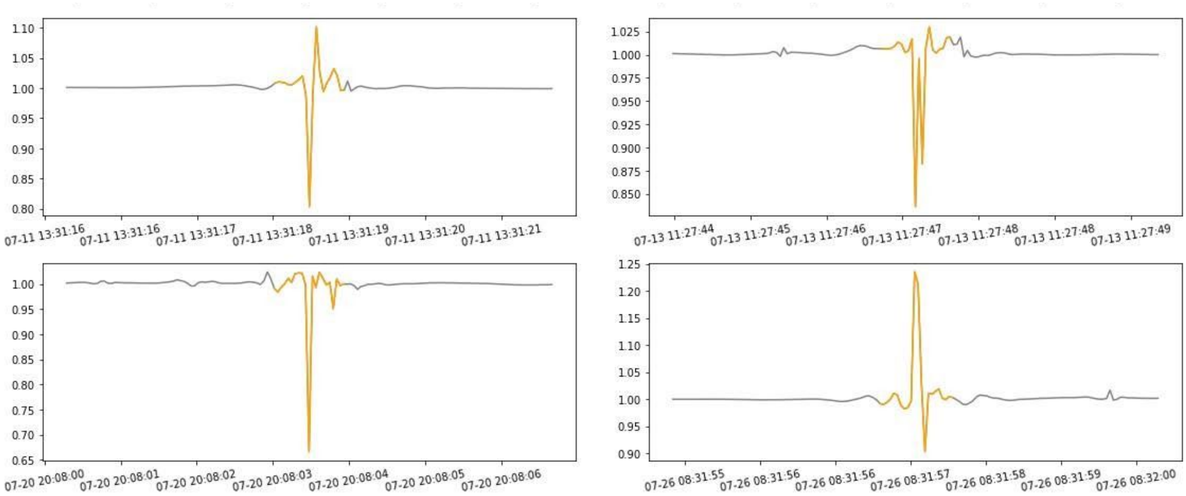
Our model also detected anomalies which appeared to be independent of trips, and many of them followed similar patterns to those found by TSBC.



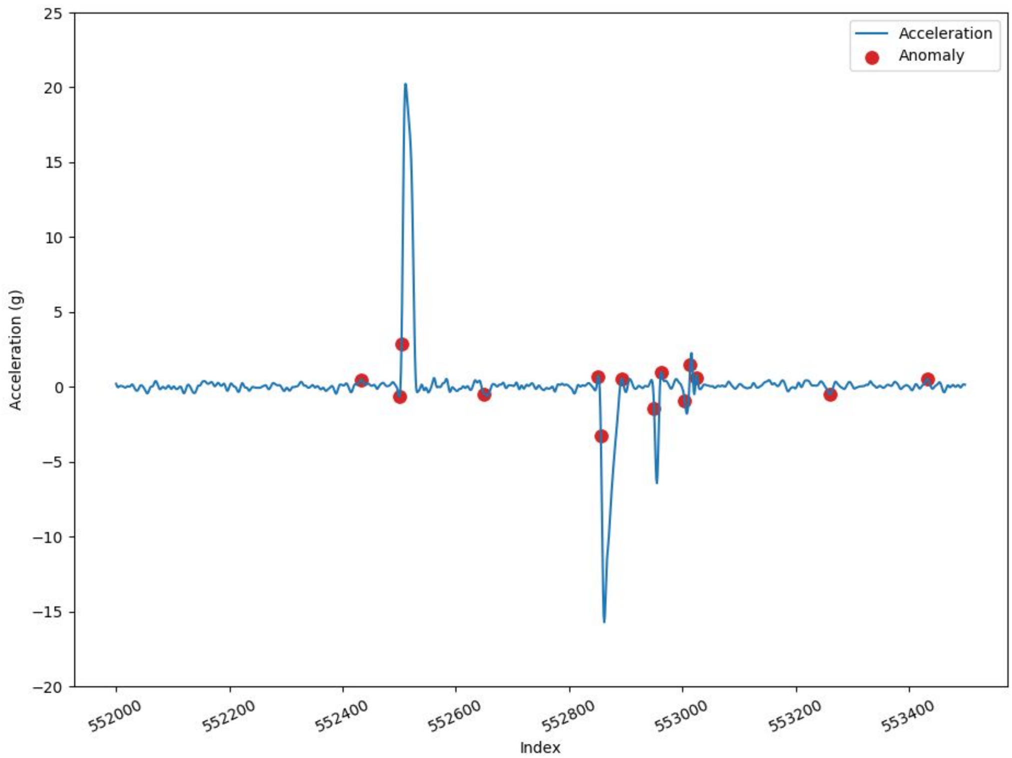
An Anomalous Pattern Detected by TSBC from Acada Consulting Inc., 2018



