A new data analytics framework emphasising preprocessing of data to generate insights into complex manufacturing systems


Published in:

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2019 SAGE. This work is made available online in accordance with the publisher’s policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
A new data analytics framework emphasising preprocessing of data to generate insights into complex manufacturing systems

Caoimhe M. Carbery¹,², Roger Woods¹ and Adele H. Marshall²

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 do not highlight the presence of unknown relationships. To address this, we have proposed a new data analytics framework for predicting faults in a large-scale manufacturing system and validated it using both a publicly available Bosch manufacturing dataset with a focus on preprocessing of the data and the open source SECOM industrial dataset. This paper is an extension to the work presented at International Conference on Intelligent Manufacturing and Internet of Things. The additional material includes a detailed focus on feature selection and the various approaches for identifying important features in the data, an updated framework methodology and description, an extension of XGBoost to allow this model to be used for prediction/classification and a comparison for classification with a Random Forest, tree-based model. The framework was used to explore two public manufacturing datasets and successfully identified the most influential features related to product failure in each production line data.

Keywords
Data analytics, manufacturing systems, classification, preprocessing

Introduction
Manufacturing is highly competitive and companies have made considerable investments to improve their production analysis capabilities by adding sensors to record information during the product manufacturing process¹. The abundance of data from multiple sources, often in different formats, that is continuously monitored creates a challenge for manufacturing data 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 issues². 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 an 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 based on the input parameters. The development of a suitable framework for generating insights into unknown data sources can provide further understanding of the manufacturing process.

Manufacturing Research Background
There has been extensive review papers in recent years focusing on data analytics, machine learning (ML) and data mining as a tool for improving the analysis of manufacturing systems³,⁴. The data in modern manufacturing production lines can suffer from high dimensionality, complexity, non-linearity and inconsistencies¹,²,⁴,⁵. To address these challenges, ML and data analytics methods have been employed²,⁴,⁶, which concentrate on predictive maintenance and rare event prediction⁷. Such approaches for the challenges of analysing industrial data are considered next.

ML and data analytics
Lee et al.¹ presented a cyber-physical system and case study to monitor the behaviour of machines using sensor data for Industry 4.0; they identified a clear need for further work to improve generalisability of the system. Susto et al.⁵ presented a new multiple classifier model for predictive maintenance along with a simulation study and benchmark dataset; the data needed to be preprocessed in order to allow for a suitable classifier such as k layer neural network (k-NN) and support vector machines (SVM), to be trained. Moldovan et al.⁶ presented different ML methods for comparison using a semiconductor manufacturing dataset and highlighted the

¹Electronic Computer Engineering, Queen’s University Belfast, UK
²Mathematical Sciences Research Centre, Queen’s University Belfast, UK

Corresponding author:
Caoimhe M. Carbery, Queen’s University Belfast, UK.
Email: ccarbery02@qub.ac.uk
benefits of reducing the data through feature selection. Lee et al. \cite{lee2018} emphasised the benefits of artificial intelligence (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. \cite{kotsiantis2006} has shown the major impact that inefficient data can have on ML models. Our previous work \cite{ourwork2020} used Bayesian Network models for modeling manufacturing data, and indicated that further research was required to develop an adaptable solution for analysing the data that could incorporate other types of models.

The readers are directed to the systematic review undertaken by Ge et al. \cite{ge2020} which demonstrated the separate aspects of ML models for the process industry. The results indicated the major areas of analysis to be fault classification, process monitoring and quality prediction. Although ML and data analytics methods have been applied to manufacturing domains, a major conclusion is that more attention should be placed on the earlier stages of data quality assessment and preprocessing to ensure the data is appropriate for the ML models. Whilst there has been recent work on the development of frameworks and toolboxes for manufacturing analysis \cite{unsupervised, supervised}, none have investigated in depth the necessity of the preprocessing stages of analysis to generate insights into the data.

