A Bayesian network based learning system for modelling faults in large-scale manufacturing

Caoimhe M. Carbery*[1,2], Roger Woods[1] and Adele H. Marshall[2]
[1]Electrical and Computer Engineering (ECE), [2]Mathematical Sciences Research Centre
Queen's University Belfast, Northern Ireland
*Email: ccarbery02, r.woods, a.h.marshall@qub.ac.uk

Abstract—**Manufacturing companies can benefit from the early prediction and detection of failures to improve their product yield and reduce system faults through advanced data analytics. Whilst an abundance of data on their processing systems exist, they face difficulties in using it to gain insights to improve their systems. Bayesian networks (BNs) are considered here for diagnosing and predicting faults in a large manufacturing dataset from Bosch. Whilst BN structure learning has been performed traditionally on smaller sized data, this work demonstrates the ability to learn an appropriate BN structure for a large dataset with little information on the variables, for the first time. This paper also demonstrates a new framework for creating an appropriate probabilistic model for the Bosch dataset through the selection of statistically important variables on the response; this is then used to create a BN network which can be used to answer probabilistic queries and classify products based on changes in the sensor values in the production process.**

I. Introduction

Manufacturing companies are investigating methods to step into *industry* 4.0 as the demand for advances to improve their product yield is continuously increasing [1]. Companies are able to gather vital process data from various stages of their production line throughout the build. Using abundant levels of available data, it should be possible to use advanced data analytics to allow companies to process it and explain uncertainties, thus permitting them to make more intuitive and informed decisions for future product builds.

Recent studies [2], [3], [8], [9] have investigated machine learning methods for modelling manufacturing domains with particular emphasis on predictive maintenance and fault detection. Rare event predictions and creation of fault prognostic systems using alternative methods have been covered in [9], [10]. Wang *et al.* has demonstrated the usage of BNs for modelling fault detection in a semiconductor manufacturing process [9], where both discrete and continuous variables representing relationships within a manufacturing dataset were deemed to be highly correlated. Using knowledge from experts, six variables were identified to monitor the state of the process under investigation. Whilst this demonstrated the potential of BNs in providing expert knowledge to create the structure, the model was only accurate when adequate training data containing potential abnormalities was used.

Lee *et al.* [4] proposed a convolutional neural network (CNN) to account for multivariate sensor signals across time points to extract faults in semiconductor manufacturing. The popular gradient descent methods were used during the training phase, allowing the authors to show that the new model had a shorter training time compared to the standard model.

Our research advances research by using a large-scale application and considering the impact of high amounts of data. Previous studies have performed analysis on these models alongside others to determine performance measures and ability to appropriately model the domain. However, these often do not consider the interpretability of models for this particular domain. The research proposed in this paper not only demonstrates the ability of learning a model of this scale, but also considers the benefit of human interpretation from the visual aspect of a BN model.

For our research, we investigate a large-scale manufacturing system from Bosch which is hosted by Kaggle [6]. We address the data size issue and create models to aid in predictions that also provide clear insight into the connections on the factory floor that may not be previously known. By interpreting the data and building accurate models, we aim to aid the company in making valuable changes to their system to prevent future issues. The chosen model has not yet been implemented on this dataset, thus providing an opportunity to produce and uncover further information about this particular system.

This paper presents a new framework for fault classification and prediction through multiple stages of pre-processing, feature selection, combining factor levels, BN learning and inference (see Fig.1). The aim is to provide a mechanism to allow for a probabilistic model to learn from a large amount of data and to provide a clearer insights. The model is then used as a tool for extracting relationships between variables providing insights into the uncertainties of Bosch's production line faults as a prediction task. Creating an appropriate model to represent the system is beneficial for future analysis of predicting rare faults as it aims to demonstrate an ability to find complex multivariate, non-linear relationships between variables and highlight any causal relationships.

