Automated Die Inking

Constantinos Xanthopoulos, *Student Member, IEEE*, Arnold Neckermann, Paulus List, Klaus-Peter Tschernay, Peter Sarson, *Senior Member, IEEE*, and Yiorgos Makris, *Senior Member, IEEE*

***Abstract*—Ensuring high reliability in modern integrated circuits (ICs) requires the employment of several die screening methodologies. One such technique, commonly referred to as die inking, aims to discard devices that are likely to fail, based on their proximity to known failed devices on the wafer. Die inking is traditionally performed manually by visually inspecting each manufactured wafer and thus it is very time-consuming. Towards reducing this cost, we introduce a novel machine learning-based methodology to learn and automatically generate the inking patterns from the failure maps, thus eliminating the need for human intervention. Effectiveness is demonstrated on an industrial set of manually inked wafers.**

***Index Terms*—Inking, die screening, automation.**

I. Introduction

As THE complexity of contemporary Integrated Circuits (ICs) and the volume of their deployment in reliability-stringent domains (e.g., automotive, health, aerospace, financial) increase, the need for more efficient and dependable testing solutions becomes paramount. To this end, several techniques have been introduced in all stages of the IC manufacturing process. While these techniques significantly improve manufacturability, testing, and production yield, the detection of latent defects remains a mounting challenge [1], both in terms of complexity and in terms of cost.

A common practice for identifying latent defects is burn-in testing, during which the chips are subjected to higher frequencies, voltages, and temperatures. The goal of these stress conditions is to accelerate the manifestation of any imminent, but latent defects. Although burn-in tests are effective in identifying the majority of such manufacturing imperfections, this incurs increased complexity of test-floor logistics and significant cost overhead [2], [3], which can be prohibitive for high-volume manufacturing. Aside from the cost, burn-in testing sometimes suffers from low coverage, meaning that the stress testing conditions do not exercise the entirety of an IC, resulting in defective devices escaping testing.

Manuscript received April 25, 2020; accepted May 8, 2020. Date of publication May 19, 2020; date of current version June 5, 2020. *(Corresponding author: Constantinos Xanthopoulos.)*

Constantinos Xanthopoulos and Yiorgos Makris are with the Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: constantinos.xanthopoulos@utdallas.edu; yiorgos.makris@utdallas.edu).

Arnold Neckermann, Paulus List, and Klaus-Peter Tschernay are with the Department of Product Engineering, amg AG, 8141 Premstätten, Austria.

Peter Sarson was with Automotive Solutions Division, ams AG, 8141 Premstätten, Austria.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TDMR.2020.2994291

To complement, or sometimes eliminate burn-in testing, depending on the reliability requirements of the product, several screening techniques [4], [5] are often used, each based on different criteria. The general premise behind these techniques is that any significant abnormality can be an indication of the presence of latent defects and, therefore, proactive screening is justified. Techniques such as Dynamic Part Average Testing (DPAT) [6] aim to identify the passing die that exhibit marginal test measurements relative to the main distribution of each wafer. A more static screening technique aims to filter out die based on their wafer location, commonly their distance to the edge of the wafer. This is based on the observation that die near the edge of a wafer are more likely to be defective and can be discarded even prior to probe testing. Another set of techniques aims to alter the test process according to characteristics of the silicon under test, in order to better identify outlier devices that are likely to fail during deployment. These techniques belong to a more broad category of techniques called adaptive testing [7], that can target other manufacturing goals, such as cost reduction, and yield improvement.

