Article title: Performance Prediction Of Production Lines Using Machine Learning Algorithm

Authors: Ayomide Emmanuel Adesiyan [1]

Affiliations: Dundalk Institute of Technology[1]

Orcid ids: 0000-0001-5211-338X[1]

Contact e-mail: e.adesiyan@yahoo.com

License information: This work has been published open access under Creative Commons Attribution License <http://creativecommons.org/licenses/by/4.0/>, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at <https://www.scienceopen.com/>.

Preprint statement: This article is a preprint and has not been peer-reviewed, under consideration and submitted to ScienceOpen Preprints for open peer review.

Links to data: <https://www.kaggle.com/c/bosch-production-line-performance>.

DOI: 10.14293/S2199-1006.1.SOR-.PPA7BE8.v1

Preprint first posted online: 03 November 2021

Keywords: Machine learning, Algorithms, MCC(Mattews correlation coefficient), AUC-ROC(Area under curve the receiver operating characteristics)

AUTHOR : ADESIYAN AYOMIDE EMMANUEL EMAIL: e.adesiyan@yahoo.com.

DISSERTATION TITLE: PERFORMANCE PREDICTION OF PRODUCTION LINES
USING MACHINE LEARNING ALGORITHMS.

FOR A PGD (POST GRADUATE DIPLOMA IN ELECTRICAL/ELECTRONICS
ENGINEERING)

BELLS UNIVERSITY OF TECHNOLOGY, OTA, NIGERIA.

ABSTRACT

Manufacturing today considers data-drive business operations at different levels leading to the growth of various paradigms in manufacturing, of which emerged smart manufacturing.

However data can be used to predict equipment failure rates, streamline and optimize inventory management and prioritize processes.

The use of parameter tuning and optimization, grid-search, cross-validation, to predict the best performing machine learning algorithm. This research work evaluates the time potential failure-rates, against the lines which peaks and drops depending on its components RUL(Remaining Useful Life). The accuracy of the machine learning algorithms that are employed in this studies, are hence subjected to some metrics for evaluation, these are : MCC and AUC-ROC.

This study has analyzed and evaluated some annoymized dataset from a manufacturing company, using some metrics and machine learning algorithms for performance prediction of their production lines using unsupervised learning. This study would served as a good reference for anyone wanting to use the best performance model, for further research work.

Keywords: Machine learning, algorithms, MCC(Mattews correlation coefficient), AUC-ROC(Area Under curve the receiver operating characteristics).

INTRODUCTION.

Traditionally, manufacturing was limited to a process or a sequence of processes through which raw material is converted into finished goods. However, the common understanding of manufacturing comprises much more. Manufacturing today considers the data-driven business operation at different levels leading to the growth of various paradigms in manufacturing, of which emerged smart manufacturing. In modern manufacturing, the volume of data grows at an unprecedented rate in digital manufacturing environments, using barcodes, sensors, vision systems etc. These data may be related to design, products, machines, processes, materials, inventories, maintenance, planning and control, assembly logistics, performances etc and may include patterns, trends, associations and dependencies. However, the use of accumulated data has been limited, which has led to the “rich data but poor information” problem.

Smart manufacturing is being touted as the next industrial revolution (Bryner, 2012). With real time monitoring of manufacturing processes, to increase productivity and stay competitive, the use of data science methods is an obvious next step. For example, the defence manufacturing company Raytheon implemented the MES (manufacturing execution system) in its missile plant in Huntsville, ALA (Hagerty, 2013). This system collect and analyzes factory shop floor data, and was able to find out exactly how many times a screw needs to be turned in a critical component to be perfect. However data can be used to predict equipment failure rates, streamline and optimize inventory management and prioritize processes. In 2012, Intel saved \$3 million in manufacturing costs through the use of predictive analytics to prioritize its silicon chip inspections (Einat Ronen and Kenneth Burns, 2013).

The huge amounts of data in manufacturing databases, which contain large numbers of records, with many attributes that need to be simultaneously explored to discover useful information and knowledge, make manual analysis impractical. All these factors indicate the need for intelligent and automated data analysis methodologies, which might discover useful knowledge from data. ICTs deal with planning and control of production (Davis *et al.*, 2015). Latest trend has shifted to harnessing the power of large scale data set using machine learning and reducing the effort of manual feature engineering. To cope with big data, dimensionality reduction techniques have commonly been used to reduce data size and select important features. In this paper, two methods of dimensionality reduction approach is proposed aiming to determine the total and explained variance of a big dataset with anonymized features.

