Cost-sensitive optimization of automated inspection

Goktug T. Cinar†
Robert Bosch LLC
4009 Miranda Avenue
Palo Alto, CA 94304 USA
Email: Goktug.Cinar@us.bosch.com

Jeffrey Thompson‡
Robert Bosch LLC
4009 Miranda Avenue
Palo Alto, CA 94304 USA
Email: Jeff.Thompson@us.bosch.com

Soundar Srinivasan
Robert Bosch LLC
4009 Miranda Avenue
Palo Alto, CA 94304 USA
Email: Soundar.Srinivasan@us.bosch.com

Abstract—Automated inspection plays a critical role in many industrial processes, including modern assembly lines. In these processes, components are inspected to ensure adherence to design specifications. Components that are determined to be out-of-specifications are rejected. The benefits of inspection are two-fold. First, defects can be removed early in the process, preventing higher costs incurred in detecting them downstream. Second, the inspection results provide information for manual troubleshooting of root-causes, potentially leading to an improvement in overall quality. However, this form of inspection also incurs costs if there are false alarms associated with the automated inspection method. Analysis of false alarm costs are rarely addressed in the data mining literature. In this paper, we develop a simple framework to estimate the value of an inspection process, and demonstrate how predictive modeling can be used to increase this value under the right circumstances. In the second part of the paper, we report results from a case-study at a Bosch manufacturing plant, involving tens of thousands of parts and hundreds of quality attributes. A key challenge in this study was the extremely low rate of defects resulting from the operation of a highly-optimized manufacturing process. We show that for such modern assembly lines, machine learning techniques that are robust to class imbalance are particularly well suited. The solution from the case-study yields a positive ROI for the manufacturing plant.

Keywords: Advanced manufacturing; quality control; optimization; prediction; Big Data

I. INTRODUCTION

Machine learning techniques are increasingly being used in manufacturing applications [1], [2], [3], [4], [5], [6]. These applications include: predictive maintenance, test-time reduction, supply chain optimization, process flow optimization, and automated inspection [7]. In advanced manufacturing plants, automated inspection systems are an important part of quality control. Measurements from individual components are recorded at strategic locations along an assembly line to ensure that they meet specifications defined by the product or process design. When a component is found to be out-of-specification, it is routed outside the normal process flow where it is either reworked or discarded (scrapped) depending on pre-defined procedures. Manufacturing plants incur significant costs from such defects, both in terms of lost materials, and lost time. The value generated by an automated inspection process depends on a number of factors including: the production volume, the defect rate at the point of inspection, the material/rework costs, and the inspection performance. An important property of many inspection systems is that they are predictive in nature. The recorded measurements do not, on their own, guarantee product performance. Instead, many inspection measurements are correlated with downstream functional performance or non-performance (failures). This occurs in a wide variety of contexts, especially in cases where the measurements have high uncertainty, or the relationship between physical and functional properties is complex. In these cases, assembly line inspection can be viewed as a binary classification problem where the input features are measurements from inspection equipment, and the label is the result of a downstream functional test. The downstream test may be internal (e.g. an end-of-the-line test on the same assembly line) or external (e.g. routine use by a customer, or acceptance testing by an original equipment manufacturer). The goal of the inspection is to prevent costly downstream failures by rejecting faulty components early in the assembly process.

While binary classification is a well studied problem, only a few studies address the cost-benefits of a classifier’s performance in real applications [8]. This is particularly critical under severe class imbalance conditions due to the fact that even a low false-positive rate of a classifier could result in a negative return-on-investment (ROI) and make it unsuitable for deployment. Therefore, in the first part of this paper, we develop an easy-to-interpret model that relates the value of an inspection to the classification performance of the inspection, expressed in terms of true and false positive rates. One use for this model is in optimization of inspection performance through predictive modeling. By linking performance directly to economic value, design trade-offs that are inherent in predictive modeling are made more concrete, helping the assessment of automated inspections in terms of their ROI. The second part of this paper is devoted to a case study of a specific inspection station at a Bosch manufacturing plant. We highlight the potential benefits and challenges of applying predictive models to assembly

†Authors contributed to this work equally.
line inspection, also give insights to the performance and limitations of two commonly used machine learning methods in this context. Finally we discuss the potential conditions that can be observed when applying the model of inspection value to real life use cases, and the benefits/limitations of predictive modeling under those circumstances.

