A Dynamic Threshold Based Approach for Detecting the Test Limits

Cristina Landin[†], Jie Liu[*§], Sahar Tahvili[* ‡]
[*]Product Development Unit, Cloud RAN, Integration and Test, Ericsson AB, Stockholm, Sweden
{sahar.tahvili, anna.a.liu}@ericsson.com
[†]School of Science and Technology, Örebro University, Örebro, Sweden
cristina.landin@oru.se
[‡]Mälardalen University, Product Realization, School of Innovation, Design and Engineering, Eskilstuna, Sweden
[§]Technical University of Berlin, Germany

***Abstract*—Finding a balance between meeting the testing goals and testing resources is always a challenging task. Therefore, employing Machine Learning (ML) techniques for test optimization purposes has received a great deal of attention. However, utilizing ML techniques requires frequently large volumes of data to obtain reliable results. Since the data gathering is hard and also expensive, reducing unnecessary failure or retest in a testing process might end up minimizing the testing resources. Final test yield is a proper performance metric to measure the potential risks influencing certain failure rates. Typically, production determines the yield's minimum threshold based on an empirical value given by the subject matter experts. However, those thresholds cannot monitor the yield's fluctuations beyond the acceptable thresholds, which might cause potential failures in consecutive tests. Furthermore, defining the empirical thresholds as either too tight or too loose in production is one of the main causes of yield dropping in the testing process. In this paper, we propose an ML-based solution that detects the divergent yield points based on the prediction and raises a flag depending on the yield class to the testers when a divergent point is above a data-driven threshold. This flexibility enables engineers to have a quantifiable tool to measure to what extend the different changes in the production process are affecting the product performance and execute actions before they occur. The feasibility of the proposed solution is studied by an empirical evaluation, which has been performed on a Telecom use-case at Ericsson in Sweden and tested in two of the latest radio technologies, 4G and 5G.**

Keywords–Software Testing; Test Optimization; Machine Learning; Regression Analysis; Imbalanced Learning

I. INTRODUCTION

Test optimization is of vital importance for the industry to get better products in quality and also affordability. As the Internet of Things (IoT) is becoming a reality, the demand for faster and cheaper products is increasing. To stay competitive in the market, Telecommunication companies apart to ensure the coverage and the quality of the emerging technologies, also need to optimize the form these products are being tested by reducing waste in the manufacturing process. In 5G (fifth generation) radio technology, the need for faster response, communication speed, capacity, and the number of features has increased remarkably compared to older radio generations. As consequence, the number of tests the new products need to comply with has increased exponentially. Therefore, those innovations demand new optimization methods that need to be applicable to the industry. Data-driven approaches have been shown to be useful in order to predict future trends of continuous data [1]. These trends can be extended to have dynamic thresholds, which give a more realistic approach to the behavior of future points than the currently utilized fixed thresholds. However, fixed thresholds do not give information of behavioral changes of the units tested nor their direction as long as they are within the predefined limits [2]. A method to measure the effectiveness of a production process is to measure its production yield. The yield is one of the major factors directly influencing the manufacturing operational costs [3]. The traditional definition of yield states that yield is proportional to the tested items, which comply with the test specifications or fixed thresholds. As the yield evolves according to the products are being tested, it follows a trend that can be modeled. This model can be useful in fault-localization and fault-prediction by finding the points where the yield diverges from normal production patterns. Furthermore, this model can identify low yield sources at a much earlier production stage compared to current practices and execute preventive actions. Our vision is to use a historical data approach to propose an intelligent framework that defines data-driven thresholds based on the yield predictions and finds abnormal points, thus optimize the test process for future radio generations. Though, this solution work with any kind of product can also be applied as a quality indicator for software testing factories for the similarities of the whole process, as test cases, test suite, and yield. This paper compares different regression methods, after data pre-processing, to predict the final test yield, define dynamic thresholds, and thereby detect the divergences of the characterized trends. Thereafter, the auto labeling process is added to label automatically the data inputs into the following three main labels: Pass, Warning, and Fail by using the Support Vector Machine (SVM) in the yield classification. The new thresholds give insightful information for the execution of future tests and the automatic labeling might also reduce the amount of manual effort in the yield loss analysis. These new thresholds can also be employed to enhance the yield by facilitating the fixed thresholds since they are often determined based on previous experience or by defining some stricter fixed test limits to ensure compliance with the regulators.

TABLE I. TEST REQUIREMENTS EXAMPLE IN THE DIFFERENT STAGES OF TESTING.

Test Requirement according 3GPP	Test Case	Test point
6.6.2 of 3GPP TS36.141 ACLR upper limit $44.2dBc$	Test Case 1: This test case will measure the Adjacent channel leakage power ratio (ACLR) of product A $Configuration$ $Procedure$ $Passcriteria > 45dBc$	$Configuration$ Test point 1.1: Send the right settings to the product Test point 1.2: Set up the carrier Test point 1.3: Send the right settings to the instrument to start measuring the ACLR $Procedure$ Test point 1.4: Measure the ACLR $Passcriteria$ Test point 1.5: Compare the results to the pass criteria

The proposed solution in this paper has a low computational complexity because it is designed to work in an online environment, despite the limitations of the infrastructure. In fact, the chosen ML-based methods have low computational cost, and the complete model is tested using an offline data set, typical of batch production and not access to Cloud compatibility. Moreover, the proposed solution in this study can handle high-dimensional input parameters, therefore it easily can be adapted and utilized in any other domain, e.g., sensors outputs and weather forecast. In order to make the proposed approach more generic and also confirming that transfer learning can be utilized, we trained the model on the 4G data set, and later we tested it on a 5G production data set. Furthermore, the obtained results in this study show a good harmony between the predicted points and the ground truth (the labeled data).