A Similar Erratic Pattern Deemed Anomalous by our LSTM Model



Sharply Shaped Anomalies Detected by TSBC, from Acada Consulting Inc., 2018



Similar Sharp Increases in Acceleration Deemed Anomalous by our LSTM Model, with some possible Masking

Our LSTM Model seemed to pick up on erratic patterns in the data, that is, when there was a sharp increase in acceleration, followed by another in the other direction, and so on. This is no surprise when we consider the role that SAX encoding plays. During training, when using a 3-alphabet SAX encoding, the most common patterns encountered would be those with a series of positive ones followed by a series of negative ones, or vice versa (remembering that all zero points were removed from the dataset prior to training). Consider the following sequence:

[1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-,1-,1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1-1,-1,-1]

It would be less often that the following pattern was encountered, and thus, such sequences of points would likely be considered anomalous:

[1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1]

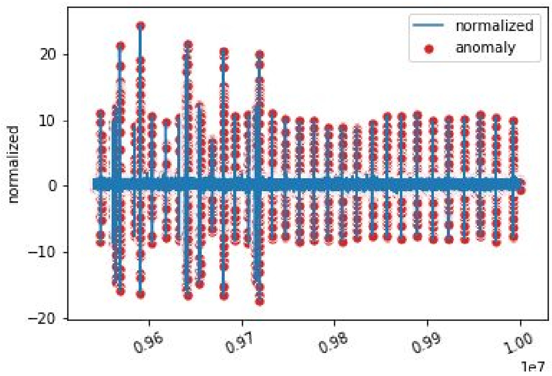
Although there was no way to confirm, the LSTM model did appear to succumb to both masking, the failure to detect certain anomalies, and swamping, classifying normal points as anomalous (Liu et al., 2008).

Random Forest

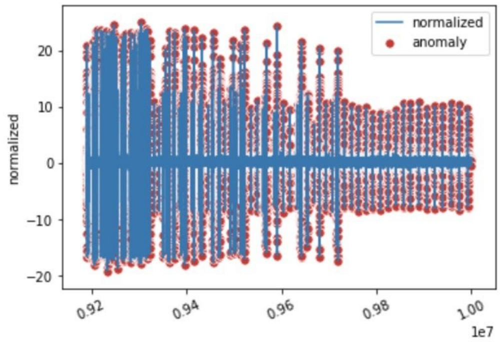
Inspite of its robustness and ease of training, the results obtained from the random forest algorithm were far from what was expected in our use case. The input to the model was the pre-processed elevator data. The value to be predicted was a part of the acceleration column which was compared with a random threshold (by trial and error) for classifying as an anomaly or not. The model's results, when plotted, showed all peak points identified as anomalies. These points had variances off of zero and the results were quite insignificant.

XGBoost

Though the research and the features of the algorithm are ideal, this did not work for our data. Like the random forest model, this model also predicted every peak (variances off of zero points) as anomalous data and did not provide the output as expected for our use case. Fine Tuning the model and modifying its hyper parameters may have provided more convincing predictions which can be considered as a part of our future work.



