A New Data Analytics Framework Emphasising Pre-processing in Learning AI Models for Complex Manufacturing Systems

Caoimhe M. Carbery[1,2(✉)], Roger Woods[1], and Adele H. Marshall[2]

[1] Electronic Computer Engineering, Queen's University Belfast, Belfast, UK
{ccarbery02,r.woods}@qub.ac.uk
[2] Mathematical Sciences Research Centre, Queen's University Belfast, Belfast, UK
a.h.marshall@qub.ac.uk

Abstract. Recent emphasis has been placed on improving the processes in manufacturing by employing early detection or fault prediction within production lines. Whilst companies are increasingly including sensors to record observations and measurements, this brings challenges in interpretation as standard approaches for artificial intelligence (AI) do not highlight the presence of unknown relationships. To address this, we propose a new data analytics framework for predicting faults in a large-scale manufacturing system and validate it using a publicly available Bosch manufacturing dataset with a focus on pre-processing of the data.

1 Introduction

Manufacturing is highly competitive and companies have made considerable investments to improve their production analysis capabilities by adding sensors to record information as products undergo manufacture [1]. The abundance of data from multiple sources and in different formats that is monitored continuously, creates challenge for analysis. Moreover, the data may not have been recorded properly, resulting in missing data which has the potential of severely impacting subsequent modeling systems and biasing results. *Missingness* can be due to faults in a machine or sensor, occurrence of noise during processing, power shortages, or some other issues [2]. In addition, imbalanced classes can occur as a result of under-represented classes, such as in binary classification where there can be a majority and minority class. Research has highlighted the problems in assessing classifiers as errors result in inaccuracy as the system is biased towards the majority class. The work here focuses on binary classification where we want to determine whether a product will be grouped into the minority (failure) class dependent on the input parameters.

K. Li et al. (Eds.): ICSEE 2018/IMIOT 2018, CCIS 924, pp. 169–179, 2018.
https://doi.org/10.1007/978-981-13-2384-3_16

Review of Relevant Literature: The data in modern manufacturing challenges can suffer from high dimensionality, complexity, non-linearity and inconsistencies [1–4]. To address these challenges, machine learning and data analytics methods have been employed [2–5], which concentrate on predictive maintenance and rare event prediction [6].

Lee et al. [1] presented a cyber-physical system with a case study as an approach to monitor the behaviour of machines using sensor data for Industry 4.0; they also highlighted the need for further work to improve generalisability of the system. Susto et al. [3] presented a new multiple classifier model for predictive maintenance along with a simulation study and benchmark dataset; the data needed to be pre-processed in order to allow for a suitable classifier such as k-NN and support vector machines (SVM), to be trained. In [5], different ML methods were compared for a semiconductor manufacturing dataset and highlighted the benefits of reducing the data through feature selection. Work in [7] emphasised the benefits of AI and ML for manufacturing, but there has been little investigation into the statistical basis of data preprocessing to improve model performance and learning procedures. Kotsiantis et al. [8] has shown the major impact that inefficient data can have on machine learning models.

Motivation and Overview: with increasing complexity in manufacturing processes, machine learning algorithms are being used to ensure earlier detection of defects, improve production performance and prediction of future performance [5]. A framework is presented that collates, pre-processes and generates training data for manufacturing and allows behaviour to be identified that can influence the production outcome. We present an approach that allows AI systems to be built for behavioural analysis and information extraction to be performed which will help engineers to improve machine performance and aid future decision making

This paper is organised as follows; Sect. 2 outlines the framework for analysing manufacturing systems which can be applied to large, inconsistent, imbalanced datasets. Section 3 presents the Bosch case study used to test the framework and create results. Section 4 presents the results of our algorithm and covers conclusions.

2 Manufacturing Data Framework

In this research, we focus on the crucial stages of pre-processing, selecting suitable algorithms and interpretation of the results to generate a suitable method for providing feedback to manufacturing engineers. Our framework shown in Fig. 1, involves four key stages to produce an appropriate learning model [9].