II. A MODEL OF INSPECTION VALUE

The lack of literature on the cost-benefit relationship of predictive models in real applications motivates us to introduce the following model of inspection value. We define the value of an inspection process \( v \) as the total reduction in defect costs \( s \) attributable to the inspection.

\[
v = s_o - s_i \tag{1}
\]

where \( s_i \) and \( s_o \) are defect costs with and without inspection, respectively. All costs are expressed as a rate in dollars per unit time. Methods for estimating defect costs are beyond the scope of this paper but typical sources include: lost materials, labor associated with rework and handling, effect on cycle time, disposal costs, and risks associated with warranty claims and recalls. Generally, these costs are specific to the product, the components used, the labor skill level and the locality within the assembly process. \( s_o \) can be expressed in terms of the production rate \( p \), the defect rate \( d \), and the downstream defect cost \( c_d \), where \( c_d \) is the cost associated with a specific defect the inspection is intended to prevent. For example, \( c_d \) could be the cost of scrapping a part at end-of-line, or the cost for the manufacturer that is associated with a warranty claim.

\[
s_o = p \times d \times c_d \tag{2}
\]

To derive an expression for \( s_i \) we consider each of the four possible inspection outcomes: true positives, true negatives, false positives, and false negatives. Here, we have defined arbitrarily a rejected component as a ‘positive’ outcome. We also consider the possibility that inspection may provide a reduction in the defect rate by virtue of improved root-cause analysis by process engineers. This effect is captured in the variable \( t \).

\[
s_i = \quad p [tp \times c_i \times (d - t) + ... + fp \times c_i \times (1 - d + t) + fn \times c_d \times (d - t)] \tag{3}
\]

where \( tp, fp, \) and \( fn \) are the true positive rate, false positive rate and false negative rate, respectively. \( c_i \) is the defect costs incurred when detected by the inspection process. This equation is composed of three terms. The first term is the cost associated with correct detection of defect parts at inspection, the second term is the cost associated with incorrect classification of a good part as bad at inspection, and the third term represents the cost of defects that go undetected by inspection. Note that true negatives are not included in the expression because no defect costs are incurred by correct classification of good components. \( c_i \) is always less than or equal to \( c_d \) because the decision is made earlier in the assembly process. However, the magnitude of this difference is dependent on specific circumstances.

Given the above expressions for \( s_o \) and \( s_i \), equation (1) can be rewritten in the following form, using the fact that \( fn = 1 - tp \).

\[
v = p[t \times c_d + tp \times (c_d - c_i) \times (d - t) - fp \times c_i \times (1 - d + t)] \tag{4}
\]

The first two terms of equation (4) represent potential cost savings. Let us examine them in detail. The first term results from improved root-cause analysis, and the second from correctly detected defects. Note that the cost savings in the second term are directly proportional to the cost differential between detecting a defect early \((c_i)\) and late \((c_d)\) on the assembly line. The last term represents additional costs associated with inspection due to false alarms. Very few, if any, classification models in the real world are 100% accurate. Therefore, when the inspection algorithm incorrectly classifies a good component as bad, an additional cost is incurred that is proportional to \( fp \) and \( c_i \). Here, we have assumed that the inspection in question does not affect cycle time, so the cost of a false alarm is limited to the cost of materials as it does not affect the production rate. In the case that the inspection is cycle-time limiting, the inspection value model would need to incorporate a value loss term to account for the reduced production rate due to false alarms.