**Missing data and preprocessing**

Real-world data are often subject to incompleteness, with a range of levels of missing data occurring depending on the characteristics of the data collection or extraction process. The issue is trivial when only a small amount missing; however, for larger rates of missing data, this can significantly impact the performance of any analytics models\cite{missing}. Missing data analysis is becoming an active area of research where different approaches are investigated to identify the best procedure for handling missing data with researchers exploring data visualisations for missing data\cite{missingvisualisation}.

Imputation approaches have been used extensively, with a few papers considering sensor data from manufacturing processes\cite{sensordata}. Although imputation procedures for missing data through prediction models have demonstrated its effectiveness in other domains, this approach may be inappropriate for production data due to unusual characteristics\cite{unusual}. Therefore exploratory analysis of missing data to identify the levels missing and structure within it should be considered in the development of a framework.

Preprocessing of data is critical in analytics processes to ensure that appropriate results can be generated from prediction models. If data is of low quality, the results from any model would not be reliable. The development of a generalised data preparation stage for manufacturing is essential for creating useful predictions\cite{prediction}.

**Feature selection and importance**

Feature selection is an active research area in computer science and statistics, yet it has not been a major focus in the area of manufacturing production lines. Utilising feature selection and ranking of features importance in classification models can identify previously unknown influential factors\cite{featureimportance}.

Shao et al.\cite{shao2020} presented an algorithm for feature selection in monitoring of manufacturing processes and indicated how recent developments in sensor technology increased the amount of generated data, some of which may not be relevant to a classification or prediction model. Feature selection is data-driven and should be considered as a tool to identify influential features, or generate new insights that can be presented back to engineers. In some scenarios, detailed process information may not be available and without feature selection, irrelevant features could potentially be included in models\cite{irrelevant}.

**Motivation and overview:** With increasing complexity in manufacturing processes, ML algorithms are being used for earlier detection of defects, improving production performance and prediction of future performance\cite{complexity}. In this paper, a new framework is introduced that collates, preprocesses and generates training data for manufacturing and allows behaviour to be identified that can influence the production outcome. The work described here focuses on the initial stages and demonstrates the capabilities of preprocessing analysis in generating insights to manufacturing data. The initial stages of analysis ensure that the raw data is appropriately transformed into the correct format for building ML models. A key aspect of the framework is the insights from the data that can be fed back to engineers and policy makers to assist in future decision making. It demonstrates the importance of preparing data for classification models and presents scenarios of these procedures on two case studies.

This paper extends our work presented in International Conference on Intelligent Manufacturing and Internet of Things (IMIOT2018)\cite{extend} by including the following; a new section focusing on feature selection methods and covering the theory on XGBoost; introduction of Decision Trees; a further analysis comparing models and demonstrating their suitability for the preprocessing framework for learning tree-based models; further validation of the framework’s procedures using an alternative data source.

**Manufacturing data framework**

In this research, we focus on the crucial stages of preprocessing, selection of suitable algorithms and interpretation of the results to generate a suitable method for providing feedback to manufacturing engineers. Our framework shown in Figure 1, involves four key stages to produce an appropriate learning model\cite{framework}. A key aspect of the framework is the key stages of evaluation between the statistician/data scientist and the manufacturer/engineer roles. The framework is starting to establish a sequence of steps and a series of checks which if achieved, should act to ensure a better quality of data is generated for the model build.

**Raw Data**

The first stage is for the manufacturer/engineer to prepare the data for the statistician. Raw, real-world data is unstructured and inconsistent in nature, often involving large dimensions, class imbalance issues and missing instances. Therefore its collation is often challenging and involves combining data

---

**Prepared using sagej.cls**
Carbery et al.

Figure 1. Data Analytics framework for analysing manufacturing data through an iterative process between statistician and manufacturing staff/policy makers.

from different sources e.g. from sensors of varying machines etc. and processing it appropriately. In addition, there can be issues of security and treatments may be needed to anonymise the data and ensure its secure transmission; of course, the data can be of the order of several orders of TeraBits.