This paper is organized as follows, Section II explains all the theoretical background necessary to understand our approach. Section III the authors aim to compare prior results of similar solutions in close areas of expertise. Section IV explains in a detailed manner each step of the proposed solution to our problem that also can be applied to other research areas. Section V explains more about our data set, characteristics, properties, and types of inputs. The results after applying our solution approach to the given data set: prediction, classification, and validation methods are illustrated in Section VI. Finally, Section VII discusses the limitations, the assumptions, and the great potential of our approach and Section VIII concluded by briefly summarizing the study, results, and the direction of future research.

II. BACKGROUND

This section provides the required academic and industrial concepts and information for understanding the proposed solution in this paper.

A. Testing process of Radio Base Station (RBS)

Radio Base Stations (RBSs) are radio transceivers produced at Ericsson and contain analog and digital components.

To guarantee product quality and coverage, Ericsson follows international Telecommunication standard regulators, e.g., 3GPP for Europe and FCC in North America. The standards are translated into technical documents called test requirements. Each RBS generation has its own test requirements that follow the international regulators. The test requirements contain test cases where each test case is divided into several test points and all together make a test suite. The reason for distributing a test case into several test points is the limitations of the product's internal components, the production infrastructures, and also the modularity search. In order to get a better understanding of the analogy between standards, test cases, and test points, Table I provides a hint on how a standard is translated to small units. As we can see in Table I the 3GPP TS36.141 standard requires measuring the power in the adjacent channels of the main transmitting carrier, also Adjacent Channel Leakage Ratio (ACLR). Later, the design team has translated this requirement into the Test Case 1 where it is divided into 5 test points according to the infrastructure limitations (see Table I).

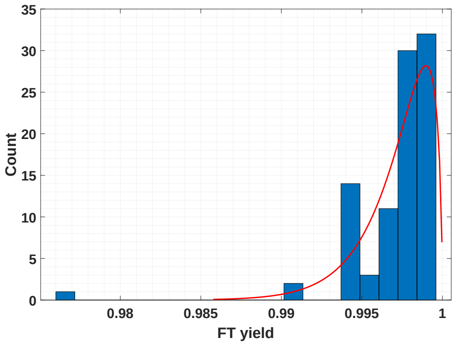
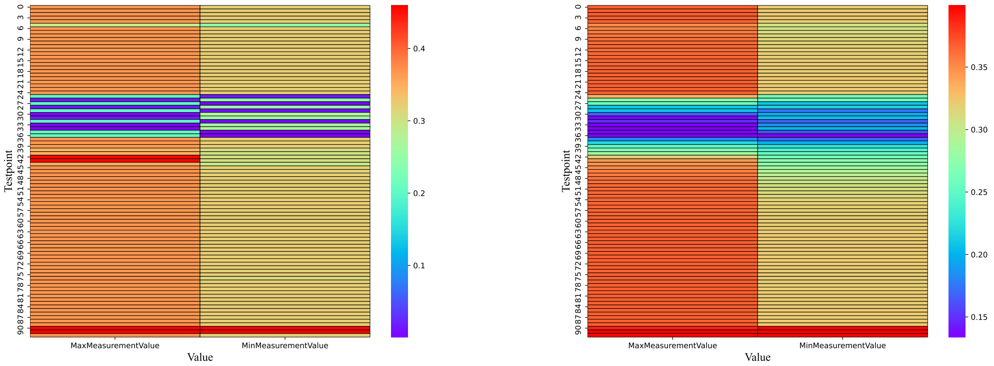


Figure 1. The 4G RBS production final yield (FY) distribution. The FY aims to reach 1 (100%) viz. It is a negatively skewed.

A test case typically tests a specific task to validate certain principles stated in the test requirement and it usually requires input and provides an expected output. The input describes the settings of the products while the output must follow the fixed thresholds given by the test requirements. On the other hand, a test point provides execution details of the different points, which are described in the test case. The test points can be executed in a different sequence, therefore a test suite (which includes several test cases and thereby test points) can be executed in sequential or parallel mode. The sequential execution mode can be beneficial if there is a dependency between test cases, i.e., the success of a test point depends on the success of the previous test points. On the contrary, the parallel execution mode can be useful when test cases are independent. In this paper, we assume that the test cases are independent of each other and can be executed in any order. The analysis of dependency between test cases is outside the scope of this paper and is carried out in an extension paper.

Moreover, in a fixed threshold, the output of the test points



(a) Captured data before smoothing.

(b) Captured data after smoothing.

Figure 2. Heatmap of feature inputs before after smoothing, for a 4G radio product.

depends on compliance with those thresholds. They can be given in form of range (low and high), greater than, equal to, or lower than - of the fixed thresholds, for instance, the Pass criteria $> 45dBc$ in Table I. Furthermore, in this study, only the test points with thresholds within a range are analyzed. However, in any case of having only one threshold, the proposed solution is still valid by assigning both boundaries to the same value.

A production yield is a performance indicator of a product over time, which measures how efficient a manufacturing process is. It also shows how the different changes, e.g., hardware, software, new components -revisions, influence the production process. Figure 1 mirrors the production yield distribution of the data set utilized in this study. As we can see in Figure 1, the production yield in our data set is left-skewed and does not show the Gaussian distribution. The final test yield (FY) is used through this paper, the final test yield is the percentage of good products produced taking into account the reworked products, unlike the First-Time Yield (FTY), which does not consider them. The FY formal definition is given in (2). The FY is chosen as an important feature in the production process of RBS at Ericsson because it gives an insight on how efficient the product is in a determined time slot and can be modeled to predict future patterns based on historical data by using regression methods.

B. Data Smoothing

Data smoothing is a statistical method for eliminating outliers from data to make the important patterns more visible [1]. Another purpose of using smoothing algorithms is to minimize statistical noise from the data set and assist prediction patterns. Some of the most known methods used for data smoothing are the Random method, simple moving average, random walk, simple exponential, and exponential moving average. In this paper, we focus on a single exponential moving average that applies weights to historical data. Those weights make the model focus on the most recent data observations. Therefore, the exponential moving average is more sensitive to the changes if compared to the moving average smoothing method.