A. Inspection optimization through predictive modeling

Based on the model described above, increasing the value of an inspection process can be achieved by: increasing the value of \( t \), increasing the value of \( tp \) and/or reducing the value of \( fp \). Clearly, increases in the value of \( t \) are the ideal outcome because this translates into a reduction in overall defect costs. Methods for increasing \( t \) generally fall under the umbrella of traditional quality control techniques such as Six-Sigma, Shainin, and Taguchi methods [9], [10], [11], [12], [13], [14]. The extent to which the data from an inspection station contributes to the success of these methods is highly context specific. It is also possible to use predictive modeling techniques for the purpose of root cause analysis. In most cases, this involves estimating the ‘importance’ of individual parameters in a model and then combining this knowledge with specific domain expertise to link the most important parameters to a specific problem on the assembly line. Although we recognize the critical importance of root cause analysis in manufacturing quality control, it is beyond the scope of data mining solutions that are addressed in this work. Instead, we focus on methods for increasing \( tp \) and reducing \( fp \) through the application of predictive modeling techniques. We also note that the value of \( t \) is, in most cases,
unaffected by changes in $tp$ and $fp$. This is because root cause analysis is often based on the raw measurements of the inspection station, instead of the decision criteria used to determine if a part passes or fails the inspection. Therefore, predictive models that are focused on improving $tp$ and $fp$ can be regarded as complimentary to methods that try to identify the root cause of defects.

In the context of an existing inspection station, where the value of root cause analysis has already been realized, we can ignore the effect of $t$ and focus on the latter two terms of equation (4). This expression can be simplified by expressing $c_d$ in terms of $c_i$ and a 'savings factor' $k$ such that $c_d = k \times c_i$ where $k \geq 1$. In this equation, $d$ represents the defect rate after accounting for root-cause improvements.

$$v = (p \times c_i)(tp \times (k - 1) \times d - fp \times (1 - d))$$ (5)

To understand how the inspection value depends on the performance of the predictive model, we identify "break-even lines" that show where $v(tp, fp) = 0$ lies on the model’s ROC curve. In order for the inspection process to be valuable, the operating point of the ROC curve must lie above this line (upper left). In figures 1 and 2, we show the effect of the defect rate $d$, and the savings factor $k$ on this break-even line. In figure 1, $k$ is set to 3, which means that the cost of a downstream defect is three times the cost of a defect detected at inspection. It is clear that the smaller the defect rate, the steeper the slope of the break-even line, and the greater the performance burden for the predictive model. Intuitively, as the defect rate is reduced, there are fewer bad parts to detect, and the absolute number of false positives increases for a fixed $fp$. Therefore, a low defect rate imposes a very low threshold requirement on the model's $fp$. An additional problem associated with low defect rates is the problem of 'class imbalance'. Class imbalance refers to the situation where the labels in a classification task occur with very different frequencies. In the case of an assembly line, the most common scenario is that there are many more good parts than defective parts. In the use-case described in the next section, the underlying defect rate was less than 1%. In Figure 2 we set the defect rate to $d = 0.01$ and varied the savings factor $k$. Note that $k$ is important because it determines the cost savings for each correctly detected defect. The larger the savings rate, the greater the opportunity for cost savings and the lower the performance requirements of the model. Taken together, these figures demonstrate the need for classifiers with extremely low false-positive rates due to the low defect rate that is typical on advanced manufacturing lines.

In the next section, we present a case study from a Bosch assembly line where a predictive model was developed for the purpose of preventing downstream failures of an electronic control unit (ECU). We discuss the advantages and disadvantages of various machine learning techniques with respect to this specific scenario and the model for inspection value given in this section.