The contributions of this paper are the following:
- an approach to model a large-scale manufacturing system which addresses the data size complexity,
- a new algorithm implementation to the Bosch dataset,
- a novel framework for fault classification and prediction for large datasets,
- a reproducible approach for building a probabilistic model.
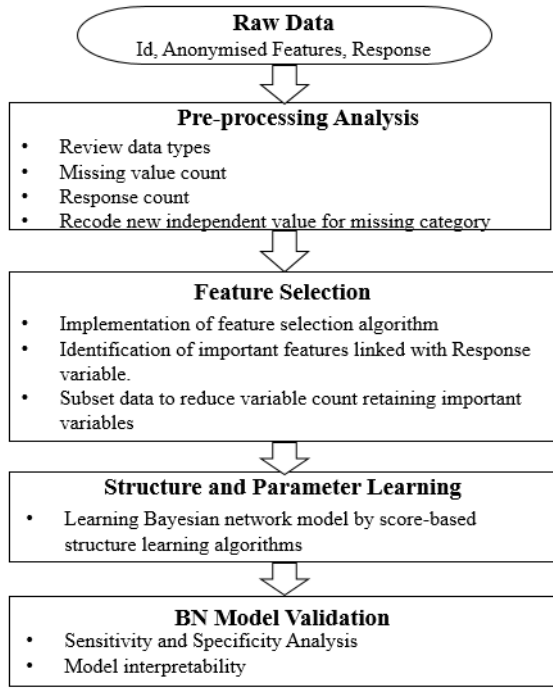
Fig. 1. Design flow showing the preprocessing stages to create the appropriate processed data for input to learn BN model.

The paper is organised as follows. Section II covers the background methodology of BNs, model validation and performance measures used. Section IV introduces the Bosch manufacturing case study and the important characteristics of the data. Section V presents the data exploration and pre-processing of the Bosch data. A BN model for this dataset is also presented in this section, highlighting the density and complexity of the resulting network. Section VI provides the results and interpretations of our findings from the BN model.

II. BACKGROUND

When there is a large amount of data, we must take into account computational issues associated with the complexity of model learning. Pre-processing in the form of feature selection can be performed on high dimensional data to improve learning procedures through reducing both complexity by discarding irrelevant variables. Fig. 1 presents the stages of organising and manipulating the data appropriately for the models.

A. Feature Selection and Discretization

After the initial pre-processing stage, we consider feature selection to identify the most influential variables. We utilise two approaches for feature selection and feature reduction; *information gain* and the Boruta algorithm which is based on random forest algorithms to determine all relevant features/variables for the dataset [17]. These two approaches are used to ensure that the appropriate variables are retained in the data. The variables selected must encode as much information

as possible so that the final classifier can obtain higher performance in classifying groups correctly. The construction of any model is dependent on the variables selected for it, as models often have a reliance on the assumption that all variables in the model are key indicators of faults in the system. Whilst smaller numbers of variables improve both the performance and accuracy, we want to ensure that a key feature is not removed.

B. Bayesian Network

BNs are a popular choice as they can handle uncertainties and provide the means to decompose complex problems into simpler ones through their use of conditional probabilities [9]. The structure of a BN is in the form of a directed acyclic graph (DAG) which is the qualitative aspect of the model; it provides insight into the relationships between random variables through arcs connecting them. Fig.2 depicts an example of a DAG where random variables are represented by nodes in the BN, and the connections between them represent dependencies. The quantitative part of the BN consists of the conditional probabilities associated with the random variables, representing the relationship between nodes and their predecessors through probability tables.