Raw, real-world data which is unstructured and inconsistent in nature, often involves large dimensions, class imbalance issues and missing instances. Therefore its collation is often challenging and involves combining data from different sources e.g. from sensors of varying machines etc. and processing it appropriately.

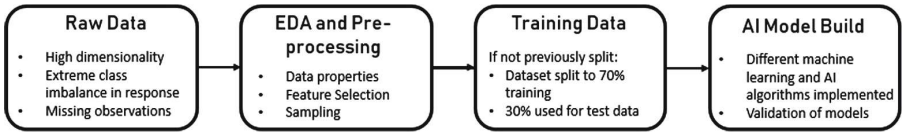

Fig. 1. Visualisation of proposed data analysis design flow for manufacturing data

2.1 Exploratory Data Analysis and Pre-processing

Exploratory data analysis (EDA) and pre-processing are crucial stages in preparing data for AI algorithms [8]. Manufacturing data can contain a large amount of redundant information which if blindly fed into a learning model, can result in a biased or unreliable outcome. Pre-processing can have a critical impact on model performance, therefore effort has been spent on standard approaches e.g. filtering and normalising to ensure that the training dataset is of an appropriate format whilst ensuring that no bias has been introduced.

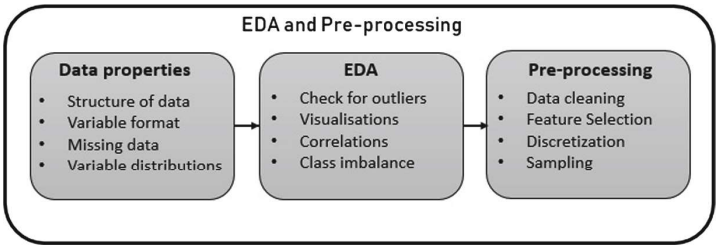

Fig. 2. Detail of the EDA and pre-processing steps required in the framework

Data cleaning removes redundant and unsuitable variables by investigating feature variance and removing those which have near to zero variance as these would not provide useful information to the model build and would only complicate the learning process. Incomplete data is unavoidable but we must try to have reasoning behind our choice for handling *missingness* whilst trying to not influence our model. We can choose from a number of methods which aim to handle missing data [10]. The most common are:

1. Remove any instance with at least one unknown variable value;
2. Mean substitution;
3. Treat missing values as a unique value;
4. K-nearest neighbour imputation method.

For our analysis, there is too much missingness to merely discard these instances, so the chosen approach, as demonstrated by other researchers [6], is to rescale the data and assign the missing instance an unique value, or if the data is of a discrete/categorical format, assign the missing instances an independent group.

To improve model performance and reliability, two feature selection approaches are performed to reduce the number of variables, namely wrapper and embedded methods both of which produce results to indicate the features that are most influential and important. The features retained must encode as much information about the system as possible in order for the final classifier model to perform well. As it is clear that a reduced feature count will improve both performance and accuracy, we want to ensure that we do not remove any features that could be influential to the model outcome [8].

Wrapper methods use a classifier model and conduct an extensive search in the space of subsets of features to determine optimal performance to produce a ranking of features. Often they are superior to filtering approaches, yet they require a larger amount of computation as they involve investigating a large search space. Embedded methods, however, can be seen as a balance between the two approaches as they use internal information of the model. Thus, we have implemented an embedded method to identify the key features for our learning model namely an extreme gradient boosting tree (XGBoost) and thus can determine the most influential and important features for building a suitable classifier [11]. XGBoost generates importance measures based on the number of occasions that a feature is selected for splitting trees in the algorithm.

Sampling methods are used if the chosen AI model cannot handle imbalanced data and re-sample the data to either increase the minority class or else reduce majority class [10]. Imbalanced data is prevalent in cases of anomaly detection or rare events, where some ML algorithms could provide biased and inaccurate results. This is a result of ML algorithms aiming to improve accuracy and not considering the distribution of the class variable. The most common approaches are the synthetic minority over-sampling technique (SMOTE) or ensemble methods which combine weak learners to create stronger learning models.