**Exploratory Data Analysis and preprocessing**

Exploratory data analysis (EDA) and preprocessing are crucial stages in preparing data for AI algorithms. 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. As preprocessing can have a critical impact on model performance, effort has been targeted on standard approaches e.g. filtering and normalising to ensure that the training dataset is of an appropriate format whilst checking that no bias has been introduced. There are a number of aspects to be considered:

- Class imbalance
- Incomplete data
- Data dimensionality
- Feature distributions

There is a key decision to be made on the preprocessing procedures and also on feature selection algorithm for dimensionality reduction. 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. 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;

For our analysis, there is too much missingness to merely discard these instances, so the chosen approach, as demonstrated by other researchers, 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.
Feature selection

To improve model performance and reliability, two feature selection approaches namely wrapper and embedded methods, are performed to reduce the number of variables by identifying the most influential and important features. 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.9

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.15. 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.15. 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.

Model Build

The next block of the framework concerns ML model building based on the features determined in the previous step. This stage considers the analysis of features within manufacturing datasets through utilising ML models for either prediction or classification purposes. The analysis goal is determined through collaboration with manufacturing engineers on their target goal, and through statisticians identifying what is possible and plausible. This framework considers two algorithms for model build; XGBoost and Random Forest.

XGBoost 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.19. 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 which is determined in earlier stages of EDA. Following feature selection, the learning algorithm can be implemented on these important variables with multiple iterations to generate a powerful classification model.

Random Forest: Random Forests (RFs) are ensemble models that are composed of decision trees (DTs).6,20 Decision trees (DTs) are a branch of models based on statistical theory that provides the mechanism to uncover structure within the data and demonstrate the underlying process in a tree format. They use a training sample of data to produce a structure which constitutes a set rules that can be used to make decisions on new samples from the test data. DTs follow a hierarchical structure that is easily interpreted; they consist of nodes and directed edges that extend from the root node, built through iterative algorithms that assess the data and searches to determine at which value of a feature best splits the data into two groups with the underlying algorithm ensuring that the split minimises the impurity of the subsets. This means that the split groups are homogeneous and do not have any bias towards one split over the other.

A RFs algorithm constructs large numbers of decision trees and use random subsets of the data to reduce potential cases of over-fitting. Each tree represents a small component of the variation in the data. By generating thousands and connecting them within the random forest, they can be used to capture the complex interactions within complicated data. Each tree within the RFs aids in the prediction of the classification by capturing combinations and interactions amongst subsets of features.

We consider the inbuilt feature importance generated from building RFs models. RFs produces an importance metric based on Mean Decrease Gini Index based on the Gini impurity metric. Gini impurity is a metric that is representative of how often a randomly chosen element in the dataset used in training the model will be incorrectly labeled if it was labeled according to distribution within the subset. In essence, the measure can be considered a probability of a new instance being incorrectly classified at each node in the DT based on training data.

As noted above, RFs are a collection/ensemble of DTs and therefore we must leverage the Gini Index to calculate the mean decrease of impurity in the model. They are based on the DT theory and can be considered as a way to uncover important features by calculating the Gini index. For tree-based models that involve splitting into subgroups, it is important to ensure that the splits are homogenous, to monitor this we must look at the impurity measures. One way is to look at the Gini index as it measures impurity between different classes within a group. A lower Gini index indicates a higher purity.20

Model validation and performance

An important component of model validation is to ensure that cross-validation (CV) is reduced. This is achieved by running all of the modeling processes on different subsets of the data to generate multiple measures of model performance. From these outputs, we can produce a more accurate representation of model quality. The most commonly used CV is K-fold cross validation where K must be specified. However, it is important to consider the size of the data and complexity of the algorithm to be implemented as larger number of folds can result in intensive computations.

In our case, we consider K=6 and thus split the data randomly into 6 subsets that contain the same number
of samples. Within the cross validation all processing and model building is performed and repeated 6 times. Throughout this process, one fold is kept for testing and the remaining folds are used as the training set. Within CV and alongside model building, we consider mostly how to evaluate the suitability of a classification model.