Manual inking, a popular screening technique based on the observation that manufacturing defects are spatially correlated on the wafer surface [8], is also being used to filter out the devices that are likely to exhibit latent defects. In other words, the assumption behind this common practice is that clusters of failing die on a wafer suggest a systematic local discrepancy that can lead to an early-life failure. Figure 1 shows the flow of the manual inking process as it is currently performed. For every newly manufactured wafer, once the specification testing is completed and wafer-level failure patterns have been generated, a reliability expert visually inspects the failure map of each wafer and manually marks any die locations that are likely to be defective. This decision is based on the proximity of each die to neighboring failed die, and the types of failure in that cluster. Figure 2 shows inking examples for the three most common failure pattern types, namely, edge, scratch, and blob type. Dark blue and red-colored die denote passing and failing die correspondingly, while the light blue colored die indicate the manual inked locations. Edge type patterns are indicated by clusters of failing die near the edge of the wafer. Scratch patterns are identified by streaks of failing die along the surface of the wafer and are usually caused by human error; therefore, they are far rarer than the other two types. Lastly, wafers exhibiting blob type failure patterns contain patches of failing die at several locations on the wafer but not solely on the edge. Although inking is a cost-effective solution, as compared to the expensive burn-in testing, it is nevertheless a manual process that requires human intervention. This entails

1530-4388 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

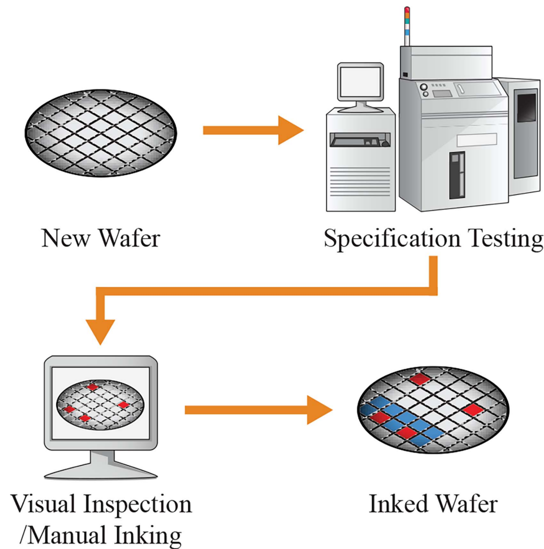


Fig. 1. Current inking process.

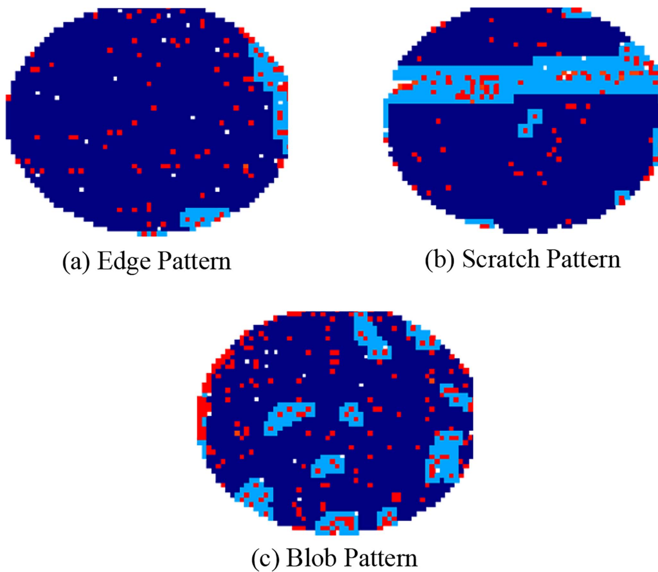


Fig. 2. Examples of each failure pattern types.

added time, cost, and often subjective discrepancies between inking decisions. As a result, manual inking is inconsistently performed between different engineers, and often with minor discrepancies between inspections by the same engineer.

In this work, we propose a pattern recognition-based approach for eliminating or limiting the degree of human intervention required for inking a manufactured wafer. Our method relies on the testing results (i.e., pass or fail) generated from Automatic Test Equipment (ATE), to predict the wafer locations that are likely to contain defective die. To achieve this, a number of features reflecting the defect proximity of each die is extracted, and a classification model is trained based on them. The dependent variable of the classification model originates from the manually inked failing die patterns. This machine learning-based approach, when compared to static solutions that use simple statistics and image processing techniques, has the advantage of being data-driven. Therefore, readily applicable with minimal modifications to all existent products where manual inking is currently performed. The proposed methodology follows the online-learning paradigm which compared to batch learning-based approaches, allows for a small initial dataset to be used for training and can also adapt to future inking policy changes without requiring a complete, thus time-consuming re-training of the model.