Consider a set of random variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ representative of the system and $1 \leq i \leq n$. If there is an arrow, defined as an arc, going from node X_1 to X_2, then X_1 is said to be the parent of X_2, and X_2 is the child. If we say the set of parent nodes of variable X_i is defined as *parents*(X_i), then the BN can be specified by the following

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i | parents(X_i)). \tag{1}$$

For a manufacturing process, we consider steps involved in the product build. Within each step there are a number of sensors to gather information relating to the status of the build. The goal is to use the data collected by the sensors at each time point to predict whether there is a fault or there will be a fault at a future time point [9]. Fig. 2 illustrates a BN with three nodes, F_1, F_2 and S. Fault variables, F_1 and F_2 represent two potential faults which may be present or not. Node S represents a particular sensor in a machine. From Fig. 2, it is clear to see that F_2 is dependent on whether there is a fault in F_1. The BN is intuitive as we can see that a sensor value in S is influenced by both F_1 and F_2. From this BN, we may consider the dependencies between variables to correspond to causal relationships, as it is logical reasoning to suggest that underlying faults in a system will have an impact on sensor values [12]. Fig. 2 also highlights the probabilities and conditional probabilities associated with each node, highlighting that they are dependent on their parent variables as presented in Eqn (1).

BNs are used as supervised classifiers and are also a highly reliable model for uncovering relationships amongst variables in a system as well as their usage in areas of predictive analytics [5]. The main usage of a Bayesian classifier is to apply Bayes' theorem of equation (2) as it allows the
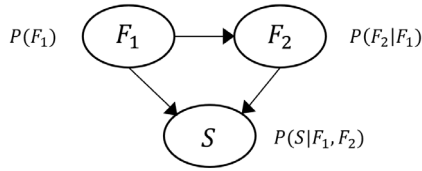
Fig. 2. A BN example for fault diagnosis in a manufacturing system.

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

probabilities of the model to be updated when new data is fed into the BN and Bayes' theorem is said to efficiently propagate information through the network.

$$P(\mathbf{X}|E) = \frac{P(E|\mathbf{X})P(\mathbf{X})}{P(E)} = \frac{P(E,\mathbf{X})}{\sum_{\mathbf{X}} P(E,\mathbf{X})} \qquad (2)$$

1) Learning Bayesian Networks: Learning a BN model involves two key stages; structure learning and parameter learning. Fig. 1 illustrates the flow of the BN learning process after the preprocessing stage.

Cognitively, it is impossible to gather so many numbers from a human expert [5], but if we wish to accurately learn the distribution from the data, we would need a large amount of data. In cases where the information is unavailable, structure learning must be used to search through the data to formulate the network structure based on the connections uncovered from the dataset. It can be computationally expensive and generally too large to store in memory.

The next stage after defining structure is parameter learning to determine conditional probability tables (CPTs). Parameters contain information on the prior probability of nodes which are calculated before gathering new evidence or information for the model. CPTs can be generated through parameter learning using historical data, as in structure learning. This approach has an advantage of being a more accurate BN parameter model, however, a disadvantage is that it can be difficult to collect an adequate amount of data. This is not an issue in a large modelling application of a manufacturing company.

Inference for BNs is the procedure of finding the outcome probability of a variable using previous values as evidence. Inference algorithms fall under two categories; exact and approximate. Approximate inference algorithms are the popular choice with larger datasets.

C. Performance metrics and validation methods

The performance of a classification model can be evaluated through the use of various measures. Table I depicts a confusion matrix which consists of four counts; 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 [16].

Confusion matrices provide values that are used for calculating various performance measures for classification, and

in the case of this paper, binary classification. Accuracy is a measure that indicates the overall effectiveness of a classifier which is calculated using the formulae $Accuracy = \frac{tp+tn}{N}$. Sensitivity and specificity analysis provide values to evaluate the effectiveness of the classifier to identify positive labels and negative labels respectively. The equations for calculating these 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 $Precision = \frac{tp}{tp+fp}$. The F-measure is another measure which is calculated by $F-measure = 2 \cdot \frac{precision \cdot sensitivity}{precision+sensitivity}$.

III. EXPERIMENTAL IMPLEMENTATIONS

Experiments were carried out using a well-known binary class benchmark dataset available from the UCI machine learning repository [7] that has been frequently used for evaluation purposes of algorithms. Prior to implementing algorithms on to previously unseen data, we tested our approach on a commonly used dataset to verify the chosen methodology and algorithms. The data is from a different application domain, but is appropriate for validation reasons prior to the Bosch implementation.