III. A CASE STUDY OF ASSEMBLY LINE INSPECTION USING PREDICTIVE MODELING

The main challenge associated with applying machine learning to automated inspection is the 'class imbalance' problem. The class imbalance problem occurs when there are many more instances of one class than another in a classification task [15]. Since there are typically many more good parts than defect parts, the models and algorithms applied must account for this reality. Otherwise, the model is unlikely to provide significant value to the assembly process, and could even generate unnecessary costs due to false positives. Other applications where class imbalance is
techniques attempt to maximize classification accuracy with-
of class imbalance is two fold. First, many machine learning
this model useless. The challenge of learning in the context
differential that favors the minority class which would render
this model useless. The challenge of learning in the context
of class imbalance is two fold. First, many machine learning
methods are trained in sequence. Each weak classifier is trained
high performing classifier. Each weak classifier is trained
in sequence on a subset of the data, and diversity among
the classifiers is promoted by favoring data samples that
are misclassified by models trained earlier in the sequence.
The prediction of each weak classifier is then combined to
produce a final prediction[2].

B. Algorithm Level Approaches to Class Imbalance

The major algorithmic approaches to handling class
imbalance includes cost-sensitive learning [23], one-class
learning [24] and boosting [25]. Cost-sensitive learning
focuses on adjusting the cost function of standard classi-
sifiers such that errors with higher misclassification cost are
weighed appropriately. These methods are appropriate when
the relative costs of misclassification are known and can
be utilized during the training of the model, for instance by
moving the decision boundary of a threshold based classifier.
One-class learning is a paradigm where classifiers are trained
only on the samples of a single class. In the context of
class imbalance, a one-class classifier would be trained
only on samples from the minority class. Raskutti and
Kowalczyk have shown that one-class SVMs yield improved
classification accuracy over standard two-class classifiers
in the presence of extreme class imbalance [26]. Boosting
is a specific type of ensemble method, where multiple
'weak' classifiers are combined together to create a single
high performing classifier. Each weak classifier is trained
in sequence on a subset of the data, and diversity among
the classifiers is promoted by favoring data samples that
are misclassified by models trained earlier in the sequence.
The prediction of each weak classifier is then combined to
produce a final prediction[2].

C. Evaluation Metrics

Since classification accuracy by itself is insufficient to
judge the performance of a classifier in the presence of
class imbalance, performance metrics that consider both the
true positive rate and false positive rate of the classifier are
preferred [2], [27]. Such metrics include the Kolmogorov-
Smirnov statistic, and area under the receiver-operating
characteristic curve (AUC) [2]. In this case-study we use
the AUC metric.

D. Description of Process Data

The use case that we focus in this paper is the prediction
of downstream failures on an assembly line. The data is
collected from an active Bosch assembly line that man-
ufactures electronic control units (ECUs). The production
line consists of multiple processes and inspection stations.
Measurements from each station are recorded in a central
database and in log files local to each station. Data is
recorded both in real-time and archived in manufacturing
execution systems and enterprise resources planning sys-
tems. We note that in safety-critical applications, such as
in automotive applications, data is archived for traceability

prevalent include: healthcare (e.g. medical diagnosis) [16],
fraud detection [17], process quality detection [18], and text
classification [19].

In the presence of class imbalance, standard supervised
learning techniques tend to learn the over represented class
very well, but at the expense of accuracy on the minority
class. For example, in a data set where the minority class
makes up less than 1% of the data set, a simple classifier
that labels everything as the majority class would obtain
99% accuracy. However, in many domains, there is a cost
differential that favors the minority class which would render
the manufacturing context where the data sets tend to be
quite large. However, a major downside to undersampling is
the information loss introduced by neglected samples. This
problem is usually addressed through ensemble methods as
explained in the next section.