It is important to emphasize that all of the above device screening methods reduce the number of marketable devices, without failing any of their specifications tests. This naturally reduces the overall manufacturing yield but also posses a crucial trade-off between the screening aggressiveness and yield loss. When any of the methods above is performed overly aggressively, resulting in the screening of devices that are unlikely to contain any latent defects, yield loss becomes the primary driver of increased manufacturing cost. Especially in the case of manual inking, where the degree of inking is driven only by the customer quality incidents (CQI), product engineers are often incentivized to overly ink, thus significantly increasing the yield loss. In this work, our main focus was to automate the current manual process, and not to eliminate all of its drawbacks. Since our proposed method relies on the manual inking decisions to train the proposed classification model, it is consequently expected to result in an aggressive automated inking, when the manual inking was performed aggressively. To mitigate the effects of this phenomenon, our proposed methodology requires the training to be performed using a dataset of manually inked wafers, generated by multiple product engineers. This requirement will allow the model to extract a strategy that is closer to the consensus between the engineers, resulting in less aggressive inking of the wafers. Moreover, a knob is being proposed that enables adjustments of the inking degree without requiring complete retraining of the model.

The remainder of this paper is organized as follows. In Section II, we discuss in detail prior work on statistical-based and automated approaches to inking. In Section III, we introduce the proposed approach for automating the manual inking process. Experimental results that demonstrate the effectiveness of the proposed method using industrial data are presented in Section IV, and conclusions are drawn in Section V.

II. Prior Work

Identification of outlier devices with the goal of diminishing the number of test escapes or customer returns has been a longstanding topic of research. Various methods have been proposed that can be grouped into three main categories i) adaptive testing-based ii) statistical-based, and iii) spatial-based outlier screening.

A. Adaptive Testing-Based Outlier Screening

The first category of die-screening techniques relies on the alteration of the test process in order to identify devices that are likely to fail during deployment. Often a common trade-off is the associated cost when a large number of tests are being performed to identify devices that exhibit latent defects. In High Volume Manufacturing (HVM), such trade-off drives

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

the elimination of tests that have low die fallout rates. Several studies focus on the selection of the minimum number of tests capable of detecting defects while keeping the test cost minimal. In [9], [10], the authors demonstrated a technique for selecting tests targeting specific defect mechanisms based on the wafer-level e-test measurements or a list of mandatorily performed tests. Towards the same goals of cost reduction and test quality, the authors of [11] presented a machine learning-based methodology for adaptively selecting groups of probe tests according to the e-test signature of each tested wafer.

B. Statistical-Based Outlier Screening

The second category includes screening methods that take into account the distribution of specific die-level test measurements to decide whether a particular passing device is likely to contain latent defects. Dynamic Part Averaging Testing (DPAT) is a statistical methodology that was introduced as a guideline by the Automotive Electronics Council (AEC) [6] in order to improve the reliability of electronic products used in the automotive industry. In contrast to the traditionally applied testing in which a lower and upper limit for every test is defined and applied for all manufactured wafers, DPAT adapts the limits according to the test distribution of each wafer. In detail, once a wafer has completed wafer sort, the wafer-level distribution of all test measurements is known, and several statistics can be calculated. These statistics are subsequently used to identify any passing outliers which are then discarded or marked for further testing. In its purest form the DPAT limits are adjusted as follow:

$$DLL_i = Mean(x_i) - k \times Std(x_i)$$
$$DUL_i = Mean(x_i) + k \times Std(x_i)$$

where DLL and DUL stand for Dynamic Lower Limit and Dynamic Upper Limit respectively, k denotes the threshold parameter, and x_i the ith test measurements of a given wafer. Other variations of DPAT, use robust statistics [12] in order to eliminate the effects of outlier measurements. In [13] the authors presented the effectiveness of DPAT when performed at final testing with the use die tracing tools.

