NORDIC MACHINE INTELLIGENCE

Nordic Machine Intelligence, vol. 10, pp. 11–14, 2023
Received 21 Dec 2023 / published XX XXX 2023
https://doi.org/10.2478/nmi-some-doi

Big data analytics solution in road construction

Fang, Amos [1]

1. *Singapore, Singapore, E-mail any correspondence to: amosfang@gmail.com*

Abstract

1.5% of Norwegian CO_2 emissions comes from construction machines, so trucks transporting construction material and equipment contribute to the industry's overall carbon footprint. By harnessing truck activity data from a road construction site in Viken county in the Oslo region, this work aims to automate truck activity monitoring and anomaly detection, by collecting kinematic and vibration data, processing big data and training deep learning models. When driving activity is monitored and its inefficiencies are identified, sustainability can be achieved through direct emissions cuts, by understanding operational norms and minimizing non-optimal driving behaviour. This work delivers a combined big data analytics and artificial intelligence solution, applied to a real-world scenario in the road construction context. Kinematic data is inferred from GPS tracking logs, while vibration data is obtained from motion detecting sensors found in iPad edge devices. Both data sources are reconciled, the vibration data is aggregated at 5 seconds intervals before feeding into an autoencoder neural network where its reconstruction loss is measured. A loss threshold value is set at an arbitrary percentile of all computed losses, and a sequence of observations is identified as abnormal if its loss exceeds this threshold. Finally, the anomaly detection provides new insights to truck operations and informs the decision maker about the vibration anomaly for any given load-drive-dump cycle.

Keywords: road construction; big data; anomaly detection; deep learning

Introduction

Skanska Norge AS collects lots of data from its construction sites and this data can be harnessed for insights to inform decision makers about the sustainability impact of their operations. In June 2023, the Norwegian Artificial Intelligence Research Consortium organized the RoadAI challenge and invited participants from outside the industry to demonstrate how this data can make road construction more sustainable through the use of data analytics and machine learning [1].

This work sets out to build automatic decision support tools by leveraging open-source big data and deep learning developer frameworks. These tools enable the decision maker to gain insights on truck activity at the construction site in Viken and to identify drifts or anomalies from normal operations. A standard loading and dumping trip involves a truck loading construction material or equipment, driving to its destination, unloading or dumping its carried weight and driving out of the unloading area. Some hypothetical examples of drifts may include prolonged driving on off-road surfaces, rapid truck acceleration or deceleration or abnormal loading or dumping procedures.

To process the large vibration dataset, both data ingestion and preparation steps are executed in PySpark [2], which is the Python's API for the large scale data processing capable Apache Spark. This work also exploits PySpark machine learning libraries to uncover hidden patterns from data. When the data pipeline is deployed in production, a task scheduler triggers a workflow to ingest new data. The prepared data tables are fed into the data lake for storage and later served to the data warehouse for consumption by reporting systems [3].

Finally, in order to detect anomalies within each trip cycle, the vibration time series data is fed into an autoencoder and its reconstruction loss (mean-squared error) is measured at each time step. A loss threshold value is chosen such that any sequence of timed observations incurring a predicted-to-actual loss higher than this threshold is assigned as abnormal.

The relevant data sources are GPS trips, trips information, machine (AEMP) data and truck vibration data, all of which are internally sourced within Skanska Norge AS and made available to participants who have signed the non-disclosure agreement.

Methods

The data pipeline begins with the data ingestion step, which is a process of reading files or databases with a

pre-defined schema from the relevant data sources into memory. The next step involves preparing tables to achieve the data quality required for downstream analytics and machine learning. When building production-ready data pipelines, both steps are integral to the data extract, transform and load (ETL) process.

Data Preparation

Feature engineering is a process of creating new features from original data columns. There are some features that are created from existing ones. Distance traveled in Km is calculated by applying the Haversine formula [4], which uses the change of longitude and latitude between two consecutive GPS pings and also accounts for the Earth's surface curvature. Truck speed in Km/Hr units is found by taking distance travelled over time elapsed. The magnitude of the vibration data field, *userAcceleration*, can be found by taking the Euclidean norm of its respective x, y and z coordinate values. A value of 1.0 represents the freefall acceleration of 9.81 m/s^2 [5]. The one hot encoding of categorical features, such as `massTypeMaterial` and the extracted hour of the day from `timestamp`, also creates additional features for training machine learning models.