Data	Observations	Variables	Classes
Adult	32561	13	2

Our experimental studies followed the process depicted in Fig. 1 whereby initial pre-processing was required before the BN as described in Section II-B. Fig. 1 demonstrated the flow process of pre-processing stages and the subsequent learning stages of the models. This process was used on this dataset to validate the steps. The network and associated parameters were compared to the literature to assess the learning algorithms chosen for this research.

IV. MANUFACTURING CASE STUDY

Bosch provided a large dataset (14.3 Gb) from one of their factories' production lines with an aim of generating interest in the creation of advanced methods for predictive modelling [6]. The dataset consists entirely of a large number of anonymised measurements from the production line, where jobs and products moved throughout various lines and different stations where each part has a unique Id. The goal is to predict which components will fail at the final stage of the production

line (i.e. *Response*=1). Our research addresses this aim through the implementation of a BN model to compare classifiers. Three different types of data are available: numeric, categorical and time based data which are described as being both scaled and anonymised. The categorical data is extremely sparse and is excluded for this research as it was for the research in [10], [14]. Initially, this research focuses only on the numerical data as preliminary data exploration found it to be the most influential, therefore the time/date variables were not within the scope of our research for this paper. A summary of the Bosch production line numeric data used for our research is provided in Table III.

TABLE III
SUMMARY INFORMATION FOR BOSCH PRODUCTION-LINE DATASET.

Characteristics of dataset	Value
Variable count	986
Observation count	1183747
Station count	51
Percentage of product fails	0.58%
Percentage of missing observation	78.5%

A number of unique characteristics are associated with the Bosch data and need to be considered for statistical analysis and learning an appropriate model.

1) Anonymised data: no access to expert knowledge to indicate the importance of variables and therefore algorithms must be used to uncover them.
2) Missing data: observations are sparse as parts do not pass through all stations of the process.
3) Large-scale data: the data is extremely large and therefore we require a scalable algorithm for Big Data.
4) Class imbalance: the data is highly imbalanced as the class of failed parts is small in comparison to those who passed quality control.

V. DATA EXPLORATION AND PRE-PROCESSING

Variables are labelled under groups associated with the manufacturing line and station with which the particular feature collected values. For example, the variable $L2\_S24\_F1565$ represents feature number 1565 which could represent a sensor situated at station 24 on line 2. The variable *Id* represents an independent part that is going through the process build. These individual parts may be processed at the same time in the same station, but may not follow the same path throughout the manufacturing build. Difficulties arise with extremely large datasets, therefore we perform feature selection methods to identify important variables.

Fig. 3 shows the count of numeric variables associated with each station. Stations 24 and 25 have the highest number of features associated with them i.e. they are stations where most measurements took place and therefore more products were processed.
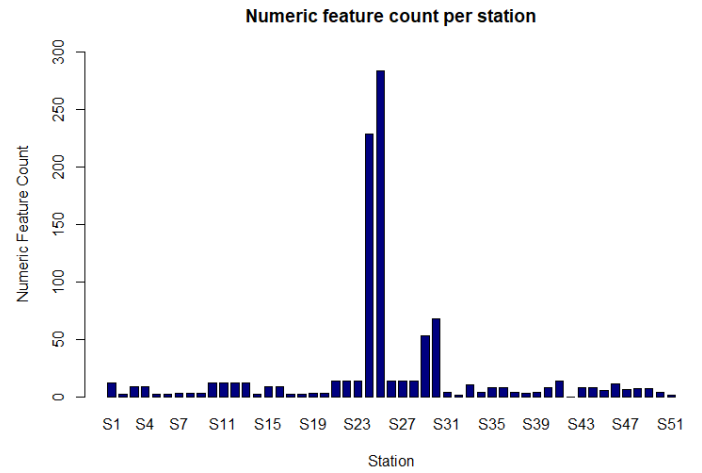

Fig. 3. Plot depicting the number of feature variables associated with each station in the Bosch production line.