All the above techniques have the benefit of being automated, thus easily integrated into the production flow, as well as, of providing control over the level of screening performed, by adjusting their corresponding threshold parameters. Unfortunately, many practical limitations hinder the performance of the techniques mentioned above. These limitations include the multimodality of the test measurement distribution and the distribution shifts due to post-silicon calibration. Multimodality often occurs in high volume manufacturing (HVM) and can be best observed through wafermap plots where similarly-colored groups of die locations manifest. Those groups correspond to devices that have been affected similarly by one or more discrete processes. Examples of such processes are the multi-site testing where discrepancies in the measurement equipment between sites occur often, or due to reticle shot dissimilarities. If the effects of multimodality are not taken into account when applying DPAT, many statistics will be skewed, and as a result, the outlier screening will be invalid. Several techniques have been industrially adopted to address the issue of multimodality, that rely on the application of corresponding statistics on disjoint sets of die locations [14]. The selection of those sets is made in accordance with the source of multimodality. For example, if the only known source of multimodality is site-to-site variation, DPAT can be performed per site, thus obtaining robust statistics for calculating unique test limits for each site. On the other hand, when several multimodality sources affect the test distribution such that no disjoint sets of die can be determined a priori, the effectiveness of DPAT gets diminished. Post-silicon calibration also affects the underlying distribution of specific test measurements by tuning marginal devices within the adapted passing limits, possibly concealing potential latent defects. Despite the limitations above, statistical-based outlier screening methods are used universally but often are complemented by other methods, which we will describe below.

Other statistical-based die-screening techniques perform their analysis on a multivariate space instead of per-test basis. An example of such technique uses Principal Component Analysis (PCA) to identify the devices whose principal components are not sufficient to reconstruct the original test measurements [15]. In [16] the authors used PCA in combination with a Support Vector Machine (SVM), one-class-classifier to detect devices with latent defects, both preemptively as well as reactively after CQIs have occurred. Authors in [17] used a combination of the moving limits and correlation testing techniques in order to identify outliers when the ICs structurally similar blocks and whose tests are highly correlated.

C. Spatial Outlier Screening

In this category of screening techniques, the main criterion is the proximity and density of failures to a given die. Their utilization in production is to resolve some of the limitations statistical-based methods exhibit. The underlying premise for this family of methods is that die located close to a cluster of failing are more likely to contain latent defects, despite the value and relative position in the distribution of their test measurements. For this reason, spatial outlier screening complements the statistical-based methods and is often used complementarily.

Several attempts to automate the manual inking process have been made by extracting neighborhood failure statistics and using hard-coded rule-based decision algorithms. One such approach is the Good Die in a Bad Cluster (GDBC) [12] that inks a device as failing if the number of failed devices surrounding it exceeds a specific threshold. A variation of this is the Good Die in a Bad Cluster with Specific Bins (GDBC SB) [12] that takes into account the type of failure of each die as denoted by the assigned bin number. This means that each failure bin will be associated with a threshold, indicating the number of surrounding failed die of that bin are needed for a passing die to be inked. A more aggressive technique called Bad Bin in a Bad Cluster (BBBC) [12], inks the surrounding die of a cluster if the number of failed die in that cluster is above a given threshold. In [18] have proposed a different approach, called Nearest Neighbor Residuals (NNR), which