2.2 Learning Model

Extreme gradient boosting classification trees have the ability to not only uncover important data features, but to construct a robust classification model. It is a popular choice among classification models due to its simple implementation [11]. XGBoost involves the construction of an ensemble of multiple *weaker* trees i.e. small trees. In order to utilise XGBoost models, the data must be in a numeric format. In the first instance of feature selection (see Sect. 2.1), now we can run the learning algorithm on these important variables with multiple iterations to generate a powerful classification model.

2.3 Model Validation

The suitability of an intelligent classification model can be evaluated using standard statistical metrics such as accuracy, sensitivity, specificity, precision and F-measure [12]. We utilised our model and testing dataset to produce values to asses the suitability of our chosen classifier and assess its performance. We calculated the number of correctly classified positive samples (true positives),

number of correctly recognised as not in the class (true negative), count of samples that were incorrectly assigned a class (false positive) and those who were not recognised as being in the correct class (false negatives), each denoted by tp, tn, fp, fn respectively [12]. These are used to construct confusion matrices which provide values that are used for calculating the common performance measures to evaluate classification models, for this paper, binary classification (Table 1).

Table 1. General format of confusion matrix.

		True Value		
		Positive	Negative	Total
Predictive	Positive	tp	fp	$tp + fn$
	Negative	fn	tn	$fn + tn$
	Total	$tp + fn$	$fp + tn$	N

The measures are highlighted as follows:

- Accuracy: indicating overall effectiveness of a classifier, calculated using the formulae $\frac{tp+tn}{N}$ but is biased when class imbalance is not addressed;
- Sensitivity and specificity analysis provide values to evaluate the effectiveness of the classifier to identify positive and negative labels respectively, and are given by $Sensitivity = \frac{tp}{tp+fn}$ and $Specificity = \frac{tn}{fp+tn}$;
- Precision is a measure of a class agreement of the data labels with the classifiers output labels, calculated by $\frac{tp}{tp+fp}$;
- F-measure is calculated by $2\ \frac{precision.sensitivity}{precision+sensitivity}$ and is more robust to imbalanced data [12].

3 Bosch Manufacturing Case Study

Bosch provided a large anonymised dataset representing one of their production lines with an aim of utilising methods to try to predict the outcome of products and is available on Kaggle [13]. This dataset is one of the largest publicly available manufacturing datasets (14.3 Gb), containing approximately 1.2 million observations and over 4,000 features. The only information provided is the manufacturing line and station associated with each feature which is contained within the variable names e.g. *L1_S24_F1695* indicates that Feature 1695 was observed at Station 24 on Line 1.

The datasets were split into three categories; date, categorical and numeric. Within each of these groups, Bosch have provided the data separated for training and testing thus avoiding in this case, the third stage in Fig. 1. The training sets contain the variable *Response* where a value of 1 indicates a product has failed quality control, and 0 otherwise. No response variable is included in the test

dataset as this is the value that our model aims to predict. The quality of the products is extremely high as only 0.58% of products fail at the final testing stage, thus introducing a major class imbalance issue with the data. Figure 3 depicts an example of the flow of a product across the factory floor, highlighting the numerous stations associated with different lines in the build.

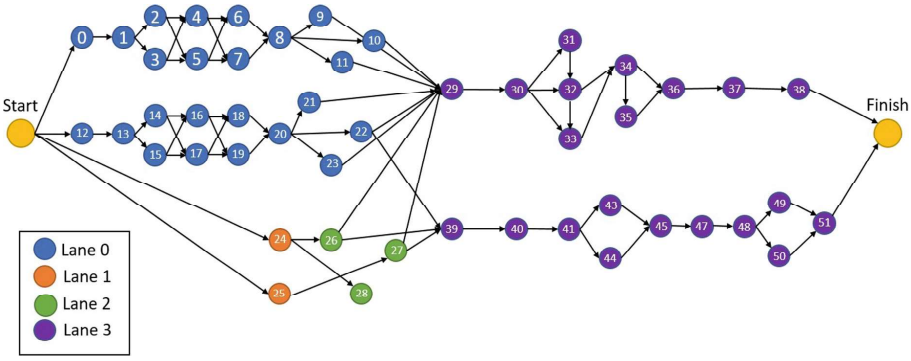

Fig. 3. Example flow of Bosch factory floor depicting stations as circular nodes.