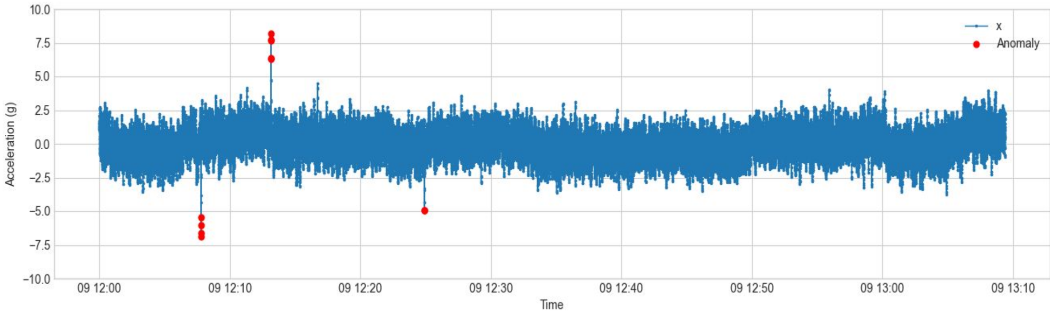
Random Forest Model

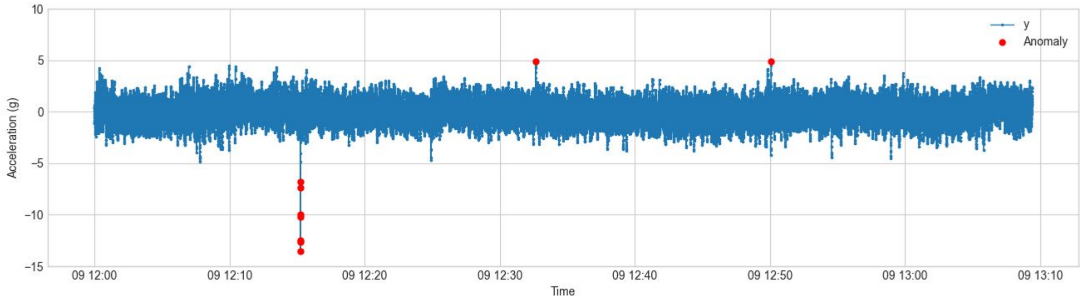


XGBoost Model

Generalized ESD Test for Horizontal Movement

After implementing a Generalized ESD test, anomalies associated with the horizontal axes, representing vibration, were detected based on their magnitude. Two plots are provided as examples, one for the x-axis and one for the y-axis, in a situation where the z-axis was the primary axis. These anomalies were even more difficult to verify than those of the vertical axis, as acceptable values for horizontal vibration depends on a number of characteristics of a particular elevator such as its type and age (Lorsbach, n.d.). This is not information we had access to.



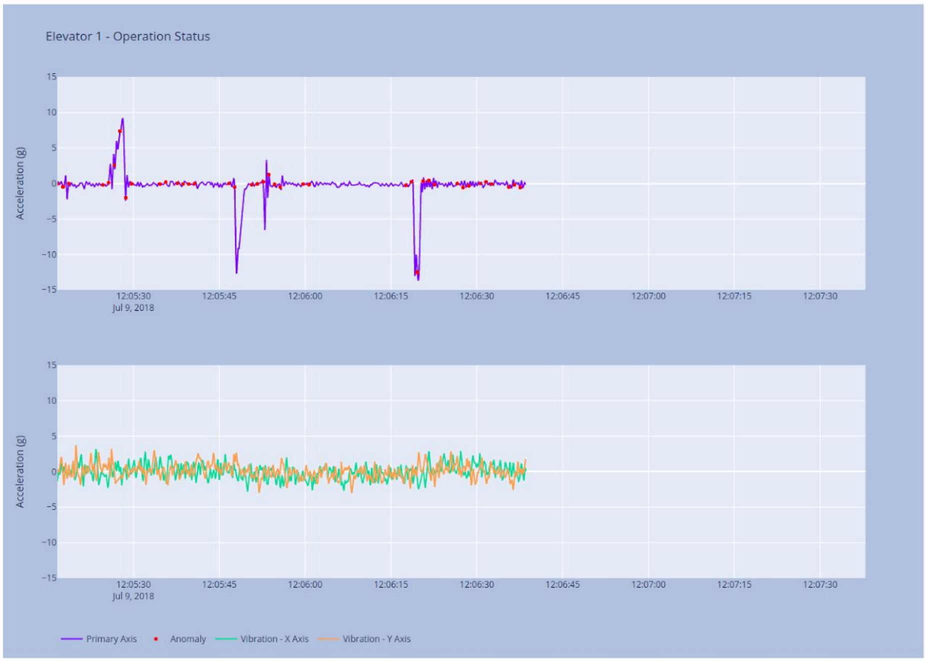


Generalized ESD Test Anomalies for the Horizontal Axes

Data Product

Our data product is a streaming dashboard built with Plotly Dash. It simulates data streaming in from an IOT sensor (accelerometer) and displays the data sequence with anomalous points detected in near real time. Data from the primary axis is displayed on the top, while the two horizontal axes (representing vibration) are displayed underneath.