To explore features of the manufacturing dataset that would be suitable to predict the line performance of production line. These features are anonymized, the naming of the feature follows the convention of “L#_S##_F#####”, which represents the line, station, and feature number (Bosch,2016). The Bosch dataset includes numerical, categorical and time data. The categorical data is extremely sparse (more than 99%) and thus not included in this work. The time data required the label of the preceding data, which may not be availed in a production line and therefore is not considered in the scope of this work. Only the train-numerical dataset is used in this work, using 20,000 samples.

Manufacturing machines are continuously equipped with various sensors, whose data enable to derive a comprehensive picture of the current state of each machine. Predictive maintenance approaches make use of this data in order to predict the occurrence of possible failures before they actually occur, thereby significantly reducing production and service costs. The application of machine learning to sensor data streams is an essential part of data-driven predictive maintenance in order to find the patterns in the data that are indicators of upcoming faults. Thus, research on machine learning for predictive maintenance is a recent and very challenging field.

The rest of the paper is organized as follows: Section II explores the Bosch production line data set using 20,000 entries. Section III To analyze the dataset and perform statistical inferences on the required sampled features using machine learning techniques. To predict the performance of each machine learning models using performance metrics: MCC and AUC-ROC. Section IV discusses the conclusion, recommendation and contribution to knowledge of this research.

Figure 1: A Sequential chart describing the implementation procedure.

SECTION II DATA EXPLORATION

To explore features of the manufacturing dataset that would be suitable to predict the line performance of production line (Kaggle,2016). These features are anonymized, the naming of the feature follows the convention of “L#_S##_F#####”, which represents the line, station, and feature number. The Bosch dataset includes numerical, categorical and time data.

The categorical data is extremely sparse (more than 99%) and thus not included in this work. The time data required the label of the preceding data, which may not be available in a production line and therefore is not considered in the scope of this work. Only the train-numerical dataset is used in this work, using 20,000 samples.

Several challenges are present in this dataset:

- Large scale: the dataset is large with about one thousand features and more than one million observations. And thus, the algorithms need to be scalable with a large dataset.
- Highly imbalanced: the amount of passed and failed products are highly imbalanced because only a small proportion of the products are failures. And thus, the machine learning algorithms need to be able to handle highly imbalanced data.
- Lack of domain knowledge: the features are anonymized. Therefore we do not have domain knowledge of the manufacturing process. The importance of features needs to be discovered by the algorithms.
- High sparsity: the dataset aggregates features from all the stations in the production line. Since many stations serve similar purposes, a product only passes through a fraction of the stations.

I. ANALYSIS OF DATA-SET

For each feature, it has been necessary to fill gaps between data samples due to missing values. Missing values are a common issue related to on-field operating machinery measurements, since many ML algorithms are not efficient in dealing with datasets affected by missing data, figure 2.1 shows how to handle missing values and white-spaces. This has been performed by means of the dropping and replacing “Missing Values” using the linear

interpolation technique, which replaces each missing value with the data.drop and data.replace python syntax.

```
data.drop("Id", axis=1, inplace=True)
```

```
In [9]: data.replace('', np.nan, inplace=True)
```

```
In [10]: data.sample(7)
```

```
Out[10]:
```

	L0_S0_F0	L0_S0_F2	L0_S0_F4	L0_S0_F6	L0_S0_F8	L0_S0_F10	L0_S0_F12	L0_S0_F14	L0_S0_F16
10511	-0.068	-0.086	0.385	0.33	0.074	0.07	0	0.048	-0.036
12687	0.023	0.078	0.33	0.294	-0.143	0.025	0.015	0.048	-0.036
2392	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14331	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3968	0.128	0.168	-0.033	0.003	0.031	0.116	-0.022	-0.152	0.138
612	-0.082	-0.153	0.33	0.33	0.118	-0.021	0.037	0.248	0.102
5436	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7 rows × 969 columns

Figure 2.1: SHOWS HOW TO HANDLE MISSING VALUES AND WHITE-SPACES

II. FEATURE ENGINEERING

Numerical imputation was employed in the code as a more preferable option to preserve the data size. The use of default value of missing values in the column, data.fillna was used to replace NAN (not a number value as 0) then the corresponding row as 1, figure 2.2 shows implementation of pipeline for feature engineering.