Classifiers can be evaluated using standard statistical metrics such as accuracy, sensitivity, specificity, precision and F-measure. We utilised our model and testing dataset to produce values to assess the suitability of our chosen classifier and 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. 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. General format of confusion matrix.

<table>
<thead>
<tr>
<th>Predictive</th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(tp)</td>
<td>(fn)</td>
<td>(tp+fn)</td>
</tr>
<tr>
<td></td>
<td>(fp)</td>
<td>(tn)</td>
<td>(fn+tn)</td>
</tr>
<tr>
<td></td>
<td>(tp+fn)</td>
<td>(fp+tn)</td>
<td>(N)</td>
</tr>
</tbody>
</table>

The measures are highlighted as follows:

- **Accuracy**: indicating overall effectiveness of a classifier, calculated using the formula \(\frac{tp+tn}{N}\) 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 \(\text{Sensitivity} = \frac{tp}{tp+fn}\) and \(\text{Specificity} = \frac{tp}{tp+fp}\);
- **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 \(2 \cdot \frac{\text{Precision} \cdot \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}\) and is more robust to imbalanced data.

The domain expert, namely manufacturing staff, can also have an input in the model governance through their assessment of the model outputs and determining its suitability to provide insights to policy makers for future decision making. This aspect can influence further analysis and adaptations of the model.

### Case studies

To test the effectiveness of the framework, the approach has been applied to two widely available datasets, namely the Bosch and SECOM manufacturing datasets which have been widely explored in the literature.

### Bosch manufacturing case study

Bosch has 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. 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. For example, \(L1S24_F1695\) indicates that Feature 1695 was observed at Station 24 on Line 1. However by investigating the correlated stations, it is possible to work out some aspects of connectivity as shown in Figure 2.

The datasets is split into three categories; date, categorical and numeric. Within each of these groups, Bosch has provided the data separated for training and testing thus avoiding in this case, the third stage in Figure 1. The training sets that 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 2 depicts an example of the flow of a product across the factory floor, highlighting the numerous stations associated with different lines in the build.

### SECOM dataset application

SECOM is an open source industrial dataset which provides data from a complex manufacturing process within the semiconductor industry. The dataset includes 590 of manufacturing operation measurement features and one response variable which classifies the product as passing or failing a quality test. Electrical testing is used to segregate the ‘good’ products from the faulty devices. SECOM has 1,567 examples which have the associated anonymised features which are consistent with sensor measurements from the process. This data has the following similar characteristics to Bosch data;

- Class imbalance with a ratio of 1:14 between number of failures to passed products.
- As some recorded features may not be useful for classification, feature selection is useful.
- Approximately 6% of the data is missing in SECOM. The knowledge behind this missingness is unknown, as in the Bosch data.

### 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.

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

<table>
<thead>
<tr>
<th>Data Characteristics</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>986</td>
</tr>
<tr>
<td>Rows</td>
<td>1183747</td>
</tr>
<tr>
<td>Lines</td>
<td>4</td>
</tr>
<tr>
<td>Stations</td>
<td>51</td>
</tr>
<tr>
<td>Percentage missing</td>
<td>78.5</td>
</tr>
<tr>
<td>Percentage fail</td>
<td>0.58</td>
</tr>
</tbody>
</table>
Figure 2. Example flow of Bosch factory floor depicting stations as circular nodes representing the complexity and interactions across stations within the manufacturing production.

Figure 3. Bar plot showing the number of features associated with the individual stations of the Bosch production line.

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 elsewhere. 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. 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.

Figure 3 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.

Preprocessing

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 157 input
features along with the target response. Whilst this reduces the dimensionality of the dataset, the relation of the features with the response variable can also be investigated. Figure 4 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 initially remove any feature with over 50% missing as any imputation could potentially bias future analysis; then we assigned the missing instances a unique value which could then be grouped into an independent category for any discretisation. As a result, our dataset was reduced to 142 features and 1,102,782 observations.

**Feature selection**

This allows selection of the 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. In addition, Table 4 shows an example of the Gini Index values produced from our RFs model based on the factored training data. This demonstrates the impact of these features based on the levels and counts within them.