Figure 2a shows the heatmap of the original input data before smoothing and Figure 2b after applying exponential moving average. Both figures have the same pattern, though the smoothed version makes the pattern more noticeable.

C. Regression Analysis

In the integration testing level, each test point is a continuous and dependent variable for different independent test configurations. Therefore, the approach to study the FY based on the test results of test points can be considered as a regression problem. The Regression models are being applied for predicting different purposes, they also measure the relationship between the input features and target data. This relationship can be linear or non-linear. There are several kinds of regression models, wherein this paper, linear regression, ridge regression, polynomial regression, and XGboost are applied and compared to each other in order to predict the production yield of RBS.

D. Imbalanced Classification

Once the production yields are predicted (by using the best regression model), then the predicted results need to be classified. The obtained investigations in the domain indicate that this classification suffers from an imbalanced dataset [1]. Therefore, the classification step of the proposed solution can be considered as an imbalanced classification, which is a typical problem in industrial applications. The imbalanced data set refers to a data set that has more labels in one class than the other classes, which makes it difficult to generalize the model. In fact, the problem arises when the important classification lies on the minority represented class. This issue may cause that one class dominates the other classes and that machine learning algorithms have poor performance on the minority class. Furthermore, imbalanced classification has shown to be challenging due to the severely skewed class distribution and also misclassification [1].

Generally, there are two main solutions for imbalanced classification: 1−employing the classification algorithm, which can handle imbalanced data such as IFROWANN (Imbalanced Fuzzy-Rough Ordered Weighted Average Nearest Neighbor Classification [1]) and 2−utilizing some data pre-processing methods to mitigate the imbalanced data sets such as random sampling (in forms of under and oversampling). The random under-sampling randomly removes samples of the over-represented class to match the minority class, while oversampling generates new samples of the under-represented class to match the majority class. However, in the oversampling, since the minority class does not add new information, instead, new samples should be synthesized from existing samples. SMOTE or Synthetic Minority Oversampling Technique, is an over-sampling method that uses the k-Nearest Neighbors (k-NN) to create a synthetic new sample [4]. In fact, the SMOTE model uses the Euclidean distance to calculate every minor point to get the k-nearest points. According to the imbalanced data set X_{origin}, select randomly n sample of the minority class, which will help us to pick up the nearest samples of X and name them X_i. Then randomly generate new samples of the minor class based on these X_{new} as represented in (1), where $i = 1, 2, ...n$. The ratio to generate new samples is $1/IR - 1$, where the Imbalance Ratio (IR) is defined as the ratio of the number of minor class samples to the number of major class samples as stated in [5].

$$X_{new} = X_{origin} + \text{rand}(0, 1) * |X_{origin} - X_i| \tag{1}$$

In this paper, the SMOTE model is used to improve the imbalanced data set for the classification and auto labeling process. On the other hand, SVM is a supervised machine learning algorithm, is used for binary classification by its kernel function, which could transform the data and classify different groups. This margin can be described as a line for two-dimension data and a hyper-plane for multi-dimension data. However, SVM can also be employed for the multi-classification problems by building different SVM models for every two labels. Especially, SVM works more effectively in high dimensional spaces where the feature dimensions are larger than the number of samples. For instance, the tool *One Vs One Classifier* [6] can separate the multiple classes or labels classification task that uses one classifier per class or label, i.e., it breaks down the problem into different binary classifications. In this paper, SVM is used to classify the different levels of acceptance and warning in the monitoring process done in production.

E. Anomaly detection and Fault Prediction

In a complex communication system, such as the production of RBS for 5G and 4G, the production data are collected in form of time series data. Due to the inherent complexity of the test production process of RBSs, the test time must be efficiently minimized, to apply traditional fault diagnosis is limited because it repairs after the fault occurs. In contrast, prognostic health management [7] provides a promising application in production where the variable time is of vital importance. Figure 3 illustrates a block diagram of a data-driven prognostic health management. It monitors relevant data from the process, e.g., production yield, analyzes them, triggers alarms or warnings, and takes actions before the fault appears. Furthermore, by adding failure predictors, it can forecast the fault occurrences [8] before they happened, and then preventing actions can be executed in the monitored system. There are also some other data-driven approaches that can contribute to prognostic health management [2] where the faulty behavior can only be seen when a huge amount of data is viewed in multidimensional space.

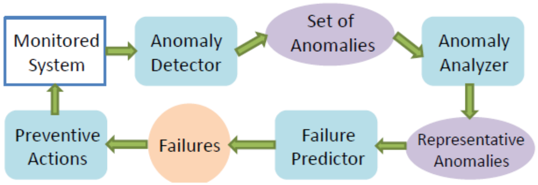


Figure 3. Data-driven prognostic health management (see [7]).

F. Transfer Learning

Many machine learning methods perform better under the assumption that the training and test data have the same distribution or have the same feature space. However, if the mentioned variables change, the algorithms might not perform properly and the methods need to be adjusted based on the changes and further data gathering might be required to update the models. Transfer learning or knowledge transfer can be considered as a potential solution to this problem. In transfer learning, the knowledge obtained in one domain needs to be transferred and applied in another domain. Conversely, another application can be able to train the models using data set from a domain where one has sufficient data and to use the same model in another domain where the data is limited. Transfer learning techniques have been applied to many real-world applications, which show promising results. One of the assumptions of transfer learning is that the source and target domains are related, which otherwise opens the possibility to negative transfer [9]. In this paper, transfer learning is employed by reusing the models found using a 4G (mature product) data set in a 5G radio product. We need to consider that, although 4G and 5G are two different radio generations, however, both products share some similarities.

