Data Preprocessing for Machine Learning Models

Rawan El Moghrabi
ECE Department
Western University
London, Canada
relmoghr@uwo.ca

Ruiqi Tian
ECE Department
Western University
London, Canada
rtian25@uwo.ca

Luisa Liboni
ECE Department
Western University
London, Canada
luisa.liboni@uwo.ca

Miriam Capretz
ECE Department
Western University
London, Canada
mcapretz@uwo.ca

Abstract— Data preprocessing is an essential step when building machine learning solutions. It significantly impacts the success of machine learning modules and the output of these algorithms. Typically, data preprocessing is made-up of data sanitization, feature engineering, normalization, and transformation. This paper outlines the data preprocessing methodology implemented for a data-driven predictive maintenance solution. The above-mentioned project entails acquiring historical electrical data from industrial assets and creating a health index indicating each asset's remaining useful life. This solution is built using machine learning algorithms and requires several data processing steps to increase the solution's accuracy and efficiency. In this project, the preprocessing measures implemented are data sanitization, daylight savings transformation, feature encoding, and data normalization. The purpose and results of each of the above steps are explained to highlight the importance of data preprocessing in machine learning projects.

Keywords— Machine Learning, Predictive Maintenance, Data Preprocessing, Feature Encoding

I. Introduction

This project collected raw data from smart electrical meters installed at a Golf Club facility in Mississauga, Ontario. The smart meters record a range of data for each connected asset. Historical and live data from the smart meters are obtained from circuit meters API. Even though the facility has approximately 84 assets connected to smart meters, only five assets were chosen for this project. The project aims to implement predictive maintenance (PdM) methodologies and create a health index for each asset that predicts the remaining useful life. The health index would then be a tool to maintain the industrial equipment at the most optimal time. Thus, the project would decrease maintenance costs and maximize an equipment's useful life.

As seen in figure 1, before the machine learning algorithms can be implemented in the PdM framework, the raw data obtained for each asset must undergo a series of preprocessing steps and transformations to achieve the desired inputs and accurate outputs [1]. These processes may include sanitization, feature engineering, normalization, and transformation [2]-[3]. The steps completed during data preprocessing and the transformations performed during this project stage largely determine a machine learning model's predictive behaviour [4]. Therefore, it is essential to perform the processing steps correctly. An ML model's accuracy becomes compromised without proper data preprocessing, and the resulting output would be ineffectual.

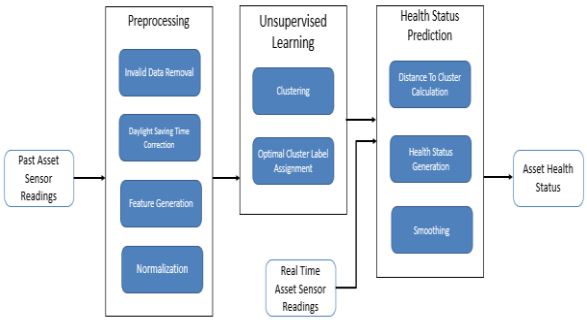

Figure 1 Project Overview

II. Background

A. Machine Learning

Machine Learning (ML) is a type of artificial intelligence. It consists of algorithms that take in an input of data and learn from it to present the desired output [5]. There are four main types of machine learning: supervised, semi-supervised, unsupervised, and reinforcement learning. Each of these categories has its own established implementations. In this project, unsupervised learning is used to implement the PdM framework. A clustering algorithm, which creates categories from unlabeled data, is developed to achieve the goal of this project.

B. Predictive Maintenance

Predictive Maintenance (PdM) utilizes machine learning algorithms to create a new category of maintenance which supports smart manufacturing and industrialization [6]. Two main maintenance types are currently implemented in industrial settings [7]. The first type is preventative maintenance, in which maintenance is scheduled at regular intervals to avoid equipment failure. The other type of maintenance is corrective maintenance which relies on repairing equipment after failure. PdM is a way to perform

maintenance at the optimal time for the asset. This solution prevents downtime from failed equipment and eliminates any unnecessary maintenance.

C. Data Acquisition

For this project, five assets were chosen from the list of assets provided by the Golf Club. The criteria set out to choose the final list of assets are as follows:

1. Assets with the most data points available.
2. Assets that were least affected by the Covid-19 lockdowns.
3. Assets whose usage was dependent on external factors such as temperature.