Table 2. Overview of data used for the analysis of Bosch manufacturing.

Data characteristics	Total
Variables	986
Rows	1183747
Lines	4
Stations	51
Percentage missing	78.5
Percentage fail	0.58

3.1 Exploratory Analysis

In the first instance, we perform EDA to identify key properties of the Bosch dataset to identify correlation, redundant variables, underlying structure and issues within the data.

Data Properties: The Bosch manufacturing dataset consists of over 2.4M jobs, each of which have an associated ID and 4364 variables. These variables/features represent either numeric, categorical or date measurements. We performed analysis to determine the proportion of missing observations per feature and also a count of missing observations per ID. Initial investigation into the categorical

features indicate an issue of extreme sparsity (around 99% missing) and thus is not included in this paper as done in [6]. Our analysis has focused on the numeric data as preliminaries found it to be most influential, therefore categorical and date variables were not within the scope of this study. Table 2 provides a summary of the dataset used for the research in this paper.

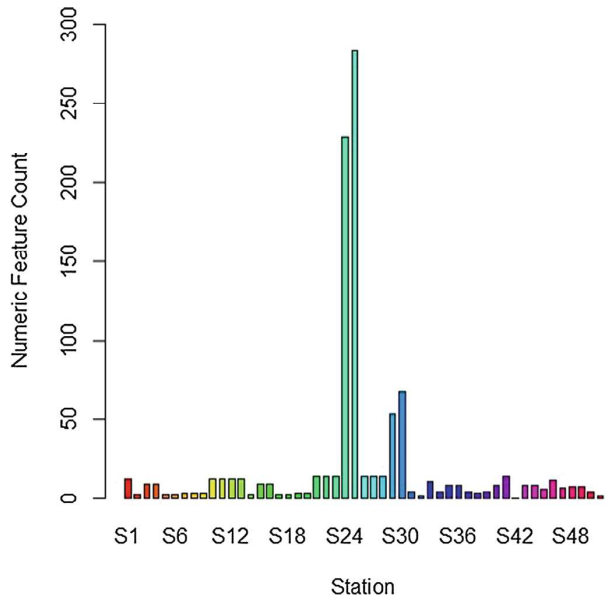

Fig. 4. Bar plot showing the number of features associated with the individual stations of the Bosch production line.

Alongside the properties of the dataset content, a number of other characteristics should be noted prior to any analysis. The chosen processing stages and algorithms must be able to account for each of these challenges if we are to appropriately model the data without introducing bias.

- As the data is anonymised, no expert knowledge can be employed to indicate the higher importance features and learning is fully data-driven.
- Missing observations represents up a large proportion of the data and could be where a product may not pass through a particular station.
- No information is related to each ID, so we could postulate that the manufacturing process involves a number of different products where they may not undergo the same processing steps.
- As the data set is large, any learning procedure must have the capabilities of processing the data of this scale.
- High class imbalance is present within the response variable as only 0.58% of products fail at the final testing stage.