A. Data Level Approaches to Class Imbalance

Data level approaches for handling class imbalance lever-
age various data sampling and re-sampling techniques. Sam-
ping methods can be divided into two groups, oversampling
and undersampling. Oversampling methods include: random
oversampling of the minority class, selective oversampling
from parts of feature space close to class boundaries [20],
and generating synthetic samples of the minority class [21].
Synthetically generating new samples (e.g. SMOTE) has
been shown to improve classification performance in the
context of class imbalance [2]. However in many data
sets, including manufacturing data, it is not clear how to
generate realistic synthetic samples. Undersampling methods
include: random undersampling of the majority class, selec-
tive undersampling from parts of feature space close to class
boundaries [20], and sequential undersampling. Sequential
undersampling is an ensemble method in which multiple
models are trained in sequence. After each iteration, cor-
rectly classified samples from the majority class are removed
from the training data [22]. One advantage of undersampling
techniques is that they are generally more computationally
efficient for training models as only a subset of the majority
class samples are required. This is especially valuable in

D. Description of Process Data

The use case that we focus in this paper is the prediction
of downstream failures on an assembly line. The data is
collected from an active Bosch assembly line that man-
ufactures electronic control units (ECUs). The production
line consists of multiple processes and inspection stations.
Measurements from each station are recorded in a central
database and in log files local to each station. Data is
recorded both in real-time and archived in manufacturing
execution systems and enterprise resources planning sys-
tems. We note that in safety-critical applications, such as
in automotive applications, data is archived for traceability

prevalent include: healthcare (e.g. medical diagnosis) [16],
fraud detection [17], process quality detection [18], and text
classification [19].

In the presence of class imbalance, standard supervised
learning techniques tend to learn the over represented class
very well, but at the expense of accuracy on the minority
class. For example, in a data set where the minority class
makes up less than 1% of the data set, a simple classifier
that labels everything as the majority class would obtain
99% accuracy. However, in many domains, there is a cost
differential that favors the minority class which would render
the manufacturing context where the data sets tend to be
quite large. However, a major downside to undersampling is
the information loss introduced by neglected samples. This
problem is usually addressed through ensemble methods as
explained in the next section.

A. Data Level Approaches to Class Imbalance

Data level approaches for handling class imbalance lever-
age various data sampling and re-sampling techniques. Sam-
ping methods can be divided into two groups, oversampling
and undersampling. Oversampling methods include: random
oversampling of the minority class, selective oversampling
from parts of feature space close to class boundaries [20],
and generating synthetic samples of the minority class [21].
Synthetically generating new samples (e.g. SMOTE) has
been shown to improve classification performance in the
context of class imbalance [2]. However in many data
sets, including manufacturing data, it is not clear how to
generate realistic synthetic samples. Undersampling methods
include: random undersampling of the majority class, selec-
tive undersampling from parts of feature space close to class
boundaries [20], and sequential undersampling. Sequential
undersampling is an ensemble method in which multiple
models are trained in sequence. After each iteration, cor-
rectly classified samples from the majority class are removed
from the training data [22]. One advantage of undersampling
techniques is that they are generally more computationally
efficient for training models as only a subset of the majority
class samples are required. This is especially valuable in
and forensic analysis. This results in a large data volume and rate (typically a few Terabytes a year). In this case study, we focus on failures at a single station where a printed circuit board (PCB) is joined to a housing unit by aligning multiple pins in the housing unit to corresponding holes in the PCB. The housing pins provide both electrical contact with actuators and also mechanical stability for the PCB. Failures at this station generally occur when one or more pins are not aligned properly with the PCB. When this occurs, both parts (PCB and housing) are scrapped. We collected data from two assembly lines during one month of production. During this period the failure rate at this station was 0.25\%(d = 0.0025), which represents an extremely imbalanced data set for classification. This low failure rate is the result of optimized manufacturing processes at Bosch plants and is typical of many advanced manufacturing lines. The data set contains approximately 700 measurements per part related to the housing component collected from stations ‘upstream’ from the press-in station. The bulk of these measurements are related to pin positions measured by automated optical inspection equipment. The task is to predict, in real-time, housing components that are likely to cause failures at the press-in station, and then removing them before they enter the station. When successful, this has the benefit of saving the cost of the more expensive PCB, which would otherwise be scrapped along with the housing component. An approximate value of the savings factor in this scenario is 3.

E. Model Features

Data from upstream stations were processed into a set of feature vectors for input to modeling algorithms. The raw data collected from the plant (in the order of Terabytes) is stored in a number of different formats and requires an involved ETL process to extract the modeling features. Below is a description of the features we used for modeling.