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1_S24_F1723</td>
<td>0.5328070</td>
</tr>
<tr>
<td>L1_S24_F1846</td>
<td>0.2248599</td>
</tr>
<tr>
<td>L1_S24_F1632</td>
<td>0.1162531</td>
</tr>
<tr>
<td>L1_S24_F1695</td>
<td>0.0611954</td>
</tr>
<tr>
<td>L3_S33_F3876</td>
<td>0.0403588</td>
</tr>
<tr>
<td>L2_S26_F3036</td>
<td>0.0132253</td>
</tr>
</tbody>
</table>

**Table 4.** Example of six variables from Random Forest based on Gini Index showing the impurity measure for the features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mean Decrease Gini</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3_S33_F3857</td>
<td>17214.795</td>
</tr>
<tr>
<td>L3_S33_F3865</td>
<td>7571.506</td>
</tr>
<tr>
<td>L3_S33_F3863</td>
<td>17949.585</td>
</tr>
<tr>
<td>L3_S33_F3861</td>
<td>16772.033</td>
</tr>
<tr>
<td>L3_S33_F3855</td>
<td>16112.618</td>
</tr>
<tr>
<td>L3_S33_F3859</td>
<td>15879.557</td>
</tr>
</tbody>
</table>

**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. Random Forests (RFs) require additional processing to account for class imbalance, however, the algorithm has the capability to include priors to learn the model based on the distributions of the class levels. An implementation of Synthetic Minority Over-sampling Technique (SMOTE) was performed to rebalance the levels to ensure there is a sufficient amount of Fail outcomes for the algorithm to learn from.

**Comparison of RF and XGBoost**

An initial study was constructed to assess a DT model for the numeric data. However, due to the features being numeric, this cause us to face issues computationally as the algorithm attempted to iteratively assess each potential split. As a result of this, we were faced with a model that contained only a root as the tree could not grow. Upon
further investigation into the data itself and by plotting the distribution of the features, we could see that there was an opportunity to bin the features into factors.

As the data is fully anonymised with no opportunity for input from experts, we utilised a supervised learning algorithm that bins the features with respect to the response variable. The chosen approach was based on Weight of Evidence (WoE) which computes an associated Information Value for each bin proposed.\textsuperscript{24,25} Figure 6 depicts the top 6 features based on their IV which indicates the important variables for a predictive model. From this analysis, we were able to perform feature selection for the RF model as any feature with IV < 0.02 is deemed pointless for prediction\textsuperscript{24,25}

![Figure 6. Information Value produced by WoE binning of features to generate suitable factor levels for the top 6 features.](sagej.cls)

As a result of WoE binning, our training data consisted of 55 important predictor variables for the classification task. This helped reduce the computational time of the model and also reduce the errors faced when working with such a large dataset.

For the initial XGBoost model, we considered the numeric levels within the features. However, to allow for a comparison between models to be performed with RF, we had to use the same binned feature set. One condition of XGBoost algorithm is that it only deals with numeric values. Thus we implemented one-hot encoding to generate a suitable dataset based on the factored features used in RF. This approach has been used in a number of research papers to allow for comparisons to be undertaken.

An updated XGBoost model was produced and Table 5 presents the overview of metrics that have been calculated for RF and XGBoost. From this, it is clear that both models have the same predictive power, with XGBoost only improving slightly on some measures. However, we determine that the interpretability of the RF is intuitive and if performs at the same level as XGBoost then we can consider this choice.