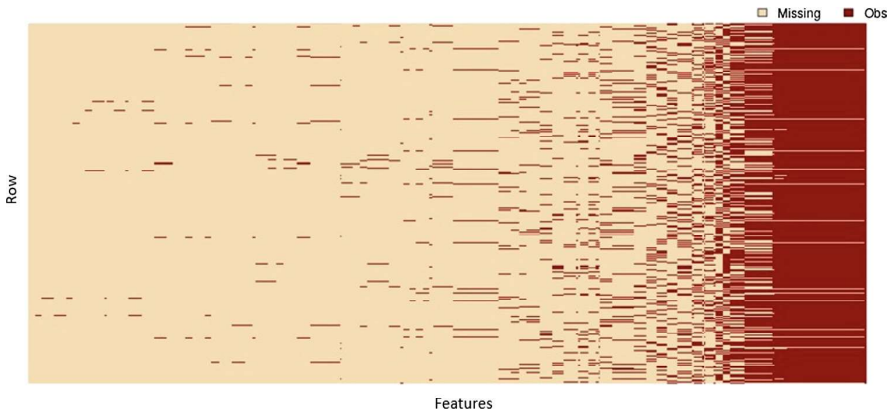

Fig. 5. Visualisation of missing versus observed instances within the Bosch data

Figure 4 shows the count of numeric features associated with each station. Stations 24, 25, 30 and 31 contain the largest number of features, so we assume that these stations process more products and could be more influential.

3.2 Pre-processing

Initial analysis was performed to check for outliers in the features through visualisations of the distributions. Correlations between features as well as the response were calculated. This demonstrated that features from the later stages of the build were more highly correlated than those from earlier in the process. The class imbalance is high, therefore if this is not handled appropriately, any model built with this data will result in a biased approach predicting that the product to be in the majority class i.e. pass. Before implementing sampling methods to handle class imbalance, a number of stages of preprocessing are necessary.

Data Cleaning: Duplicated rows were removed as they provided no further information. Variances for each feature were calculated allowing removal of redundant features with zero variance. Our feature count reduced to 158. Whilst this reduces the dimensionality of a dataset, the relation of the features with the response variable can also be investigated. Figure 5 shows missing data observations in the dataset where the lighter shaded portion represents missing data. It is clear that the later stages are where more information is recorded and would appear in the final model. This needs to be accounted for and our approach was to create an independent category for when an instance was unobserved, by performing discretization on the 158 features to include another factor representing *unobserved* instances.

Feature Selection: Feature selection allows selection of key influential variables which influence the outcome whilst improving the predictive accuracy and

improving interpretability. Here we used the top 50 features indicated from the algorithm and their associated observations to train a new XGBoost classifier model. Table 3 shows an example of the 'Gain' values produced by XGBoost.

Sampling: To account for the extreme class imbalance, one must consider sampling methods to rebalance the class variable, but investigation into the XGBoost algorithm demonstrated its robustness to imbalanced data and was not performed for this initial analysis. However, sampling methods must be considered for our general data analysis framework when implementing alternative learning algorithms (Fig. 6).

Table 3. Example of six variables from XGBoost which show the accuracy of model gained by retaining these features.

Feature	Gain
L1_S24_F1723	0.5328070
L1_S24_F1846	0.2248599
L1_S24_F1632	0.1162531
L1_S24_F1695	0.0611954
L3_S34_F3876	0.0403588
L2_S26_F3036	0.0132253