1) Measurement Features: The large majority of our feature vectors are generated from upstream measurements of the housing component. In this context ‘upstream measurements’ refer to any measurement made on the assembly line prior to the press-in station. We used 689 measurement features, the large majority of which were derived from automated optical inspection of the housing pins. An optical image of each housing component is used to estimate the positions of each pin. These positions are compared against design specification and the deviations from the spec are recorded in a log file. These measurements include, absolute pin locations, deviation from expected locations, and a variety of reference positions and image quality metrics. Of the 689 measurements, 666 are generated from this optical inspection. The remaining 23 measurements are related to various process steps such as pin insertion and placement of various small components.

2) Temporal Features: The features described above apply only to the current part. However, as the manufacturing process is sequential in nature, there may be predictive value in measurements from previous parts or machine events. For example when a machine fault occurs, there may be an increase in the likelihood of near-term future failures. To capture this information, we generated the following features from the raw data:

1- Time since last failure
2- Recent failure rate (failure rate within last 200 parts)
3- Recent model transition rate (the rate at which the model number was changed within last 200 parts)
4- Per part pass/fail history from the press in station over the previous 50 parts.

F. Model Description and Performance

Quality inspection through predictive modeling and machine learning has gained increased attention in recent years [1], [2], [3], [4], [5], [6]. Solutions proposed in the literature for quality control and prediction are highly use-case dependent. However, one common finding is that sampling techniques, in general, tend to improve classification performance in the presence of class imbalance [2]. For the current use-case, we evaluated a small set of random undersampling (RUS) techniques combined with ensemble methods in order to minimize information loss. Figure 3 depicts how ensemble methods can be used in conjunction with RUS.

Specifically, we applied the following classification models in this case study:

1- RUSBoost [25]
2- RUSBoost with a cost matrix that accounts for the cost differential of true and false positives
3- Ensemble of Neural Networks (NN) with a fixed topology and fixed feature set
4- Ensemble of Neural Networks with a randomly varied topology and feature set

Table I: Model false positive rates (for true positive rate of 0.4) for different feature sets

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Measurement (L1 reg.)</td>
<td>0.037</td>
</tr>
<tr>
<td>Temporal Measurement (KS test)</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.022</td>
</tr>
</tbody>
</table>

Due to the large number of measurement features, the
potential for over-fitting is high, especially with such a small number of minority samples. Therefore, we used two different methods to reduce the number of measurement features while maintaining predictive accuracy:

1. Top 40 features (highest regression weights) from an L1-regularized logistic regression model.
2. Top 10 features (that show the highest variability) from a Kolmogorov-Smirnov test comparing the distributions of each feature between positive and negative samples in the training set.

To select a feature set, we trained an ensemble of neural networks with a fixed topology (1 hidden layer 30 hidden units) on three different candidate feature sets. In Table I we report the false positive rate for a fixed true positive rate of 0.4. As the ROC curve is desired to be close to the y-axis, the smaller values are a sign of better classification performance in this case. We found that the feature set with temporal and KS test measurement features performed the best. Therefore we trained the final ensemble models on all temporal and measurement features identified by the KS test statistic.

We used ensemble models in conjunction with random undersampling to predict failures in the presence of class imbalance. In all, we trained 4 predictive models: 2 based on RUSBoost [25] and 2 based on a custom implementation of neural network ensembles. The first ensemble neural network model had 200 equally weighted weak learners trained on a fixed set of features and a fixed topology (one hidden layer with 30 hidden units). To create a more diverse set of weak learners, the second ensemble neural network model used a randomly varied topology and feature set. For each set of features and network topologies (1 hidden layer with 10/20/30 hidden layers), we trained 50 weak learners. Therefore the second model had a total of 450 equally weighted weak learners.