Joining tables

Joining tables that are prepared in the previous section combines information obtained from different data sources. The truck GPS table is created by joining GPS trip logs with GPS trip information. The machine (AEMP) table contains periodic location tracking information about the construction equipment such as loaders and excavators at the construction site, but it cannot join with the truck table as both tables do not share the same primary keys. Truck vibration data can be joined with the combined GPS trip table using both `timestamp` and `tripLogId`.

The remainder of this section details the steps to join tables for the competition.

In the first join query, *joined_trips* table is the output of joining both GPS trips log and info on *TripLogId*. Conversely, *mapped_machines* table is output from joining truck to machines, but GPS data cannot join machine data directly, given that both tables do not share any common keys. Also, joining truck to machine is only possible by `timestamp`, `latitude` and `longitude`, but these table keys will not match given that a truck and a loader will not be found in the same spot at any given time. Additionally, loaders only relay back their locations once or twice daily so their position could vary a lot.

One of ideas presented in this work is to map slow moving or stationary trucks with all machines within a 48 hour period, irrespective of their location, and filtering for those that are at most 100 meters away. The accuracy can be improved further by filtering for those machines that are in proximity to the load location given by the operator and also matching the load material type which the loader that is designated to handle. The latter is not available due to data masking.

In the second join query, *vibration_df* is the output of joining vibration data with *joined_trips* on both `tripLogId` and `timestamp`. Vibration data is collected at 15Hz frequency, whereas GPS trip locations are updated at lower and more variable frequencies. Vibration data is aggregated and averaged every second before reconciling with the GPS trips table. The missing values in the joined tables, such as `speedinKmHr` are imputed with the last known values using the fill forward method.

Clustering

Having joined tables from different data sources in the previous section, GPS, speed and vibration information are available in a single table. The information can be used to perform summary statistics and machine learning. Unsupervised learning is a machine learning technique that is applied to datasets where the target variable does not exist, but discernible patterns in the data do. An example in this problem could be that driving on normal roads and within the construction site have different speed and vibration profiles, or that the load, drive and dump phases of each trip cycle can be identified by clustering. Two well known clustering techniques, k-means and Gaussian mixture models, are performed in this work.

Both *imputedSpeedKmHr* and *userAcceleration* are chosen features for clustering the dataset. The elbow method is used to determine the optimal number of clusters to predict, as the variance of the data points drops and reaches a steady value. To simplify the analysis, 3 clusters are used instead of 6 that is recommended by the method. Each cluster corresponds to on-road, off-road or at stationary for loading and dumping to occur.

Anomaly Detection

This section describes the steps to create an AI anomaly detection model using deep learning, specifically LSTM autoencoders. This approach was applied to fault detection of NASA's equipment with vibration data, as discussed in [6]. LSTM, or long short term memory, cell is a type of neuron that enhances the neural network performance on sequential data, by remembering the earlier context of a time series. The autoencoder is a type of neural network architecture used in some use cases to denoise signals. The encoder reduces the noise of the incoming signal, the latent layer captures the key features for reconstruction and the decoder learns from the reduced dimensions to produce a noise cancelling signal output. Essentially, the idea is that an abnormal vibration signal will produce a higher reconstructed output loss compared to a normal signal.

The *userAcceleration* time series is combined at intervals of 5 seconds and configured into a moving window of sequence length of 51, before feeding to the autoencoder. Each prediction is assigned to the midpoint of a sequence, so an odd number sequence length is chosen so

that its trailing and leading number of observations, when the midpoint reference is taken, are equal. The model architecture of the LSTM autoencoder is depicted in Figure 1. Dropout of 20% and L2 regularization of 0.2 are added to prevent model overfitting. The batch size is set at 32 and an Adam optimizer with learning rate of 0.0001 and momentum of 0.5 is chosen.

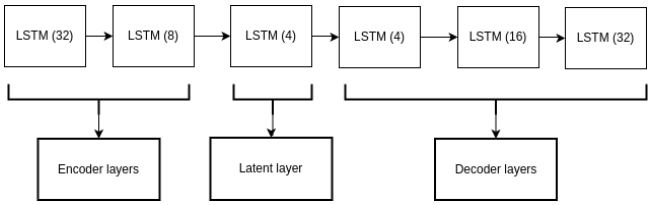

Figure 1: LSTM autoencoder neural network architecture - number of LSTM cells for each layer is indicated in parenthesis

The input to the autoencoder is a 3-dimensional tensor, with shape equivalent to (*sample size*, *sequence length*, *number of features*). The sample size is determined approximately to be the total number of observations over the batch size (32). This work used one feature, *userAcceleration*, and the MinMaxScaler() normalizes the vibration values.