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

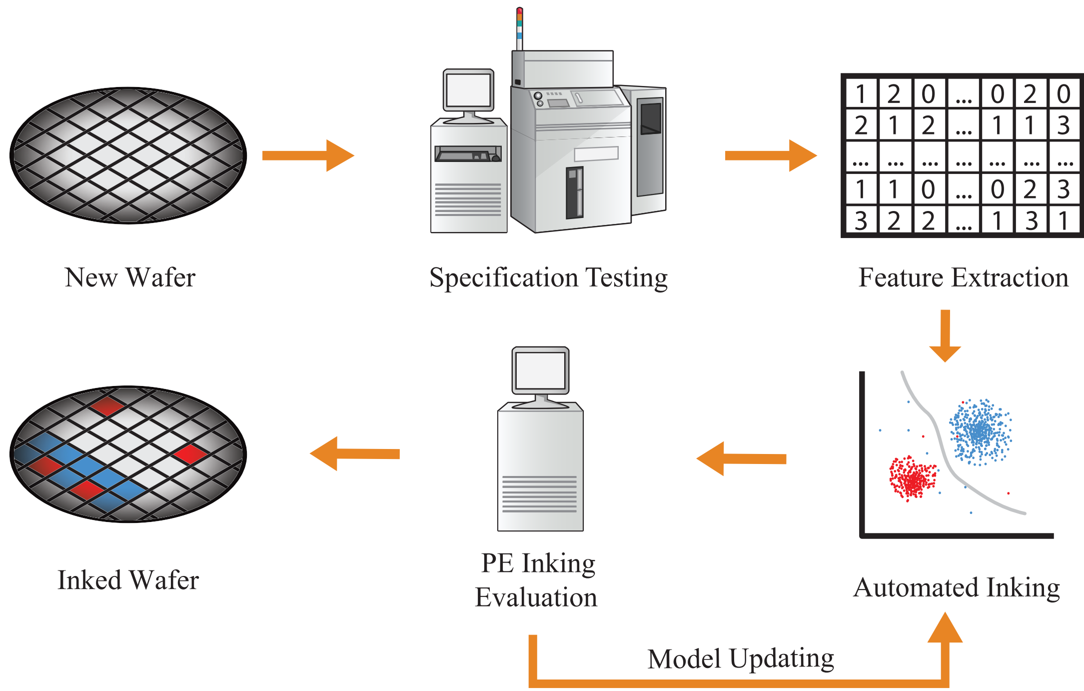


Fig. 3. Overview of the proposed on-line learning automated inking method.

performs a statistical analysis of tests after calculating residuals based on the die neighborhood. Variations of this technique have been used successfully in the industry to detect burn-in failures [19].

Although automation of the inking process can be achieved by developing a rule-based system that utilizes the failure maps as those described above, designing, maintaining and adapting such system to new products can be a mounting challenge. This system would require a series of rules to be defined and coded, avoiding conflicts, and taking into account several parameters such as the location, topology, density, and failure type of each failed die. Moreover, adaptation of these methodologies for existing products requires the reverse engineering of the current manual inking practices, to be reflected by the algorithms parameters.

III. Proposed Approach

To address the limitations of the rule-based spatial screening methods and automate the manual inking process, in this work, we introduce an online classification modeling based methodology. We seek to emulate the decision process that is usually performed manually by an engineer, using either previously inked wafers or by integrating the model learning in the production flow. Figure 3 shows an overview of the proposed approach. After specification testing, the failure maps are used to extract useful features which in turn, are utilized for training or predicting the inked die locations. Once these wafers have been automatically inked, an evaluation step for assessing the effectiveness of the automated inking system follows. During this step, product engineers (PEs) have the ability to visually inspect the inked wafer and adjust the predicted ink patterns by marking additional die or unmarking inked ones. Any potential corrections by the product engineers are then used to update the inking model. This online learning flow is closer to the current manual inking paradigm, which often includes the evaluation of the screening decisions by a committee of engineers. At the same time, it allows for model initialization through batch learning when a historical dataset is available. In other words, the online learning schema enables the training and improvement of the model to be performed either iteratively or by using previously collected datasets. At the same time, once the confidence in the model is high, inking becomes fully automated and does not require any intervention by the product engineers.

A. Feature Extraction

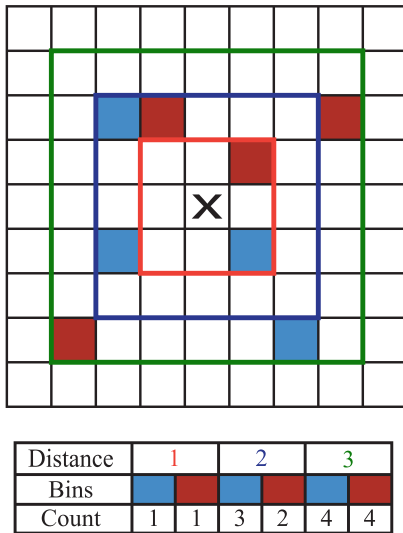
To simplify the test-floor logistics, one of the constraints for developing a solution was to rely only on the pass/fail decisions of the specification test and avoid the use of the actual test measurements. As a result of this, our classification model must be trained by features that can be extracted from the map of failed die locations for each wafer. These features must be selected so that they can portray the failing conditions in the neighborhood of each die. Similar to the spatial outlier screening methods GDBC and BBBC, to enhance the prediction accuracy of our model, rather relying only on the pass/fail status of each die, the different binning groups were also considered. Usually, the binning groups denote different types of defects or performance characteristics of each chip, and a different identification number, commonly termed bin number, gets assigned by the test program.