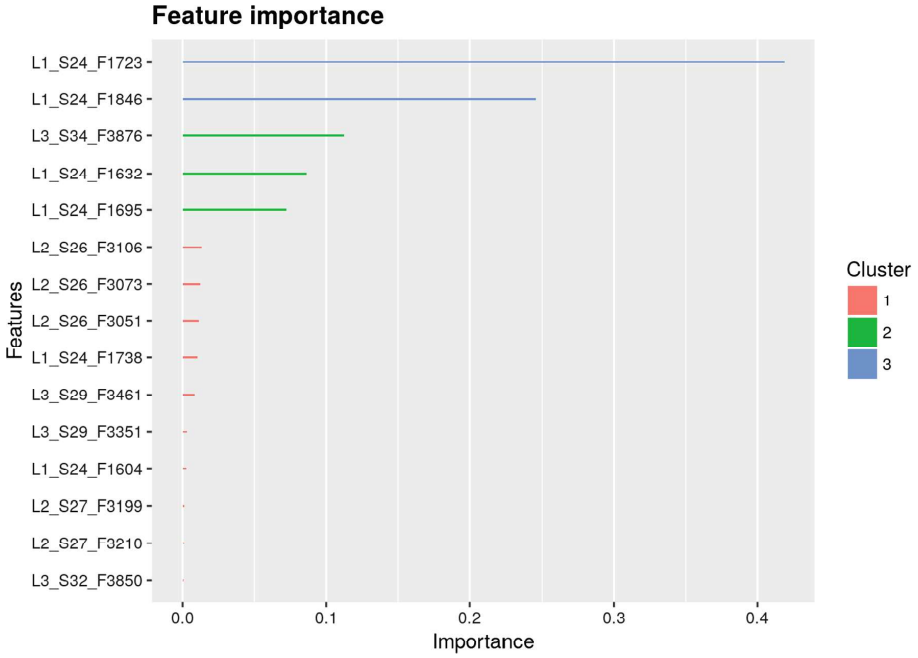

Fig. 6. Output from XGBoost showing the top 15 features of importance

4 Conclusion

In this paper, a new framework has been presented which combines useful analytics tools into a different format from those previously implemented. Using a widely available dataset from Bosch, an appropriate training dataset containing 50 features was produced, allowing an extreme gradient boosting tree to be used as a classification prediction model. Using the framework (Fig. 2), data preprocessing and exploratory analysis was used to create a reduced data size highlighting the most influential features. This allowed us to perform an R implementation of an extreme gradient boosting (XGBoost) model [11] and employ R's inbuilt performance metrics to demonstrate a high accuracy and F-measure. The research highlights the necessity to pre-process and we are currently working with a commercial partner to apply the research to their system to produce an automated system for performing the analysis on manufacturing data.

References

1. Lee, J., Kao, H., Yang, S.: Service innovation and smart analytics for industry 4.0 and big data environment. Proc. CIRP **16**, 3–8 (2014)
2. He, Q.P., Wang, J.: Statistical process monitoring as a big data analytics tool for smart manufacturing. J. Proc. Control **67**, 35–43 (2017). https://doi.org/10.1016/j.jprocont.2017.06.012
3. Susto, G., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine learning for predictive maintenance: a multiple classifier approach. IEEE Trans. Industr. Inf. **11**(3), 812–820 (2015)
4. Wuest, T., Weimer, D., Irgens, C., Thoben, K.D.: Machine learning in manufacturing: advantages, challenges, and applications. J. Prod. Manufact. Res. **4**(1), 23–45 (2016). https://doi.org/10.1080/21693277.2016.1192517
5. Moldovan, D., Cioara, T., Anghel, I., Salomie, I.: Machine learning for sensor-based manufacturing processes. In: Intelligent Computer Communication and Processing (ICPP) 2017, pp. 147–154. IEEE (2017)
6. Zhang, D., Xu, B., Wood, J.: Predict failures in production lines. In: IEEE International Conference on Big Data, pp. 2070–2074. Washington, USA (2016)
7. Lee, K., Cheon, S., Kim, C.: A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. IEEE Trans. Semicond. Manufact. **25**(5), 1167–1180 (2014)
8. Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E.: Data preprocessing for supervised learning. Int. J. Comput. Sci. **1**(2), 111–117 (2006)
9. Carbery, C.M., Woods, R. Marshall, A.H.: A Bayesian network based learning system for modelling faults in large-scale manufacturing. In: IEEE International Conference on Industrial Technology 2018, pp. 1357–1362. France (2018)
10. Chawla, N.V.: Data mining for imbalanced datasets: an overview. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09823-4_45

11. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: 22nd International Proceedings on knowledge discovery and data mining, pp. 785–794. ACM (2016)
12. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Proc. Manag. **45**(4), 427–437 (2009)
13. Kaggle.com: Bosch production line performance (2016). https://www.kaggle.com/c/bosch-production-line-performance. Accessed Nov 2017