III. Related work

In many industrial applications, test cases' limits are still defined by the test requirements as fixed thresholds, and not data-driven modeling of these thresholds is used to optimize the total testing time. Furthermore, there is not enough follow-up of the sources of yield losses. The main goals of this paper are to design a dynamic monitoring tool that supervises critical variables, predicts normal patterns, and sends warning messages to the user when anomalies are observed. One of the methods commonly used in industry is based on sample test measurements such as Process capability index (CPK). CPK is entirely done in offline mode and assumes

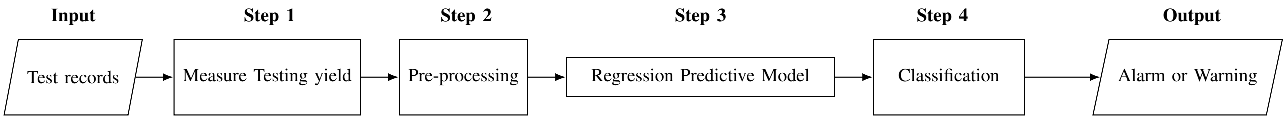


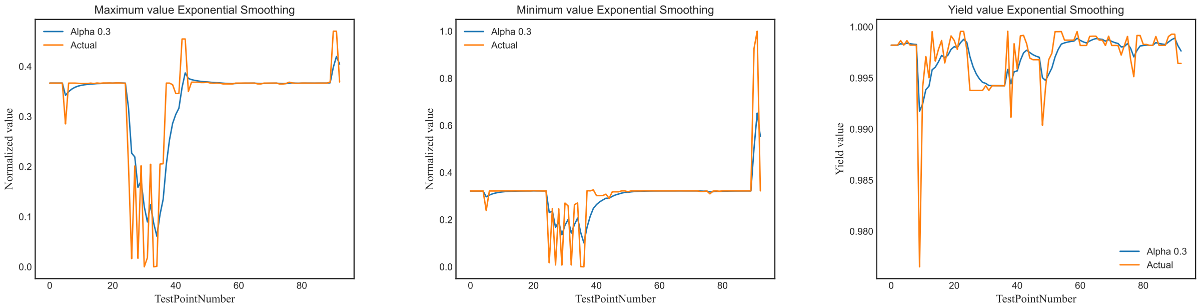
Figure 4. The required input, steps and expected output of the proposed solution in this study.

that the process is consistent over time. Though CPK has worked well in the past in specific samples, it does not give enough information to understand the whole process. For instance, it may perhaps represent only one side of the process when the data distribution is not centered within the specific thresholds [10]. Likewise, most of the methods for anomaly detection found in the literature are based on ruled-based methods, statistic approaches, or a combination of both [11]. Regarding anomaly detection in linear dynamic processes for simple inputs, Cho et al. in [12] studied the behavior of a gas regulator using Multiple output Gaussian Process Regression (MO-GPR) and Extreme Value Theory (EVT) to predict the output and directions of the anomalies. The real-time acquisition and updating of the coefficient are left as future work. Similarly, Chang in [13] uses linear regression to predict the anomalies in mine earthquake and update the counts above a certain threshold in real-time because of the importance of the application. Nevertheless, this method is designed to work online, the author does not use the advantages of machine learning and still uses a fixed threshold, making this solution hard-coded to work with only this application. On the other hand, for multiple inputs, Pang et al. [14] find the anomalies of multiple sensors using Multiple output Gaussian Process (MOGP) and Square Prediction Error (SPE) to find the anomaly score in real monitoring series. In his study, MOGP shows better results than PCA for dimension reduction giving more flexibility and adaptability in the findings of anomalies that otherwise need to be labeled by domain experts or by using fixed thresholds. Those approaches assumed the data set follows a Gaussian distribution, which is not always the case for another kind of application such as the analysis of multi-modal yield distributions. The production yield usually has heavy-tailed distribution as shown in Figure 1. On the data-driven anomaly detection, Chae [15] uses a statistical analysis-based Anomaly-based Detection System (ADS) to set an appropriate anomaly threshold in dynamic environments such as distributed systems. The difficulties the authors faced are multiple due to the inherent problem of dynamic environments and not further comparison between their method and the existing algorithms are explained in the paper. The same authors in [16] find the adaptive thresholds in trust-based detection systems where the anomalies come from known attacks and not ’smart attackers’ showing the difficulties this model suffers to adapt to a broader field where there are abrupt changing conditions. Regarding anomaly prediction, Chen et al. [2] study the prediction of system-level test (SLT) failures on system-on-chip (SoC) products where their analog circuits provide space to search faulty behavior by analyzing the outliers. A chip fails when any measured parameter falls outside its specifications, i.e., fixed threshold. However, this is not enough because they might be data points (parameters) that are far away from the nominal values but still comply with the specifications. This approach is the closest we could find to our application due to the similarity of the products, though this approach looks promising, it can not find the root cause of failures. In our case, the yield value of the different test points is analyzed, then whenever we see an abrupt change that does not follow the nominal trend, it will be easy to identify which test point is the source of failure in the whole process. Respecting test yield prediction using machine learning methods, many studies have been done in the last years. Jiang et. al [3] developed a data analysis tool for semiconductor manufacturing that predicts the final test yield in the early stages of production, hence improve the operational efficiency and reduce the production costs. The framework uses Gaussian mixture models (GMM) to identify and cluster the FY, Encoders to manage the difference on categorical or numerical inputs and does not need knowledge of the previous low yield root cause. Furthermore, this paper tries to find the root cause of low yield using the Gini importance. The problem with this approach is that their solution does not take into account the passed values of the important features and does not give importance to how the fixed thresholds can affect the FY. Based on our extensive survey, there are limited studies for FY-related problems in the production of RBS. In general, there are two major common difficulties for RBS FY prediction problems, which are high dimensional input data and complex process variations. For the sake of simplicity, we only use numerical data as inputs, and not feature reduction was used in our solution.