After the model is trained, the threshold of the loss value is set at the 50% percentile of observed *userAcceleration* readings. The threshold can be set arbitrarily, to accomodate to operational requirements.

Results

Clustering

In Figure 2, the coloured data points indicate the cluster assignment predicted by the Gaussian mixture model algorithm [7] and are plotted on the geographical longitude and latitude axes.

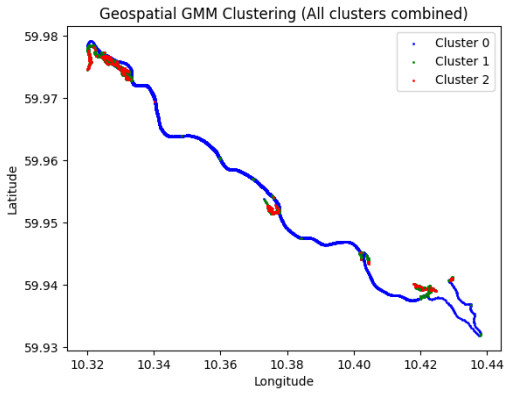

Figure 2: Combined plot of three predicted classes using Gaussian mixture model algorithm on 26th April 2023

The mean of the centroid for each cluster is tabulated in Table 1. The objective is to discover trip characteristics that explain phases in the loading and dumping cycle.

26th Apr Cluster	Mean Vibration	Mean Speed	Observed Count
0	0.171	42.67	68512
1	0.184	7.50	57789
2	0.104	0.39	69568
27th Apr Cluster	**Mean Vibration**	**Mean Speed**	**Observed Count**
0	0.157	40.12	47271
1	0.262	10.58	58239
2	0.073	1.87	38322

Table 1: Average cluster values of *UserAcceleration* and *Speed* in Km/Hr on 26th (top) and 27th (bottom) April 2023

Autoencoder model training

The model is fitted on the training and validation 26th April 2023 dataset for 20 epochs.

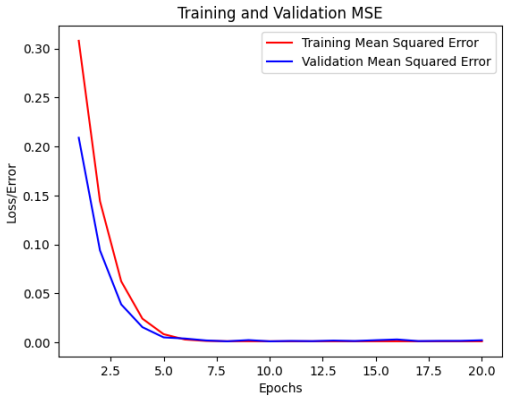

Figure 3: Training and validation loss over 20 training epochs

The model loss (mean-squared error) dropped to a steady value of about 0.0012 at around the eight epoch. The model is then evaluated on the 27th April 2023 test dataset and achieved a loss of 1.3139e^{-05}.

Figure 4 displays the loss distribution, where the mean squared error is found from the difference between the predicted and actual *userAcceleration*. The loss threshold to detect anomaly is set at the 50th percentile of all observations.

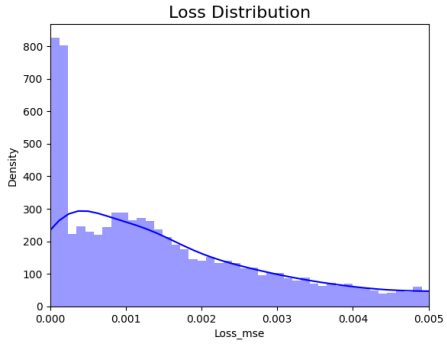

Figure 4: Loss distribution of *userAcceleration*

Anomaly detection and clustering results

The automatic detection of abnormal vibrations is shown for four trips in Figure 5. The abnormal readings are indicated in red for each trip. Truck speed is also presented in green on the same plot for visualizing the speed and vibration profiles.