Scaling was also used in the numerical features of the dataset which do not have a certain range and they differ from each other , the use of the response feature and the L_S_F values. The standardization scaling (or z- score normalization) was employed since the standard deviation of the L_S_F values is different from the range of the response, the standard scaling reduces the effect of the outliers in the features.

The formula of standardization, the mean is shown as μ and the standard deviation is shown as σ .

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

The pipeline enables the sequence of the data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome.

```
pipe = make_pipeline(StandardScaler(),
                     PCA(n_components=0.95),
                     LogisticRegression(solver='lbfgs'))
# make param grid
param_grid = {'logisticregression__C': [0.01, 0.1, 1, 10, 100]}

# create and fit the model
lr_model = GridSearchCV(pipe, param_grid)
lr_model.fit(X_train, y_train)
y_pred = lr_model.predict(X_test)
```

Figure 2.2 : IMPLEMENTATION OF PIPELINE FOR FEATURE
ENGINEERING.

Section III Statistical Inference on required feature samples using ML techniques.

The machine learning model hyper-parameters have been optimized by means of a grid search and cross validation technique. The results have been validated by means of cross validation technique to assess how accurately the model would perform in practice and to avoid over-fitting. It's important we reduce variability by performing multiple rounds of cross-validation with different subsets of the same data. We then combine the validation results from this multiple rounds to come up with an estimate of the model's predictive performance. The dataset was split into features and targets. This research evaluated three machine models: Logistic Regression, Gradient Boosting, Random Forest.

- a. **Logistic Regression:** This is a classification algorithm used to assign observations to discrete set of classes. It is a good predictive analysis algorithm and based on concept probability, logistic regression is a linear regression model but it uses a more complex cost function which can be defined as the "SIGMOID FUNCTION". The hypothesis of logistic regression tends to limit the cost function between 1 and 0.
- b. **Gradient Boosting:** Boosting is a method of converting weak learners into strong learners. In boosting each new tree is a fit on a modified version of the original dataset. Gradient boosting trains many models in a gradual, additive and sequential manner. Gradient boosting identifies weak learners using gradients in the loss function $Y = ax + b + e$, where e is the special error estimator term.

This research work used the loss function to measure specifically how good are the model coefficients are at fitting the underlying data. Haven divided the data into train and test set, where the $n_components = 0.95$, we therefore made a predictions and gauge the performance of the model, using a confusion matrix.

- c. **Random Forests:** This technique can handle big data with numerous variables running into thousands just like our dataset. It can automatically balance datasets when a class is more infrequent than other classes in the data. The method also handles variables fast, making it suitable for complicated task. Random forests present estimates for variable importance, i.e, neural nets. They also offer a superior method of working with missing data. Missing values are substituted by the variable appearing the most in a particular node.

The description of the dataset is, as shown in Fig. 3. The statistical description of the dataset, using data frame command in python.

```
In [98]: df.describe()
Out[98]:
```

	Id	Response
count	20000.000000	20000.000000
mean	19967.531350	0.005600
std	11555.759467	0.074625
min	4.000000	0.000000
25%	9974.750000	0.000000
50%	19924.000000	0.000000
75%	30033.500000	0.000000
max	40065.000000	1.000000

```
In [41]: print(df.shape)
df.head()
(20000, 970)
```

Fig 3: Statistical inference of the dataset reading 20,000 entries.

3.1 PERFORMANCE METRICE FOR CLASSIFICATION/EVALUATION

a. **MCC (Matthews Correlation Coefficient):**

This metrics takes into account all four values in the confusion matrix, and a high value (close to 1) means that both classes are predicted well, even if one class is disproportionately under – (or over-) represented.

The Matthews correlation coefficient (MCC) was chosen as the metrics, defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \dots\dots (2).$$

Where TP, TN, FP, FN represent True Positive, True Negative, False Positive and False Negative values in the confusion matrix. The MCC score usually ranges from zero (random guess) to one (perfect classification). MCC is calculated directly from the binary prediction results. A threshold is required to convert the probability into the binary result the threshold was optimized toward the highest MCC score on the cross-validation set.