The feature extraction algorithm shown in Algorithm 1 works as follows. Since our goal is to capture the local failing conditions of each non-failing die, we calculate the number of failing neighbors for each bin number. This process is repeated for distances up to a predefined maximum neighborhood size.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Feature Extraction

```
1  for d in passing die do
2  |  for b in failing bins do
3  |  |  for k in max_k ; // max_k: The maximum
   |  |  considered distance for a
   |  |  neighborhood
4  |  |  do
5  |  |  |  features = GetNumberOfNeighbors(d, b, k);
   |  |  |  // Function that returns the
   |  |  |  number of neighbors within k
   |  |  |  distance of die d that belong
   |  |  |  to bin b
6  |  |  end
7  |  end
8  |  features += GetDistanceToEdge(d); // Function
   |  that returns the distance of die d
   |  to the nearest edge of the wafer
9  |  features += GetLotFailureRate(d); // Function
   |  that returns the failure rate of die
   |  d across the wafers in a lot
10 end
```



Distance	1		2		3	
Bins						
Count	1	1	3	2	4	4

Fig. 4. Feature extraction example for a maximum die distance of three.

In contrast to the static methodologies presented in Section II, the significance of each bin, does not have to be taken into account during the feature extraction since the importance of each feature will be evaluated by the model training process.

Figure 4 summarizes the failure density-based feature extraction for a 9×9 die segment of a wafer. Blue and red colored boxes correspond to failing die of different failure types, which would have been expressed by different bin numbers in the probe-test report. The feature extraction process counts the number of failures at certain distances away from the target die, as shown by the *count* row in Figure 4, for a maximum die distance of three. Generating these counts separately for each different bin number is essential as it allows the model to infer the significance of each failure type for the inking decision.

The implementation of the above-mentioned *GetNumberOfNeighbors* function was based on the K-Dimensional Trees [20] algorithm to avoid the high computational complexities of the brute force approach. This algorithm works by eliminating areas of the search space based on the already calculated distances, thus avoiding unnecessary calculations for very distant die locations. The parameter that is used for this elimination is the radius *(k)*, which in our implementation is iteratively increased until a predefined maximum distance is reached. With this technique, we aim to increase the granularity of the various levels of proximity to defective die, which is an essential factor in our application. In other words, extracting features for different neighborhood sizes, allows the model to weight failure density proximate to the target die higher compared to a more distant one. As it will be shown in Section IV, the evaluation of the maximum neighborhood distance is done by comparing the accuracy of the generated models while increasing the *max_k*.

As proposed in [21] and shown in line 8 of Algorithm 1, we append the distance of the die to the nearest edge of the wafer, in order to assist the modeling of the commonly occurring wafer edge defects. With this feature, the model is able to accentuate clusters of failures that are near the edge compared to ones closer to the center even if they consist of identical bins.

In this work, we propose the addition of one more feature, called lot-stack, that represents the failure sensitivity of a die location according to the failure rate across all wafers in the same lot. Once the internal loop for each failing bin is completed, and the corresponding number of failing neighbors has been determined, another feature is appended to the feature vector. This feature is simply the failure rate of each passing die location across the remaining wafers in the same lot, taking values within $[0, \frac{c_{x,y}}{N-1}]$, where $c_{x,y}$ is the number of failing die in the lot for die coordinates x, y and N is the number of wafers in a lot. A high intra-lot failure rate for a die location can indicate a systematic cause across wafers in the same lot. This is often the result of a flaw in one or more of the manufacturing procedures that are common for all wafers in a lot. This addition of a single feature allows the model to ink areas across all wafers in a lot, and it imitates the manual process performed by engineers that use the same metric in order to enhance their inking decisions. An alternative approach would be to ink the die locations in a post-prediction step by simply using a single hard-coded threshold, instead of relying on the model to utilize the lot-stack feature. The benefit of entrusting the model for making this decision is twofold. First, we can utilize existing historical inking data to help the model identify the appropriate threshold per product as well as to adjust it based on the future corrections made by the product engineers. The second benefit is that the final decision becomes a function of the remaining features allowing for a more dynamic threshold according to the location of the die (i.e., as the lot-stack feature interacts with the distance from the edge feature) or the