![Table 5. Performance metrics produced for Random Forest and XGBoost classifiers on the reduced factor feature set.](sagej.cls)

Table 5. Performance metrics produced for Random Forest and XGBoost classifiers on the reduced factor feature set.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Random Forest</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.998</td>
<td>0.9981</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.9791</td>
<td>0.9792</td>
</tr>
<tr>
<td>Precision</td>
<td>0.9979</td>
<td>0.9980</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.9989</td>
<td>0.9989</td>
</tr>
<tr>
<td>Kappa</td>
<td>0.9883</td>
<td>0.9884</td>
</tr>
</tbody>
</table>

Figure 7 presents the plot of the complexity parameter of a decision tree within the random forest model to determine the error with increasing tree size. Figure 8 presents a visual graph of a decision tree within the RF following pruning. The splits are based on the binned levels from features and have identified the most influential features on the products passing or failing. These features are also identified through XGBoost model and depicted in Figure 9.

**SECOM Data Analysis**

The SECOM dataset has been used extensively as a benchmark for assessing ML models for the area of manufacturing\textsuperscript{26–29}. The motivation of analysing this data alongside Bosch, is to demonstrate a further application of our analytics procedure in highlighting influential features that could be presented to policy makers in a real world application. Analysis of the products and characteristics of the data provide feedback to the manufacturing staff on the design process by clarifying or identifying failure causes which can prevent recurrence in future builds and improve product quality.

From the EDA phase, a visualisation of the missingness present within SECOM dataset can be created (Figure 10). Figure 11 shows the top five important features identified by XGBoost, which is considered as part of our feature selection stage of analysis. From this, we can identify one feature which has a major influence on the outcome of the products in SECOM, namely Feature592. The other features, although identified as having some importance on the outcome are at a much smaller level than the others.

Utilising the RF and DT models on SECOM, it was identified that Feature592 was the feature impacting the outcome of the products substantially and following a pruned tree, was the only remaining feature. This is consistent with the models built for SECOM throughout the literature\textsuperscript{29}.

**Discussion**

In this paper, a new framework has been introduced 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 142 features was produced, allowing an extreme gradient boosting tree to be used as a classification prediction model. Using the framework (Figure 1), 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\textsuperscript{19} and employ R’s inbuilt performance metrics to demonstrate the performance of this approach in comparison to RF. Our study of RF also demonstrates how preprocessing procedures change depending on the chosen prediction model and RF, although notably require less processing than other approaches, face some issues computationally for features with large number of factor levels. Therefore, extra processing is necessary to reduce these levels, in our case we implementing WoE binning and produced a model that had an accuracy of 0.998 and was robust to misclassifications, as demonstrated by its sensitivity and specificity levels. To demonstrate the validity of this framework, an application to a different
Figure 7. Plot of the complexity parameter (cp) of a decision tree within the random forest to determine the relative error with increasing tree size.

Figure 8. Sample decision tree from within the RF with pruning performed based on complexity parameter from Figure 7 to determine the optimal tree size. The splits are based on the binned levels from the features and the leaf nodes are counts from the sample for whether a plot failed or passed.

manufacturing dataset was conducted. The SECOM dataset underwent the analytics process of the framework to generate models and insights.

Through pre processing and analysis of the models, we concluded that there is no possibility of merely using the raw data as input into the algorithms and that there is a requirement for performing the necessary processing of the data to ensure a useful and powerful model is produced. The main aspects of the processing were removing redundant data which helped to reduce the data size and also the identification of highly correlated features may introduce bias. The procedures of RF and XGBoost, although based on DT, vary slightly and therefore, some of the results are not consistent. Further research into the algorithms could help to understand which is the best choice for different scenarios, however this research wanted to focus on the capabilities of the models for predicting and identifying critical features within an anonymised dataset and to demonstrate the necessity for appropriate preprocessing of data.

Conclusion
A new data analytics framework for predicting faults in a large-scale manufacturing system has been introduced and
applied to two manufacturing datasets, namely Bosch and SECOM. To the best of the authors’ knowledge, this is the first framework that has been implemented on two of the most widely used benchmark manufacturing datasets. Its key advantage is that it has generated insights into data without prior knowledge of the features and provided insights of the unknown data in identifying the most influential features for building a classification model. The framework is also being assessed on data provided by a commercial partner to assist in developing an automated system for performing analysis on manufacturing data.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article. This work is supported by the Engineering and Physical Sciences Research Centre (EPSRC).

**References**