IV. Proposed Solution

This section provides our proposed solution for solving the initial problem stated in Section I. The overall FY for predicting and finding its dynamic thresholds flow framework is illustrated in Figure 4. As can be seen in Figure 4, the required inputs to the proposed solution are the test records, such as the test results recorded after the execution of each test point. As mentioned earlier, the main goal of this paper is to utilize historical test records to predict the normal FY by applying some regression models. The regression models are able to solve the problem with the continuous data, assist the finding of dynamic thresholds, and also the yield classification for optimization purposes. The following paragraphs provide more information regarding the mirrored steps in Figure 4:

- **Step 1. Measure Testing Yield**: the test results of each test point can mainly be divided into *Pass* or *Fail*. For employing the proposed solution in Figure 4, we utilize



(a) Maximum test inputs. (b) Minimum test inputs. (c) Target test inputs.

Figure 5. The original and smoothed versions of maximum, minimum, and target test inputs for product A.

the final test yield as (2):

$$FY = \frac{Pass\ units}{Total\ processed\ units} \tag{2}$$

Moreover, *Total processed units* is the total number of times that a unit has been tested, then $0 < Yield \leqslant 1$. The yield values closer to 1 mean that the product is mature and around 100% of the products that have been tested have passed. While yield values lower than 0.94 are considered as low yield in our application.

- **Step 2. Pre-processing:** in order to eliminate undesirable characteristics in the data (e.g. anomalies) and use the results values from different test points, we need to normalize the data. Since the measurement results are largely divergent, such as *Temperature* equal to $+50C$ or the *Current* equal to $10mA$, normalization of all the studied test records needs to be done before training the data or feeding it to the machine learning models. For instance, the minimum and maximum measurement values of each test point can be normalized by all the test points in a form of a matrix [17]. Furthermore, to detect the test points' results that are considered abnormal or do not fit any particular pattern, noise removal needed to be applied. Noise in this context contains the values outside the yield range, i.e., 0 to 1 and also the outliers. In our current data set, most test points have extremely good yields because they belong to a mature product. In this study, we utilize smoothing methods to remove the noise in our data set. The smoothing process is based on (3),

$$y_i = \alpha \cdot y_i + (1 - \alpha) \cdot y_{i-1} \tag{3}$$

where α is a smoothing factor that defines the forgetting rate of previous values. The lower α indicates the lower weights, which are applied to the true observed values. Moreover, y_{i-1} is the previous model value, and multiplying $(1-\alpha)$ is a solution for the recursive function to smooth the remaining data. We need to consider that α is between 0 and 1, which measures how much the true observation and previous model value influence the stability of data. Basically, the single exponential method here makes use of moving average in the exponential way to decrease the weights. The recursive behavior can be described as (4):

$$y_j = \sum_{i=1}^{j} \alpha(1-\alpha)^{j-i} y_i \tag{4}$$

where the hyper-parameter α is tuned by the grid search method to find the marginal point when the Root Mean Square Error (RMSE) starts to drop. Both, the input features and the target needed to be tuned individually. Therefore, the comparable best α for maximum value input, the one for minimum value input, and for the target all drop at $\alpha = 0.3$. Figure 5 shows the smoothing result for the test inputs (test points) and the target (final test yield) respectively.

- **Step 3. Regression Predicting Model:** as we can see in Figure 4, the output from Step 2 is a set of smoothed and normalized data. Moreover, the minimum and maximum measurement values from each test point execution are employed to build an FY model using regression methods. Our input data in Step 3 is represented as an array X. Each element of the array X has two columns: a minimum and a maximum of each test point. We need to consider that, the assumption here is that all row elements in X are independent:

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix} \tag{5}$$

and Y is the target value that represents the FY, where m is the number of test points.

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \tag{6}$$

Since the relationship between features and target is unknown, the following regression methods are applied in this paper in order to compare their performance. 1-Linear, 2-Polynomial, 3-Ridge, and 4-XGBoost regression. The mentioned regression methods are chosen due to their low complexity and computational efficiency, which easily can be adapted for solving industrial real cases. The linear regression models a linear relationship between two dependent and independent variables. It can also be modeled using simple linear approximation under the assumption if there is a linear relationship between variables. However, if this assumption is not entirely true and there is under-fitting, polynomial regression can be applied to model the non-linear relationship between inputs and target. In order to avoid the over-fitting problem proper of linear and polynomial regression, the Ridge regression utilizes regularization to punish the learning process to reduce the complexity. Extreme Grading boosting (XGBoost) is a type of Ensemble learning based on decision trees and can be used in regression prediction modeling by applying the advantages of regularization and weak learners. The XGBoost is known to be efficient and fast for prediction purposes. Later in this paper, the evaluation results of all the mentioned regression methods are compared using an industrial case study at Ericsson. The dynamic thresholds are found based on the regression models using three sigma and empirical approximations. This model provides feedback, which allows to update the constants of the formula, to avoid false anomaly detection.

- **Step 4. Classification of Imbalanced Data:** after performing the regression model for the prediction problem in step 3 for the FY measurement, the models are evaluated using RMSE, MAE, through comparing the prediction against the ground truth. In the utilized data set in this study, the best prediction is found for the XGboost model, which has a very low error rate and a much better prediction trend compare to other regression models. The results found using the XGboost model are then used to label the original data set, i.e., 0: Pass, 1: Warning, and 2: Stop. However, this auto-labeled data set is highly imbalanced. For instance, the imbalanced ratio (IR) between labels is 23.25. In order to balance the data set the SMOTE model is applied. The new balanced data set is used for the classification task. In a close consultancy with Subject Matter Expert (SME) at Ericsson, we classified all test points into three main classes: *Pass*, *Warning* and *Stop* using the SVM model. The number of classes is flexible and can be adapted based on the different optimization applications.

  Note that the model tends to find the best classification lines using the linear kernel function. The binary classification models can be seen as logistic regression but the SVM model does not support multi-class classification naturally and require meta-strategies.