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

types of failures (i.e., interacting with the bin failure density features).

B. Classification Modeling

The goal of our proposed approach is to predict the inking decision which takes two states, namely, *ink* and *no-action*, for each non-failing die on a given wafer, based on the feature vector explained in the previous section. This task is best achieved by the utilization of a multivariate binary classifier, which goal is to learn a hypothesis $h(\mathbf{X})$ to better separate the two classes, as a function of the feature vector $\mathbf{X}$. The selected classifier, as well as the produced hypothesis, must satisfy several practical requirements.

- Given the nature of the extracted features, the decision boundary will most likely be non-linear; thus, the selected classifier must be capable of producing such boundaries.
- The classifier must support online learning to allow for seamless integration to current production flows.
- Prediction and model updating must be computationally efficient to minimize the time required for a wafer to be inked, especially for products whose wafers consist of thousands of devices.

Batch learning, which is the most common classification learning paradigm, requires the availability of multiple input-output pairs $(\mathbf{x_i}, y_i)$, to learn the separation boundary. This approach, although technically possible [21], is not suitable for automating the manual inking process in a production environment. This is due to the characteristics of the training data; specifically, that input data arrive continually as each wafer lot is being manufactured and tested. Another characteristic is that the output inking decision requires human feedback making the solution less fitting to the batch learning approach. The alternative approach, referred to as online learning, can process a single input-output pair and the size of the training population of pairs does not have to be known a priori. One of the main benefits of online learning is that it is incremental and does not require data to remain available after its initial utilization. In contrast, updating a batch learning-based model requires access to data from all previously utilized wafers along with data from the new wafers. Although storing the failure wafermaps is already a common industrial practice, the re-training of the model in the batch learning scheme would be prohibitively expensive, given the long time required to access large sets of data and re-run the feature extraction process. Another benefit of the batch learning scheme is the ability to revert to previously trained models, provided that before each model update, the model objects are serialized and stored. Several models [22] support online learning, some of which include Multilayer Perceptron [23], On-line Random Forests [24], Incremental Support Vector Machines [25], Naive Bayes [26], and Learn++ [27]. In this work, we use a Multilayer Perceptron (MLP) classifier with Stochastic Gradient Descent (SDG) [28], [29].

An MLP is a feed-forward artificial neural network consisting of at least three layers, namely an input, an output, and a hidden layer in between. The number of hidden layers and their connectivity affects the level of functional complexity the network is able to approximate. The hidden and output layers consist of neurons that implement a non-linear activation function, usually a sigmoid or unit step function. Each layer is connected to its neighboring ones through synapses, which are assigned weights and serve the purpose of adjusting the strength of the carrying signal. Due to the multiple layers and the use of non-linear activation functions, MLPs are universal approximators capable of learning non-linear separation boundaries when used for classification. Moreover, it is important to note that the architecture of the artificial neural-network described here might not be optimal for other devices and other wafer sizes, as the corresponding complexity of the inking strategy changes. To identify the proper architecture, an early set of wafers can be used to determine the various hyper-parameters (e.g., number of hidden layers) using well-known tuning methods. Alternatively, the proposed architecture could be used initially, and once a sufficient number of manually or semi-automated (automatically inked, but manually corrected) wafers have been collected, it can be updated, through a similar hyperparameter tuning process.