In Table II and Figure 4 we report performance of the four ensemble models trained in this case study. All models perform very similarly in terms of their ROC curves. However, this does not imply that all models behave identically or have learned the same failures. Due to the severe class imbalance in this data set, the jumps in the true positive rate axis of the ROC curve are determined by a small set of positive samples. This makes the significant differences that take place in the false positive rate (FPR) axis hard to visualize. To visualize this we zoom in closer to the origin in Figure 5. We see that the models capture different failures (captured as jumps in the ROC curve). Also in Figure 6 we project the test samples from the feature space to a two-dimensional space for purposes of visualization, and highlight the samples that are false positives for two of the algorithms at 0.25 true positive rate. Only a small section of the projected space is shown for clarity of the visualization. As it is infeasible to visualize the decision boundaries of these ensembles in high dimensional feature spaces, this simple visualization gives us a sense of the different structures captured by the two models. Overall, the ensemble neural network model, with varying topology and feature sets, outperformed the other models by a small margin in terms of AUC. The ensemble of these models (RUSBoost and NNs) won’t create a significant difference as the new decision boundary would be a function of the existing boundaries. It is obvious that with the class imbalance the decision boundaries are highly populated with negative samples. As none of the ROC curves has significantly superior performance over the others in any region, the ensemble of them is not likely to improve the classification performance. Therefore alternative methods where we map the feature space to higher dimensions, such as kernel classification methods, can prove useful.

### Table II: The table shows AUC values for the ROC curve shown in Figure 4.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUSBoost</td>
<td>0.8195</td>
</tr>
<tr>
<td>RUSBoost w/ cost</td>
<td>0.8228</td>
</tr>
<tr>
<td>Ensemble NN</td>
<td>0.8282</td>
</tr>
<tr>
<td>Split Ensemble NN</td>
<td>0.8397</td>
</tr>
</tbody>
</table>

### G. Conclusion and status of the case study

In the first part of this paper we have introduced a framework to estimate the value of an inspection process. We have shown the dependence of the value on two parameters, the savings factor (k) and the defect rate (d). Here we will assume four different combinations of the savings factor and the defect rate, and share our insights regarding the possible machine learning techniques to be used in these scenarios.
Figure 5: A closer look at the low false positive rate region of the ROC curves depicted in Figure 4.

Figure 6: The figure shows the false positive samples labeled by two different algorithms at a 0.25 true positive rate.

- **High savings rate & high defect rate**: This is the scenario where there is a lot of positive samples (defects) in the data set and removal of these defects early in the production line accounts to significant savings. In this scenario the problem will not suffer from extreme class imbalance. Standard machine learning techniques (i.e. decision trees, ANNs, SVMs etc.) can lead to useful predictive models as they will be able to capture the structure of the positive class easily.

- **Low savings rate & high defect rate**: In this scenario the effects of the class imbalance is also not severe, however the value loss associated with false positives from the inspection becomes a significant issue. Similar to the previous case, standard machine learning algorithms can capture the structure of the data and can be used for predictive modeling. As the savings line moves closer to the y-axis, the operating points of the models should be chosen to have lower false positive rates. However, the previous two scenarios are generally unlikely to be encountered in a manufacturing setting. Manufacturing plants have teams of process engineers that actively monitor and adjust the manufacturing process to achieve very low defect rates. The promise of big data and machine learning techniques is that they may allow for further improvements in addition to this existing low defect rate.

- **High savings rate & low defect rate**: This scenario applies to manufacturing lines where significant value is added to a part after the inspection process and/or the cost of a downstream failure is very high relative to the cost of a failed inspection earlier in the line. As lower defect rates leads to class imbalance, ensemble methods and sampling strategies should be used as described in Sections III-A and III-B. Similarly the predictive models developed in this scenario would have to operate under low false positive rates. However, given the high savings rate, significant savings may be realized even when operating at a low true positive rate.