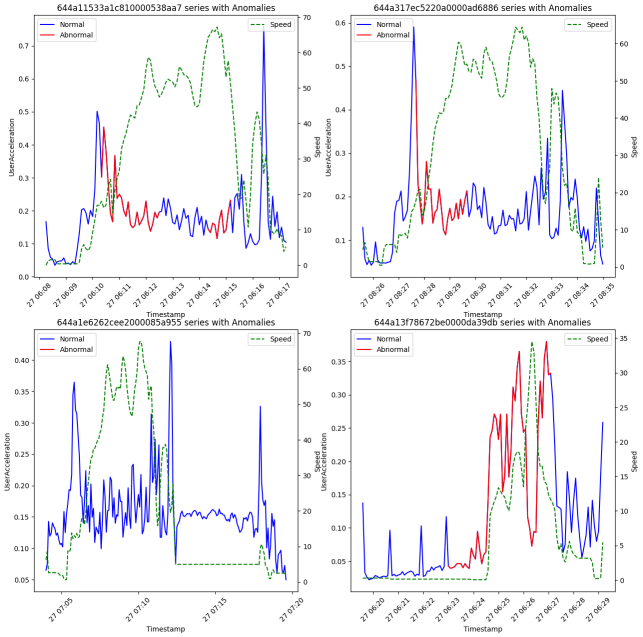

Figure 5: Anomaly prediction of four trips on 27th April 2023

The clusters that are predicted by the Gaussian mixture model are used to analyze the profile of four trips. The phases of each trip are visually represented in colours indicative of the clusters they belong to.

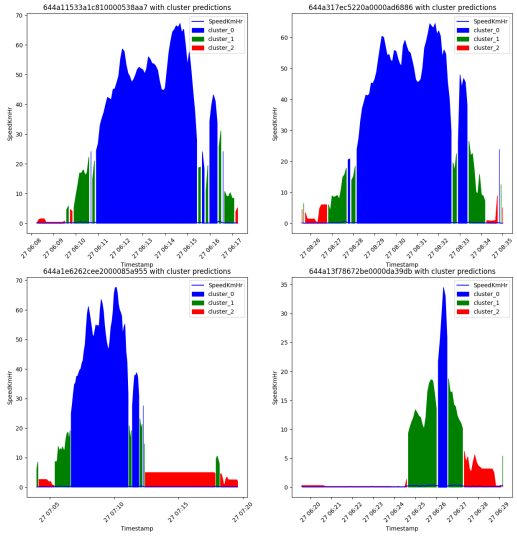

Figure 6: Clusters analysis of four trips on 27th April 2023

Discussion

Both clustering and deep learning are used to solve the automatic detection of phases and anomalies. The techniques that are employed in this work are unsupervised learning methods that discover patterns in the dataset. The clustering algorithms produced three clusters using speed and *userAcceleration* features, that relate closely to the driving phases - on-road, off-road and at stationary conditions. The green (cluster 1) and red (cluster 2) sectors correspond to areas where loading and dumping takes place.

For analysis of anomaly detection for four trips, the acceleration and deceleration phase contributed to the anomalous vibrations, seen in Figure 5. The anomaly detection is performed by a LSTM autoencoder model trained on sequences with length 51 and each observation is averaged every 5 seconds.

In this work, two data pipelines were built to demonstrate the ETL (extract, transform and loading) steps. Phase detection is accomplished by clustering and anomaly detection is performed by an autoencoder.

Conflict of interest

Author states no conflict of interest.

References

1. Signer Riemer-Sørensen Helga Margrete Bodahl Holmestad LH. RoadAI: Reducing emissions in road construction. Nordic Machine Intelligence 2023; 1:1–4. DOI: 10.5617/nmi-some-doi
2. Apache Spark Python API Documentation. The Apache Software Foundation. Available from: https://spark.apache.org/docs/latest/api/python/index.html
3. Mertens O and Van Baelen B. Azure Data and AI Architect Handbook. Packt Publishing, 2023. Chap. 3 - Batch and streaming ingestion. Available from: https://books.google.com.sg/books?id=64XPEAAAQBAJ
4. scikit-learn Documentation - Haversine Distances. 2023. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.haversine_distances.html
5. Apple Developer Article, Getting raw accelerometer events, Retrieve data from the onboard accelerometers. 2023. Available from: https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events
6. Larzalere B. LSTM Autoencoder for Anomaly Detection. Accessed: Sep 25, 2019. 2019. Available from: https://towardsdatascience.com/lstm-autoencoder-for-anomaly-detection-e1f4f2ee7ccf
7. Apache Spark - pyspark.ml.clustering.GaussianMixture. The Apache Software Foundation. Available from: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.clustering.GaussianMixture.html