3.2 AUC-ROC (Area Under the Receiver Operating Characteristics):

The Area Under the Receiver Operating Characteristic Curve (AUROC) is used as an additional metric for the robustness of the algorithm. The classification results are provided as the probability of whether or not the data belongs to one class. The ROC curve is created by plotting the true positive (TP) rate against the false positive (FP) rate at various threshold settings. The AUROC usually ranges from 0.5 (random guess) to one (perfect classification.).

In the next chapter, we are going to discuss the result of the performance of each model varying its relationship with the metrics.

Results obtained from the analysis of the data collected from Bosch production line, with the use of phyton programming language are presented in this section. Also, results of output value of two components, variance between two components, increments in the components versus the explained variance, the UMAP graph describing sparsity between the components, performance of samples statistical description, relative indexing, confusion matrix describing the performance of the model, failure rate versus Production line and performance of machine learning algorithms used are all presented in figure 3.1 to 3.5 respectively.

Consequently, the analysis results are compared with other existing predicting models.

figure 3.1(a): Shows the data-frame of the two components using `n_components = 2`, after applying PCA (Principal Component Analysis) to `fit_transform` the data. The result then shows that the components variance lies between 0 and 1 , showing its total variance

```
# Use the resulting components to create a new data frame
pca_df = pd.DataFrame(data=pca_fitted,
                      columns=["Component 1", "Component 2"])
print(pca.explained_variance_ratio_)
pca_df.head()
```

```
[0.014347  0.01249095]
```

	Component 1	Component 2
0	-5.571398	-0.010247
1	5.629177	0.308387
2	0.799222	-1.868581
3	1.592479	-0.701956
4	-1.640958	-0.351579

Fig 3.1(a) SHOWS THE OUTPUT VALUES OF THE TWO COMPONENTS

figure 3.1(b): Describes the total variance of the two components, using two line features L1 and L2. The two components only contains 2.68% of the information in the response (TARGET). The total explained variance of the two components is 95.03%, this shows that despite the dispersion in the dataset, the dataset is useful. CV of component1 ≥ 1 indicates a relatively high variation than component2 whose CV < 1 which is considered a low variation.

Total Explained Variance with just two components: 95.03%

FIG 3.1(b) SHOWS THE VARIANCE BETWEEN THE TWO COMPONENTS.

Fig 3.2(a): Describes the increment in components the need to reduce dimensionality of the features by projection, but this in-turn may lead to loss of the variance (information). In order to see how much variance PCA is able to explain as the number of components increases to 500 the explained variance would be equals 1, we can make a plot. This will enable us to know many dimensions to ultimately keep or analyze.

Fig 3.2(a) DESCRIBES THE INCREMENTS IN THE COMPONENTS VS THE EXPLAINED VARIANCE.

Fig 3.2(b): Describes how UMAP (Uniform Manifold Approximation Projection) is used to scale the dataset in-terms of size and dimensionality. Although it poses as a strong algorithm for local clustering of similar groups, as seen in the graph where the components that have variance equal or greater than 1 are the yellow points.

Fig 3.2(b). A UMAP GRAPH DESCRIBING THE SPARSITY BETWEEN THE COMPONENTS.

Fig 3.3(a): Using the confusion matrix to classify the predicted lines and their corresponding response, then proceeded further to iterate the lines against their response, each line is situated in a station it was therefore observed from the code that about 112 lines out of the total sample dataset 20000 have potentially failed, this therefore has caused a decline in the total line performance, however it wasn't specified which line has the majority potential failure. The accuracy score here utilizes MSE but this is not a true reflection of the predictive power of the model because the classes are imbalanced (many fail i.e 0) and as such, the model fails woefully in its precision (precision of 20%) when the performance is passed (i.e 1). This can be illustrated better using confusion matrix.

FIG 3.3(a) THIS CONFUSION MATRIX DESCRIBES THE PERFORMANCE OF THE MODEL.