Since the main goal of this study is to monitor the production of RBS and alarm the operator or the system manager for abnormal yield risks in advance, the outputs of the proposed solution in Figure 4 can be considered in form of different high-level applications. For instance, the proposed solution can be used as an alarm signal, a pop-up window, or a flag in a more advanced software of the testing process.

V. Industrial Case Study

In order to get a better understanding of the proposed solution in this study, an industrial case study is designed using an ongoing Telecom project at Ericsson AB, Sweden. The provided industrial case study in this work is following the proposed guidelines for conducting and reporting case study research in software engineering by Runeson and Höst [18] and specifically, the way guidelines are followed in [1] and [19].

The units of analysis in the case under study are test points, extracted from an internal database at Ericsson of a 4G RBS from now on called product A and a 5G RBS from now on called product B. The case study is performed in several steps.

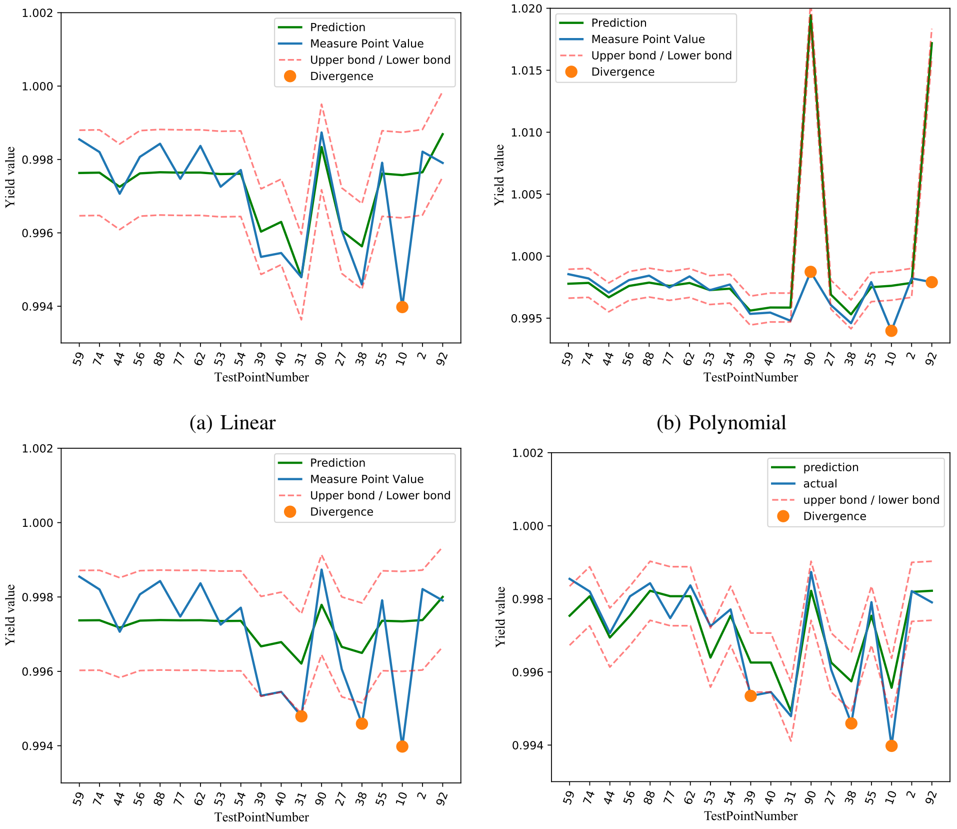
TABLE II. Detailed information of product A and B, 4G and 5G RBS respectively, for the conducted case study at Ericsson.

Product A		**Product B**	
Description	**Quantity**	**Description**	**Quantity**
Test units	1581	Test units	8
Test Points (Pass) with limits	2737	Test Points (Pass) with limits	12643
Test Points (Fail) with limits	165	Test Points (Fail) with limits	408
Test Point classification	1103	Test Point Labeling/classification	286
Test Points yield (0-1)	93	-	-

1) A total number of $5,018,925$ test records are captured from Ericsson's database for product A and 836174 for product B.
2) The captured test records include the following information *Test units* and *test points results (pass or fail)* where the quantity of them are summarized in Table II for product A and B respectively.
3) The final yield, FY, for each test point is measured using (2). Its noise and outliers are smoothed as shown in Figure 5c.

VI. Result

The obtained prediction results in this study are presented in Figure 6 for the linear, polynomial, Ridge, and XGBoost regression models. The upper and lower dynamic thresholds illustrated in Figure 6 are based on our prediction models. The X-axis represents the different test points (sub-parts of test cases) and the Y-axis is FY. The predictions follow the ground truth in most cases, however, the best prediction is found using the XGBoost. For the linear regression, the prediction suffers from the under-fitting problem and finds one divergent point, while in the polynomial there is an over-fitting problem for some points. As can be seen in Figure 6, the yield prediction goes above 1 which is not acceptable, and therefore found three divergent points, which are false alarms. For the Ridge regression, the prediction seems highly optimistic and slightly under-fitting, where it does not consider the divergences proper of the test process and consider them as real divergent points.



(a) Linear

(b) Polynomial

(c) Ridge

(d) XGBoost

Figure 6. The smoothed data utilized regression models on product A. The dashed lines are the predicted thresholds.

On the other hand, the XGBoost follows continuously the ground truth and finds also three divergent points which could mean anything according to the sensitivity of the application. In this case, all three divergent points can be classified as normal process behavior but their automatic discovery can save a lot of time for the engineers which otherwise will need to do this analysis manually. The classification model after auto labeling is evaluated by Receiver Operating Characteristic (ROC) curve from prediction scores. ROC curves represent the performance of the classification model. In order to get an optimum result, the iteration was done for a random state, which is a hyperparameter in the SMOTE method of K from 1 to 100 to get the best value to balance the data distribution and eliminate unnecessary noise then fix the random state value to get a consistent result. The best ROC score is 0.94 with K equal to 32. The optimized ROC curve is shown in Figure 7 for product A. The auto labeling process and the usage of the advantages of the SMOTE show an outstanding classification result.