A. Feature Selection and Variable analysis of Bosch data for BN

The algorithm indicated that there were fifty variables deemed to be important for classification models. Examples of the top six variables are presented in Table IV. The algorithm reviews possible combinations of variables to build a predictor model through evaluating the importance of variables based on a calculated score.

TABLE IV
EXAMPLE OF SIX VARIABLES CONFIRMED FROM FEATURE SELECTION ALGORITHM.

Feature	Mean Importance
L3_S33_F3857	9.2554
L3_S33_F3865	8.5671
L1_S24_F1846	8.3445
L3_S29_F3407	7.4475
L3_S29_F3412	7.4185
L3_S33_F3855	7.3758

Following the feature selection process one must consider transforming variables into a discrete form for learning the BN with a class outcome. Upon reviewing the data, it is found that each variable is not necessarily continuous, but is a factor variable with extremely large number of levels. For example, variable $L0\_S0\_F20$ contains 997 levels. This adds another difficulty associated with the learning procedures. Further data cleaning and investigation could be performed to review and combine the levels of each variable, into a smaller number of relevant factor levels. This can be done using clustering techniques, however, this is not the focus of this paper as we are focused on the main aspects of BN build rather than improving computational complexity. Factor variables are required for building BN model due to limitations with a categorical/binary outcome variable.

B. Bayesian Network model

Using the factor versions of the subset of features, we consider these as the input variables into the BN model build. By considering the variables as discrete, the BNs conditional probability distributions are defined by conditional probability tables, which can improve the efficiency of inference and learning in the BN model [15]. As the data provided by Bosch provides no prior information regarding the structure of the network and the direct or indirect relationships between sensors, structure learning is essential as a first stage in the model build. An R realisation is utilised to compute dependencies between the important features to determine the relationships present within the system. A score-based structure learning algorithm, based on BIC score is implemented to produce the direct connections between the variables to generate the appropriate structure of the Bosch BN. This provides clearer understanding to the user of the connections and influences of the factors on the response variable.

As this dataset is anonymised, the structure does not indicate the type of sensors and values influencing the response. The premise of this design would ensure that on un-anonymised data, however, the structure would provide distinct connections and further information, not be observed by domain experts.

Fig. 4 presents the structure learned from the data through the score-based learning algorithms. There were 50 variables found to be relevant to the data with regards to the response, however through the structure learning algorithms, there were some variables i.e. sensors which had no relevance to the probabilistic outcome of the variable. From this structure, it can be seen that there are a large number of relationships and connections amongst variables. One can conclude that this model has grasped complex connections from the large dataset.

This BN structure was then used to assist in the learning of the parameters of the BN model through the construction of conditional probability tables. In case studies where information is provided by the model, one could intuitively interpret the connections. From our understanding of the BN structure, the later stage sensors of the process, i.e. those in Lanes 2 and 3, have connections to the Response variable. Therefore, it could be said that these sensors have more of an insight into the outcome. Monitoring the changes in these variables will provide engineers with a focus on key sensors to allow for the early detection of changes which could cause a failed product further down the line.

TABLE V
BOSCH PERFORMANCE MEASURE RESULTS

Performance Measure	BN
Accuracy	0.9358
Sensitivity	0.9395
Specificity	0.3035
Precision	0.9957
F-measure	0.9668

Table V provides us with the performance measures from our BN model. A common measure seen in other implementations of this dataset is that of Matthew's Correlation Coefficient (MCC), which gives a final numeric value representing the overall fit of the model. In this research, we have calculated sensitivity, specificity, and precision for the model, which highlight the individual classification performance. These measures can then be combined to produce an accuracy and F-score for the model which gives an overall indication of model performance. These are those commonly used in evaluating the performance of classifiers in machine learning [16]. The model performs well for most of the measures. The accuracy levels, alongside sensitivity and specificity analysis demonstrate that the model is appropriate for this type of task and performs well for predicting the classes where an item will be defective or not. The purpose of calculating these performance measures, is to determine if our BN model performs well enough for the task at hand. From these results we can conclude that the model is an appropriate representation of the manufacturing domain.