The system could be used by a building manager or maintenance worker to review elevator movement and determine when and where maintenance may be required. Plotly Dash has the capability to cycle through past data and take snapshots of the plots.



Descending...

Elevator Anomaly Detection Streaming System in Progress

Lessons Learned

- Working with sensor dataset and IOT: Without any prior experience with sensor data, extensive research was required to determine the best methods to use to complete this project.
- Large dataset: In a Big Data sense, a dataset of 75 GB is not particularly large, but it is large enough that we were required to explore different storage and computing options.
- Unsupervised anomaly detection: Without a labelled dataset, model training/testing and evaluating results were challenging.
- Experimenting with various time series models: This project allowed us to explore many different time series models.

Summary

The main goal of our project was to identify and predict anomalous movement in elevators in order to curtail incidents which are on the rise in Canada. We explored a large volume of elevator acceleration data, while learning about unsupervised anomaly detection, IOT and signal processing. Extensive preprocessing and exploratory data analysis were required to better understand the data.

We experimented with various machine learning models including LSTM, Random Forest, XGBoost and Generalized ESD Test. Our LSTM model was able to detect anomalous vertical elevator movement in line with those found in literature. Anomalies on the horizontal axis, representing vibration, were detected using a Generalized ESD Test.

We developed a streaming dashboard using Plotly Dash which is used for streaming the elevator data, identifying anomaly points in the data and also presented whether the elevator was ascending or descending.

We can improve upon these findings by creating or making use of labelled data or maintenance logs which can confirm anomalous conditions. We can also experiment with other deep learning models and technologies such as H2O.ai to provide insights into various models and make comparisons between them.

References

Acada Consulting Inc. and Technical Safety BC (October 2018). Harnessing the Power of Artificial Intelligence (AI) and Internet Of Things (IoT) for Elevating Safety. Available at https://www.technicalsafetybc.ca/safety-data/iot-elevating-safety

Brownlee, J. (August 2016). A Gentle Introduction to XGBoost for Applied Machine Learning. *Machine Learning Mastery.* Available at https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning

Hayati, S. (May 2018). Prediction and Detection of Abnormal Usage of an Elevator. Master's Thesis. Available at https://trepo.tuni.fi/bitstream/handle/123456789/27038/Hayati.pdf

Kuppusamy, Manoj & Kaliyaperumal, Senthamarai. (January 2013). Comparison of Methods for detecting Outliers. *International Journal of Scientific & Engineering Research. 4.* (pp. 709-714).

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (pp. 413–422). IEEE.

Lorsbach, G.P. (no date). Analysis of Elevator Ride Quality, Vibration. (pp. 162-165). In *Educational Focus : Testing Equipment.* Available at https://www.elevatorworld.com/pdf/ed_book_chapters/ed_focus/Chapter_15.pdf.

Malhotra et al. (July 2016). LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. Presented at *ICML 2016 Anomaly Detection Workshop.* Available at https://arxiv.org/pdf/1607.00148.pdf

Nikulski, J. (February 2020). Time series forecasting with AdaBoost, random forests and XGBoost. *Towards Data Science.* Available at https://towardsdatascience.com/go-highly-accurate-or-go-home-61828afb0b13

Perkel, C. (May 2017). Elevator incidents, injuries on the rise in Canada. *Edmonton Journal.* Available at https://globalnews.ca/news/3450212/elevator-incidents-injuries-canada

Yang, T & Kaji, K & Kawaguchi, N. (July 2013). Elevator Acceleration Sensing: Design and Estimation Recognition Algorithm Using Crowdsourcing. Presented at *IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW)*. 10.1109/COMPSACW.2013.97. (pp. 534-539). Available at https://www.researchgate.net /publication/261051196_Elevator_Acceleration_Sensing_Design_and_Estimation_Recogniti on_Algorithm_Using_Crowdsourcing

Yiu, T. (June 2019). Understanding Random Forest. *Towards Data Science*. Available at https://towardsdatascience.com/understanding-random-forest-58381e0602d2