A. Model Performance Evaluation

Besides the graphical results, the evaluation performance for regression predictive modeling is done using the RMSE and MAE, their results are displayed in Table III. The XGBoost model outperforms the other regression models as well, showing better results in both evaluation methods.

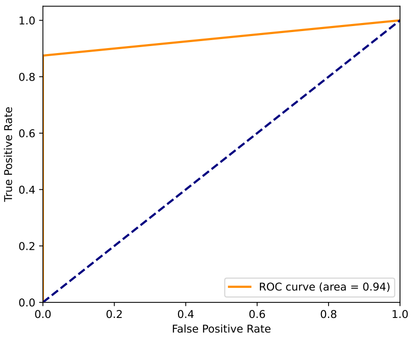


Figure 7. Classification evaluation using ROC for product A.

TABLE III. A SCORES SUMMARY OF THE SMOOTHED DATA VALIDATION.

Model Name	RMSE	MAE
Linear Regression	0.00099	0.067
Polynomial regression	0.0049	0.202
Ridge regression	0.0012	0.095
XGBoost	0.00073	0.00014

B. Model evaluation using unseen data

The problem of predictive modeling is to create models that have an acceptable performance making the predictions on new unseen data [20]. Therefore, the best model trained using product A data set is tested on product B to transfer the knowledge of the mature product and see if the model is still valid for another product with similar characteristics. The model prediction based on the XGBoost regression works for product B as well by detecting the yield divergences. Since product B is not as stable as product A and it is at the beginning of the production process, a minimum threshold of 0.94 of acceptance is necessary to be defined. Unlike product A, which does not have a definitive Fail, product B is labeled as follows: label *2* for yields lower than 0.94 - Fail, label *1* for divergence detected by the regression model- Warning, and label *0* as acceptable yield - Pass. After auto-labeling, this data set is a three-class classification task. The data set is obviously imbalanced. Referring to the solution of the imbalanced data set of product A, the SMOTE is implemented in the data set for product B by separating the data set into two classes at a time. The first one with labels *0* and *1*, the second with labels *0* and *2*, and then combining them. For the multi-label classification, the tool *OneVsRestClassifier* allows to build a classifier per class. For the unseen data set, we have three classes $0, 1$ and 2. The evaluation is based on the ROC curve which is demonstrated in Figure 8 for each class. The Macro ROC score is based on the average of each label's individual Precision and Recall, unlike the Micro ROC score which combines all three labels' recall and precision to do the average. The Macro score emphasizes more the small class label and the Micro score is the opposite because considers more the label with the larger class. Here both scores are comparably demonstrating that our labeled data set is well balanced and classified.

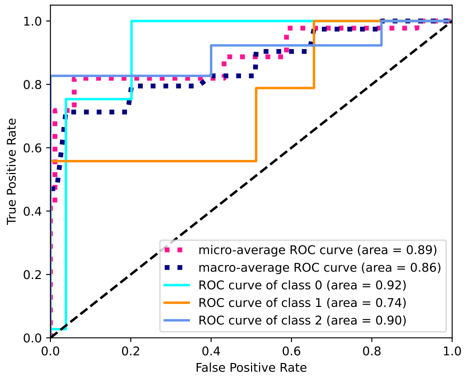


Figure 8. Classification evaluation ROC using unseen data captured from product B.

VII. Discussion

The main goal of this study is to design, implement and evaluate an ML-based solution that estimates the divergences points from dynamic thresholds based on yield predictions of two radio generations instead of using the traditional form to analyze the test process via fixed thresholds. To this end, we have made the following contributions:

- An ML-based approach is proposed for finding the dynamic thresholds for the final test yield (FY) with the purpose to develop a prognostic health management tool to work along the production process. The proposed approach has been implemented as a prototype in Python. It uses the XGBoost regression model to predict the test points' yield evolution and also the SVM model for classification of the predicted yield and automatic labeling of the input features.
- The evaluation of the proposed approach was performed using the test records of a 4G product at Ericsson. Furthermore, the risk of failure for the utilized product has been predicted by performing several regression models.
- The prediction error of the proposed regression models has been measured employing RMSE and MAE and for the classification, the ROC curve has been utilized.
- The proposed solution in this paper is applied to a set of unseen data, using test records of a 5G product. Moreover, the validation of the SVM model for the classification is showing good results. Considering the obtained result opens the possibility to transfer the knowledge learned in one product and use it in another product with similar properties. Furthermore, this approach can be used in an early stage of the testing process to find the largest sources of yield drop [3].

The yield prediction based on test points is modeled using normalized data inputs because the different kinds of testing processes, called test points in this paper, have different amplitude levels. For simplicity, all test points are assumed to have normal distributions and thereof are normalized. No further analysis is done in this paper on whether this normalization influences the final results. According to the reviewed stated of the art, this may differ depending on the application [21]. On the other hand, smoothing methods were implemented to remove the noise and the outliers in both inputs and targets before the prediction modeling was applied using several regression methods. Smoothing is a powerful technique uses in data analysis. Nevertheless applying smoothing in regression analysis to find divergences with respect to normal patterns can be very sensitive, especially when the smoothing process may remove important information one wishes to discover. Studies have been done regarding false positive reduction in networks using smoothing methods [22], the authors highlight the advantages of using smoothing as a method to averaging the unstructured false positives in anomaly detection, thus improve the accuracy. In this paper, we have also seen improvements in the prediction analysis after using smoothing instead of statistical approaches based on quartiles that are better applicable to variables with normal distributions. In this study, we assumed that test points are independent and the sequence of evaluation is not important, which is not always the case in different real-world applications. However, this assumption does not affect the prediction of the final yield because it is based on each test point and its respective evolution through time. The dependency between test points is outside of the scope of this paper but we consider it an important matter in the test optimization, therefore it is left

ISBN: 978-1-61208-894-5

as future work. An important variable in this study is the low computational complexity of the different machine learning methods because the monitor tool is planned to be used in an online mode. Although the whole modeling was done in offline mode due to the limitation of the data set, it is prepared to be implemented as part of a larger test program used at Ericsson production and can accept any kind of inputs, the normalization and removal of outliers are done automatically. The results of the execution of the monitoring tool in a production site are left as future work.