Fig 3.3(b): Describes the use of parameter tuning and optimization with Grid search and cross validation can aid in predicting the best performance machine learning algorithm for performance prediction that suits this dataset. However failure of line components can causes total breakdown of the production line and thereby causing the station downtime. This research work evaluates the time potential failure rates against the lines which peaks and drops depending in its components RUL (REMAINING USEFUL LIFE).

Fig 3.3(b) . SHOWS THE FAILURE RATE VS THE LINES.

Fig 3.4: Shows the accuracy of the machine learning algorithms that are employed in this studies, are then subjected to some metrics for evaluation, there are MCC(MATTEWS CORRELLATION COEFFICIENT) and AUC-ROC(AREA UNDER CURVE THE RECIEVER OPERATING CHARACTERISTICS).AUC - ROC curve is a performance evaluation metric for classification tasks in ML. It tells us how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. An excellent model has AUC near to the 1 which means it has good measure of reparability. A poor model has AUC near to the 0 which means it has worst measure of reparability. In the next chapter we would discuss about the best performance algorithm, limitations of the work and its contribution to knowledge.

A	B	C	D	E
	Algorithm	Accuracy (%)	MCC	AUC (%)
0	Logistic Regression	99	-0.2	50
1	Random Forest	99	10.65	52.8
2	Gradient Boosting	100	0	49.9

fig 3.4 SHOWS THE PERFORMANCE OF EACH MACHINE LEARNING ALGORITHM.

Fig 3.5 SHOWS A GRAPHICAL ILLUSTRATION OF THE PERFORMANCE VALUE ALL THE EMPLOYED ALGORITHMS AGAINST THE METRICS.

SECTION IV

CONCLUSION

This research has considered three major ML algorithms which are Logistics Regression, Gradient Boosting and Random Forest. It is observed that Random Forest gave the best performance in terms of MCC and AUC-ROC metrics. Although the data used is an anonymized, Random forest selects random samples(lines) from the dataset, constructs a decision tree for each sample and get a prediction result from each decision tree, then performs a vote on each predicted result. Selects the prediction result with the most votes as the final prediction. From this research it is obtained that about 0.0056% of the lines have failed due to lack of proper maintenance, their inability to produce optimally would lead to loss in revenue.

The prediction probabilities for the entire training population are sorted descending, and then divided into deciles. For a random chance model, in order to cover half of the failed lines, we need to validate 50% of the entire products population. However, with our model, we can see that if we target the top 2 decile failure probabilities from the model, we capture close to 50% of our failure target (response) population. This is much better than checking for defects at random.

RECOMMENDATION

To achieve better predictions, the features need to be clustered into these categories and separate models should be made for each category. We observed that there is a weekly periodicity to the number of observations recorded per day. This insight provides us an opportunity to improve the model by incorporating complex date based features taking into account the variability observed due to change in factory conditions over-time. Due to the nature of Big-Data Analytics manufacturing companies can make do of cloud based services

like HADOOP and SPARK for big-data analytics problems. Large-scale manufacturing data is being generated as processes becoming more intelligent. The emergence of big data provides both the opportunity for using predictive models in performance management and the challenge for data processing. In this paper, we proposed methods to predict performance in a production line. During the first step, the data is clustered into similar process groups. Then unsupervised learning is applied to each cluster to predict performance of the line. This approach reduces the sparsity of the data set by eliminating irrelevant features.

CONTRIBUTION TO KNOWLEDGE

This study has evaluated the accuracy of the employed machine learning algorithms for performance prediction of production lines. The results of this study can be used as a base reference technique against which other results obtained from other performance indexes can be compared with.

REFERENCES

1. Davis J, Edgar T, Graybill R, et al. (2015) Smart manufacturing. *Annu Rev Chem Biomol Eng* 6: 141–160.
2. Bryner, (2012). “Smart manufacturing: The next revolution,” *Chemical Engineering Progress*, vol. 108, no. 10, pp. 4–12.
3. J. Hagerty, (2013) “How Many Turns in a Screw? Big Data Knows,” *The Wall Street Journal*.
4. A. C. Einat Ronen, and Kenneth Burns, (2013) “Utilizing Predictive Sales Analytics with Big Data,” Intel, Tech. Rep., Online.
5. Kaggle, “Bosch Production Line Performance,” 2016. [Online]. Available: <https://www.kaggle.com/c/bosch-production-line-performance>.