- **Low savings rate & low defect rate**: This scenario is the most challenging in terms of predictive modeling efforts. This category requires models that operate with a high true positive rates and low false positive rate. Given the severe class imbalance, standard machine learning models can have difficulty in learning even with the use of ensemble methods and sampling strategies. As the savings line is very close to the y-axis, even models with high AUCs (≥ 0.8) might not lead to significant savings. Below we explain the outcomes of our predictive modeling efforts in such a scenario.

In this case study, we have a relatively low savings factor ($k = 3$) and very low defect ($d \approx 0.0025$) rate. To put the defect rate value in this study into a perspective, a recently published study in [28], compares multiple machine learning algorithms such as neural networks, decision trees, rotation forests, k-nearest neighbor etc. and reports good performance by rotation forests in presence of class imbalance “as low as 6.5%”. In comparison we are working with a class imbalance of $\approx 0.25%$. This brings the savings line that the ROC curve should cross very close to the y-axis, making it difficult to realize significant value from an additional inspection step. Although the models presented in the previous section have considerably high AUC values, and in some cases intersect the break-even curve, we have not realized substantial savings based on the model for inspection value given in equation (5).

1) **Effect of Pre-existing Inspections**: Another challenge in this particular case is the pre-existence of other established inspection steps. In this particular case, there is an automated inspection station prior to the station that was the focus of our work. This station is an optical inspection station. The
pins, holes, fixation locations etc. on the units are inspected and compared to reference values in the specifications to detect possible deviations. This inspection already has a conservative threshold to prevent defects from advancing down the line. Therefore, our task is to improve an already conservative threshold to prevent defects from advancing detect possible deviations. This inspection already has a removed from the line before they have a chance to exhibit downstream failure. This is difficult because, in the course of normal production, failed parts are removed from the line before they have a chance to exhibit downstream failure.

IV. Discussion

In this paper, we have developed a framework to estimate the value of an assembly line inspection process as a function of the performance of a predictive model. This is accomplished by representing the value of the inspection as a function of the predictive model performance and on-site root-cause analysis. This framework shows that savings can be achieved either by improved root-cause analysis, or by increasing true positive rate and/or decreasing the false positive rate. We highlighted the sensitivity of this framework to intrinsic properties of a process such as the defect rate and the savings factor of an inspection (the cost saved by removing a defective part from the process). We have shown that, to realize savings from an inspection process, the predictive models have to have extremely low false positive rates in the presence of low savings factor and/or low defect rate. This is the case for most modern manufacturing processes. In addition, it is important to recognize the difficulty of estimating the savings factor in the case of end-of-line inspections. Failures that escape the plant, and are realized either by an OEM or end-user, can incur significant costs to manufacturers through warranty claims and/or recalls. Hence, predictive models may have potential to generate significant value where high costs are associated with the risk of failures experienced by the customer. There is a need to increase research efforts in this area given the intrinsic, extreme class imbalance of the application domain and the recent increased interest in applying machine learning techniques to manufacturing data (e.g. the Industry 4.0 initiative in Europe and the advanced manufacturing and industrial internet initiatives in the USA).

In Section III, we have presented a case study, highlighting a process that has both low savings factor and low defect rate. We have used machine learning tools such as RUSBoost, and Neural Networks as predictive models and shown promising results, but even with relatively high AUC values ($\geq 0.8$), were unable to cross the break-even line at high true positive rates. In future work, we plan to use machine learning tools to assist process engineers with root-cause analysis. We also plan to examine the ROC curves of existing inspections, to look for opportunities to optimize their current performance, as opposed to inserting an entirely new inspection step. Despite these challenges, we view automated assembly line inspection as fertile ground for predictive modeling, and we expect the application of cost-sensitive analysis, such as the model for inspection value described above, to be an important tool in this area going forward.

Acknowledgments

The authors would like to thank Gabriela Dominguez, Michael Hasenbalg, Cesar Gabriel Galvan Rubio, and Arnoldo Saenz from the Bosch Juarez Plant, for providing data, and domain expertise, and business understanding in the activities related to the case study portion of this paper.

References