Based on the above considerations, the five chosen assets were four air conditioning units and one electric heater. A python script requested data from the Circuit Meter API in compliance with the company-provided documentation. According to the API documentation, historical data could only be acquired in increments of 15 minutes, while live data is available in increments of two seconds. Note that the API used in this project could only provide live data up to 24 hours from the request. The data received dates back to the year 2019 when the Golf Club facility installed smart meters. To satisfy the project criteria, the python script queried the API with requests for predetermined electrical metrics. The metrics obtained included voltage, current, power, apparent power, reactive power, frequency, external humidity, and ambient temperature. The Python script implemented a predetermined delay to avoid overwhelming the API.

III. Preprocessing Method

A. Data Sanitization

Data sanitization is an essential first step to improving the overall data quality used for the machine learning algorithms. Its main goal is to perform noise treatment, reduce redundancies, and identify outliers [8]. in this step, a Python script was developed to remove any null values and identify the outliers in the data. The Python script iterated the entire data set to remove empty columns. In addition, it was responsible for identifying any metrics that were requested from the API but did not have any data stored.

B. Daylight Savings Correction

One of the project's goals is to identify, using a clustering algorithm, the times of the year when the assets are most in use. To do that, the data must accurately represent the local time for each asset. Furthermore, since the assets are located in a region where daylight savings applies, a transformation had to be applied to account for the variation due to daylight savings. This transformation was also applied using a Python script. The script takes the timecode provided from the API in UNIX time and converts it into a DateTime object. Then, a custom function takes in this conversion and shifts the time by an hour when daylight savings was in effect. The conversion was not transformed for periods where standard time was applied. The daylight savings correction provided more accurate input data, increasing the algorithm's efficiency.

C. Feature Encoding

Feature encoding is another crucial step in data preprocessing. This step creates specific features from the raw data provided [9]. Typically, raw data does not include human-understandable features. ML algorithms have limited success without these features and produce outputs with less efficacy. Features need to be encoded to represent real-life phenomena to counteract this problem. Examples of these features are cyclical features such as hours, weeks, and months. These features need to be encoded in such a way to preserve their cyclical aspect [10]. One way to do that is to create a sine-cosine coordinate system representing the cyclical characteristics of date and time objects. Figure 2 shows an example of this coordinate system as it applies to daily hours.

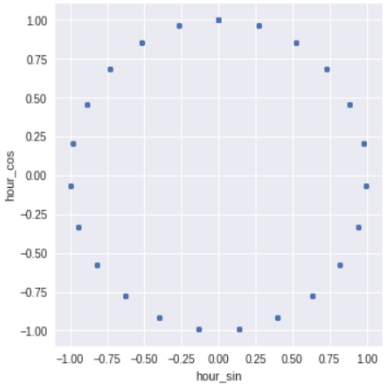

Figure 2 Plot of two-dimensional hour feature

As seen from this graph, the day hours would appear linear without cyclical encoding. If these periodic attributes are not considered and encoded, the DateTime objects cannot be meaningfully used in model training.

D. Data Normalization

Data normalization is the process in which the available data set is scaled or transformed to fit a specific range giving each feature an equal contribution [11]. The resulting values are all within a new range of 0 to 1 unless a value for x is provided the exists outside the min and max bounds. Equation (1) is used to normalize values where min is the minimum value for x and max is the maximum value for x.

$$y = (x - min) / (max - min) \quad (1)$$

This process is usually the last step of data preprocessing and is applied to improve the data quality by ensuring that each feature has equal weight.

IV. Results

After applying the preprocessing method, the raw data underwent a transformation that improved its quality and its representation of naturally occurring phenomena. Table 1 shows an example of the first five data points of asset #5173, which is one of the air conditioners. In the table, the voltage metrics can be seen as well as the timecodes which represent seconds since epoch.

Table 1: Raw Voltage Data for Asset #5173

Timecode	Volts_avg	Volts_std	VoltsA_avg
1.66E+09	344.6	0.1	344.4
1.66E+09	345.2	0.2	344.8
1.66E+09	346.1	0.1	345.9
1.66E+09	346.6	0.1	346.3
1.66E+09	347.6	0.1	347.5

Once the data sanitization Python script was applied, the power metrics were removed from the data as they were null values. Moreover, any columns that included periods before the smart meters began recording were removed as they did not contain any data. The data sanitization script also reorganized the data in ascending time order.