VI. CONCLUSIONS AND FUTURE WORK

The Bosch dataset is extremely sparse with a large number of missing values (due to the items not passing through the station) and a highly imbalanced response class. These issues were addressed through our new framework by assigning the *missing* values with an independent value to keep these instances distinct from other observations. The class imbalance issue was accounted for in our learning procedure. The large number of features meant that we were required to perform feature selection procedures to reduce the dataset size into one that included only the statistically important variables that enabled us to utilise the BN theory to learn an appropriate probabilistic model. The BN model produced indicated that there were 38 arcs present in the structure of the data connecting different variables. This allows for further work to investigate the strength of these connections, but this is not the focus of our research.

The BN model was verified through hold-out validation method and tested on its performance using numerous measures. Statistical tests were carried out to validate it and its representation of the Bosch manufacturing domain, meaning it could be deployed on the production line which would have the ability to provide feedback and information to the engineers throughout the production process. As data is fed through the model, the parameters are updated to produce probabilities of the products failure. A BN model can be used to infer solutions to probabilistic queries, for example, manufacturers and engineers may *ask* a query about a particular sensor and its impact on product yield, to provide solutions or predictions. This type of inference may only be done once the model is fully specified.

Issues associated with BN include the computational learning aspects due to the large amount of data. Pre-processing and feature selection was used to produce an appropriate subset of variables that had an associated importance to the
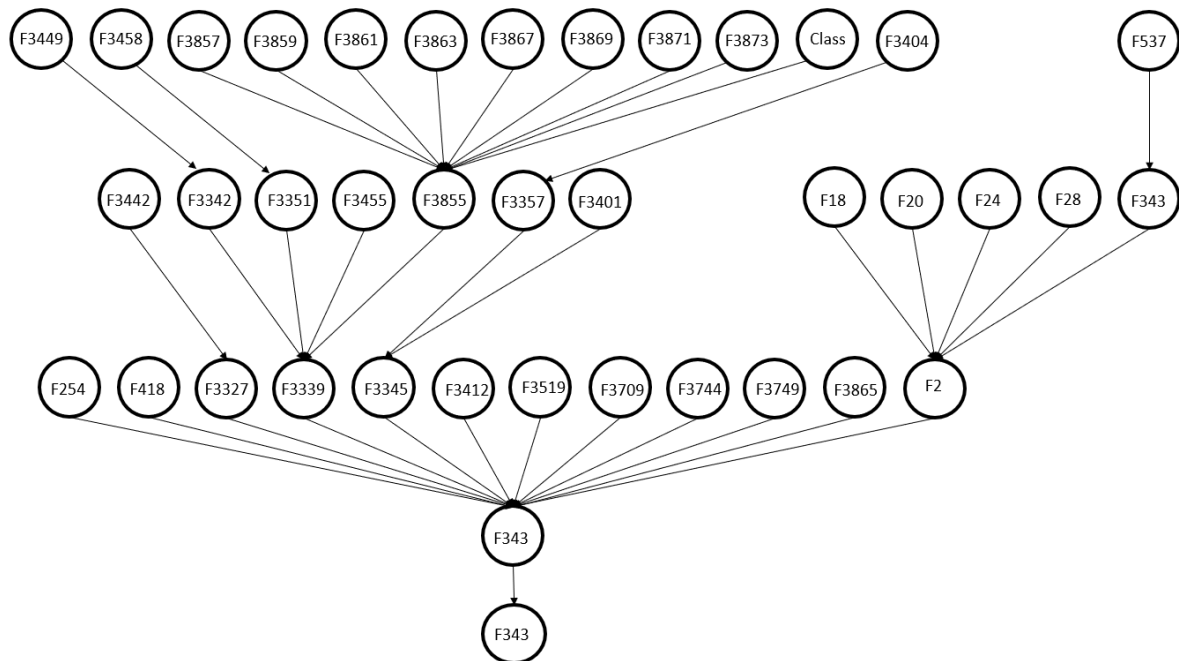
Fig. 4. Structure of the BN model produced by structure learning algorithm in R.

model and outcome. Another issue is the lack of interpretation available from the structure due to the anonymised data. It is important, however, to note that this design flow of building a BN model for data of this type could be replicated on non-anonymous data which would provide crucial information on another system. The purpose of the comparison and model builds for Bosch, was to generate test bench models for use in future analysis in another manufacturing domain which is currently being done.