VIII. Conclusions

In this paper, we have introduced a novel framework to predict the final test yield, proposed new data-driven thresholds based on the predictions, found divergent points, and automatically labeled the results for two of the latest radio generations. This framework is generic and can be applied to any manufacturing process with continuous data sets. Besides, it is robust, scalable, and configurable to adapt to the sensitivity of the application. The pre-processing covers automatic noise and outliers removal by smoothing the inputs and target, inputs' normalization and solve the problem of imbalanced data sets for classification purposes. Four regression models were used successfully to model the historical trends of final test yield, whereof XGBoost showed better performance than linear, polynomial, and Ridge regressions. Firstly, divergent points were found using the prediction model and the dynamic thresholds. Secondly, automatic labeling of the prediction results was implemented using SVM. In our case labels: Pass, Warning, and Stop were relevant, however, the model can be scalable to many other cases where automatic labeling is needed. Hence, preventive actions can be executed before those divergences happen. These actions can be continuing execution with close monitoring or stop the production of one unit and continue with another one, instead of trying to pass the unit after many trials with a risk of suffering quality problems in the near future. Additionally, transfer learning has been briefly studied in this paper. The results show that it is possible to use the modeled trained in a 4G radio product and tested with excellent results in a 5G radio product, giving this approach some kind of generalization with a minimum amount of tuning. One pre-requisite is that the products have some kind of similarity to avoid a negative transfer. Future studies of this work can use the dynamic thresholds to update the test points' current thresholds to secure the values are in a safe region to guarantee acceptable final test yield. Besides, research the percentage of knowledge that is possible to transfer to future radio generations and still keep the quality of the product, thus reducing the amount of time consumed in the manual test optimization (which is still used in many manufacturing areas) will be in focus for future studies.

Acknowledgement

This work has been supported by the Swedish Knowledge Foundation (KKS) and VINNOVA through CoAIRob industrial research school and INDTECH research school (2020013201H) and also AIDOaRt project via the ECSEL Joint Undertaking (JU) under grant agreement No 101007350. The JU receives support from the European Union's Horizon 2020 research and innovation program and Sweden, Austria, Czech Republic, Finland, France, Italy, Spain.

References

[1] S. Tahvili, L. Hatvani, E. Ramentol, R. Pimentel, W. Afzal, and F. Herrera, "A novel methodology to classify test cases using natural language processing and imbalanced learning," *Engineering Applications of Artificial Intelligence*, vol. 95, pp. 1–13, August 2020.

[2] H. . Chen, R. Hsu, P. Yang, and J. Shyr, "Predicting system-level test and in-field customer failures using data mining," in *IEEE International Test Conference*, 2013.

[3] D. Jiang, W. Lin, and N. Raghavan, "A novel framework for semiconductor manufacturing final test yield classification using machine learning techniques," *IEEE Access*, vol. 8, pp. 197885–197895, 2020.

[4] N. Chawla, K. Bowyer, L. . Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, p. 321–357, June 2002.

[5] L. Ma and S. Fan, "Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests," *BMC Bioinformatics*, vol. 18, 03 2017.

[6] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] S. Jin, K. Chakrabarty, and Z. Zhang, "Anomaly-detection-based failure prediction in a core router system," in *International Conference on Advances in System Testing and Validation Lifecycle*, 2016.

[8] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, "Fault prediction under the microscope: A closer look into hpc systems," in *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–11, 2012.

[9] J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, no. 10, pp. 1345–1359, 2010.

[10] O. Khan, "A practical guide to utilizing cp and cpk," *Understanding Statistics for Quality by Design*, p. 1, 2015.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 1–58, July 2009.

[12] W. Cho, Y. Kim, and J. Park, "Hierarchical anomaly detection using a multioutput gaussian process," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 261–272, 2020.

[13] L. Li and J. Chang, "Real-time detection for anomaly data in microseismic monitoring system," in *2009 International Conference on Computational Intelligence and Natural Computing*, pp. 307–310, 2009.

[14] J. Pang, D. Liu, Y. Peng, and X. Peng, "Multiple-output-gaussian-process regression-based anomaly detection for multivariate monitoring series," in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pp. 326–332, 2018.

[15] Y. Chae, N. Katenka, and L. DiPippo, "An adaptive threshold method for anomaly-based intrusion detection systems," in *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pp. 1–4, 2019.

[16] Y. Chae, N. Katenka, and L. Dipippo, "Adaptive threshold selection for trust-based detection systems," in *IEEE 16th International Conference on Data Mining Workshops*, pp. 281–287, 2016.

[17] S. Tahvili, R. Pimentel, W. Afzal, M. Ahlberg, E. Fornander, and M. Bohlin, "sortes: A supportive tool for stochastic scheduling of manual integration test cases," *Journal of IEEE Access*, pp. 1–19, 2019.

[18] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, p. 131, 2008.

[19] S. Tahvili, *Multi-Criteria Optimization of System Integration Testing*. PhD thesis, Mälardalen University, December 2018.

[20] S. Tahvili, W. Afzal, M. Saadatmand, M. Bohlin, and S. Ameerjan, "Espret: A tool for execution time estimation of manual test cases," *Journal of Systems and Software*, vol. 161, pp. 1–43, September 2018.

[21] P. Sherman, "Tips for recognizing and transforming non-normal data." available at https://www.isixsigma.com/tools-templates/normality/tips-recognizing-and-transforming-non-normal-data/, Aug 2021.

[22] M. Grill, T. Pevný, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 43–57, 2017.