After the data sanitization was done, daylight saving correction was applied. This Python script applied modern concepts in software engineering and reusable Python functions to apply a daylight saving pattern representing the assets' local time. Table 2 shows the added columns that indicate the date and time represented by that timecode. The time shown in the table is in local time, which in this case is either Eastern Standard Time (EST) or Eastern Daylight Time (EDT).

Table 2: First five entries after Data Sanitization and Daylight Savings Correction

Timecode	Year	Month	Day	Hour	Minute	Weekday	Volts_avg
1.56E+09	2019	7	17	8	0	3	0
1.56E+09	2019	7	17	8	15	3	0
1.56E+09	2019	7	17	8	30	3	0
1.56E+09	2019	7	17	8	45	3	0
1.56E+09	2019	7	17	9	0	3	0

After that, a linear transformation on the axes using a rotation matrix of cosines and sines was applied to encode the cyclical attributes of datetime objects. Table 3 shows an example of the sine and cosine coordinates for the month, day, and hour features.

Table 3: Example of the sine and cosine coordinates for select date and time features

month_sin	month_cos	day_sin	day_cos	hour_sin	hour_cos
-0.23932	-0.97094	-0.40674	-0.91355	0.866025	-0.5
-0.23932	-0.97094	-0.40674	-0.91355	0.866025	-0.5
-0.23932	-0.97094	-0.40674	-0.91355	0.866025	-0.5
-0.23932	-0.97094	-0.40674	-0.91355	0.866025	-0.5
-0.23932	-0.97094	-0.40674	-0.91355	0.707107	-0.70711

Finally, data normalization was applied using a Python that applies (1) to ensure that each feature was weighed equally.

V. Conclusion

The data processing applied in this research directly supports the data-driven decision-making predictive maintenance framework currently under development in the software engineering lab at Western University. Using modern Python scripts and software engineering principles, the data set was improved by eliminating outliers, applying real world patterns, and encoding cyclical attributes to the data features.

Data preprocessing represents one of the significant steps of machine learning projects. It directly impacts the success of learning modules and the accuracy of the outputs. Data processing is essential to ensure that raw data is transformed into understandable data representing the real world and producing outputs applicable to real-life situations.

References

[1] J. Nalić and A. Švraka, "Importance of data preprocessing in credit scoring models based on data mining approaches," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018, pp. 1046-1051, DOI: 10.23919/MIPRO.2018.8400191.

[2] Y. Huang, M. Milani, and F. Chiang, "PACAS: Privacy-aware, data cleaning-as-a-service," in 2018 IEEE International Conference on Big Data (Big Data). IEEE, dec 2018.

[3] C. Li, 'Preprocessing Methods and Pipelines of Data Mining: An Overview'. arXiv, 2019.

[4] awakuli, D. Kaiser and T. Engel, "Synchronized Preprocessing of Sensor Data," *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 3522-3531, doi: 10.1109/BigData50022.2020.9377900.

[5] P. Domingos, "A few useful things to know about machine learning," Communications of the ACM, vol. 55, no. 10, pp. 78–87, 2012.

[6] W. Zhang, D. Yang, και H. Wang, 'Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey', IEEE Systems Journal, τ. 13, σσ. 2213–2227, 9 2019. doi: 10.1109/JSYST.2019.2905565.

[7] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, και S. G. S. Alcalá, 'A systematic literature review of machine learning methods applied to predictive maintenance', Computers & Industrial Engineering, τ. 137, σ. 106024, 11 2019. doi: 10.1016/J.CIE.2019.106024.

[8] García, S., Ramírez-Gallego, S., Luengo, J. *et al.* Big data preprocessing: methods and prospects. *Big Data Anal* **1,** 9 (2016). https://doi.org/10.1186/s41044-016-0014-0

[9] M. Oyamada, "Extracting Feature Engineering Knowledge from Data Science Notebooks," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 6172-6173, doi: 10.1109/BigData47090.2019.9006522.

[10] T. Mahajan, G. Singh, G. Bruns, G. Bruns, T. Mahajan, και G. Singh, 'An Experimental Assessment of Treatments for Cyclical Data', στο Proceedings of the 2021 Computer Science Conference for CSU Undergraduates, Virtual, 2021, τ. 6.

[11] D. Singh, B. Singh, 'Investigating the impact of data normalization on classification performance', *Applied Soft Computing*, τ. 97, σ. 105524, 2020.