Another aspect of future development is the computational speed of the model build, as we need to consider the larger datasets and the means available to process this data through a complex model building process. The next stage will be to consider dynamically changing environments using data from various time points, to ensure a more realistic model for a working manufacturing company.

BNs offer a means of being easily understood to those not from a machine learning background. BNs are notably intuitive in their design and also allow users to update the models and increase the performance with further data. The model presented here is focused on the ability to learn a network of this scale with little information of the dataset and the domain.

REFERENCES

[1] J. Lee, H. Kao, S. Yang, "Service innovation and and smart analytics for industry 4.0 and big data environment," *Procedia Cirp*, vol. 16, pp. 3-8, 2014.

[2] G. Susto, et al., "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11(3), pp. 812-820, 2015.

[3] T. Wuest, C. Irgens, K-D, Thoben, "An approach to monitoring quality in manufacturing using supervised machine learning on produce state data", *Journal of Intelligent Manufacturing*, vol. 25(5), pp. 1167-1180, 2014.

[4] K. Lee, S. Cheon, C. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes", *IEEE Transactions on Semiconductor Manufacturing*, vol. 30(2), pp. 135-142, 2017.

[5] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, 2nd ed. Morgan Kaufmann, 2014.

[6] Kaggle.com (2016) *Bosch production line performance* [Online]. Accessed: Jul. 2017. Available: https://www.kaggle.com/c/bosch-production-line-performance

[7] UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, Accessed: Oct. 2017. Available: http://archive.ics.uci.edu/ml/index.php

[8] B. Kroll, et al., "System modelling based on machine learning for anomaly detection and predictive maintenance in industrial plants", *IEEE Conf. on Emerging Technology and Factory Automation*, Barcelona, Spain, 12 Jan. 2015, pp. 1-7.

[9] G. Wang, et al., "A novel Bayesian network-based fault prognostic method for semiconductor manufacturing process," in *IEEE International Conference on Industrial Technology*, 2017, pp. 1450-1454.

[10] J. Herbert, "Predicting rare failure events using classification tress on large scale manufacturing data with complex interactions," in *IEEE International Conference on Big Data*, 2016, pp. 2024-2028.

[11] J. Nawaz, et al., "Fault diagnosis in semiconductor etch equipment using Bayesian networks," *Journal of Semiconductor Technology and Science*, vol. 14(2) pp. 252-261, 2014

[12] B. Cai, et al., "Bayesian networks in fault diagnosis," *IEEE Transactions on Industrial Informatics*, Vol. 13 (5), pp. 2227-2240, Oct. 2017.

[13] N. Stein, C. Flath, "Applying data science for shopfloor performance prediction", in *Proceedings of the 25th European Conference on Information Systems*, Guimares, Portugal, 2017.

[14] D. Zhang, et al., "Predict failures in production lines," *IEEE International Conference on Big Data*, Washington, USA, 5-8 Dec. 2016, pp. 2070-2074.

[15] N. Friedman, M. Goldszmidt, "Discretizing continuous attributes while learning Bayesian networks," *International Conference of Machine Learning*, Bari, Italy, July 1996, pp. 157-167.

[16] M. Sokolova, G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45(4), pp. 427-437, July 2009.

[17] Kursa, M.B. and Rudnicki, W.R., "Feature selection with the Boruta package." *Journal of Statistical Software*, 36(11), pp.1-13, Sept. 2010.