A multilayer perceptron is trained using backpropagation which allows the weights of all the layers to be adjusted by distributing the output error to all previous layers. Traditionally, during backpropagation batch gradient descent optimizers are used to adjust the weights of the neurons. Unfortunately, while the classic gradient descent algorithm is efficient for relatively small datasets, it has the added disadvantage of requiring all training samples in order to minimize the error. Alternatively, the Stochastic Gradient Descent (SGD) algorithm perturbs the weights at each iteration by taking into account a single training sample at a time. This property of SGD not only allows training of the MLP when large training datasets are used but also enables the transition from batch learning to the online learning paradigm.

C. Post-Prediction Processing

Certain operations are better performed after prediction that enhance the automated inking. In this work, we propose two such post-prediction techniques, namely inking degree tuning and scratch correction.

1) Inking Degree Tuning: Due to the subjectivity of manual inking, models have to distinguish the noise (i.e., overly aggressively inked die locations) from the correctly inked locations. Although failure density is the primary criterion that drives the inking decision, in the interest of time product engineers often use inking tools with regular shape brushes (e.g., square or circle shaped), instead of inking one die at a time. Based on the above, the automated inking model usually performs a less conservative inking, marking fewer die locations compared to the manual approach. Although this is desirable in most cases, as it reduces unnecessary yield loss, sometimes a more aggressive inking is preferable. To enable such post-prediction calibration, an image-processing-based step was introduced in [21]. During that step, the size of the automatically inked areas was reduced or increased by a pre-determined and hard-coded degree.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

In this work, we propose an alternative approach for post-prediction tuning of the inking result based on the confidence estimation of the inking prediction. In a multilayer perceptron, the activation value of the output layer can be used as a reliable prediction confidence metric [30]. This technique allows for a dynamic adaptation of the degree of inking performed by the proposed methodology by adjusting the decision threshold (t). If p_i denotes the probability estimate for die i to be inked, the default classification is performed by evaluating the $[p_i \geq 0.5]$ Iverson bracket for all die locations. By generalizing the above to $[p_i \geq t]$, we can adjust the degree of inking by increasing or reducing $t \in [0, 1]$. When $t \in [0, 0.5)$ a more aggressive inking is performed. On the other hand, when $t \in (0.5, 1]$ the prediction is less aggressively adjusted, thereby resulting in fewer inked die locations.

2) Prediction Confidence Monitoring: Another advantage of using the probability estimations instead of the binary inking decision is to allow for the integration of a monitoring routine. Such a routine can be used during production to notify the product engineers that the inking model has low confidence for a provided input failure map, thus initiate a manual review of the input pattern and the automated inking decision. Such automated monitoring capability is essential in an industrial setting as it eliminates the need for constant supervision. This is especially the case during the mature phase of the model in our proposed automated inking solution, where no human intervention is needed, and all inking is performed automatically.

To achieve this, the monitoring routine has to calculate wafer-level statistics based on the distribution of the probability estimates. For example, a routine that periodically checks newly inked wafers can evaluate the Brier Score (BS) [31] of the probability estimates (p_i). When BS is above a predefined monitoring threshold t_m, a notification can be sent to the product engineers to evaluate the wafer manually and, if needed, provide the model with the corrected inked wafermap in order to update its weights.

3) Scratch/Line Correction: The main strength of the proposed machine learning-based methodology is the identification of blob and edge-type clusters on the wafer, that are more likely to be inked. In contrast, the proposed model has limited performance against scratch-type ink patterns. Specifically, when there are multiple co-linear blobs of failures, the model can identify their neighborhood as one that requires inking, but it cannot interpolate between the blobs to form a line. This limitation is by design as we seek to maintain the simplicity and interpretability of the feature vector and any attempt to model scratch-type inking patterns would require the addition of more complex features. Instead, to address this, we propose an image processing post-prediction correction methodology, capable of identifying any commonly occurring scratch type inking pattern, thus improving the automated die inking result.

Our proposed methodology uses the failure maps as inputs and includes the following steps:

1) Image transformation of the failure map
2) Filtering to remove sole failures and keep only clusters
3) Skeletonization of the remaining clusters
4) Skeleton filtering based on hard-coded shape parameters
5) Line interpolation for each group of co-linear line-segments. Optionally, extrapolation can be used to extend the inked area to the edge of the wafer.

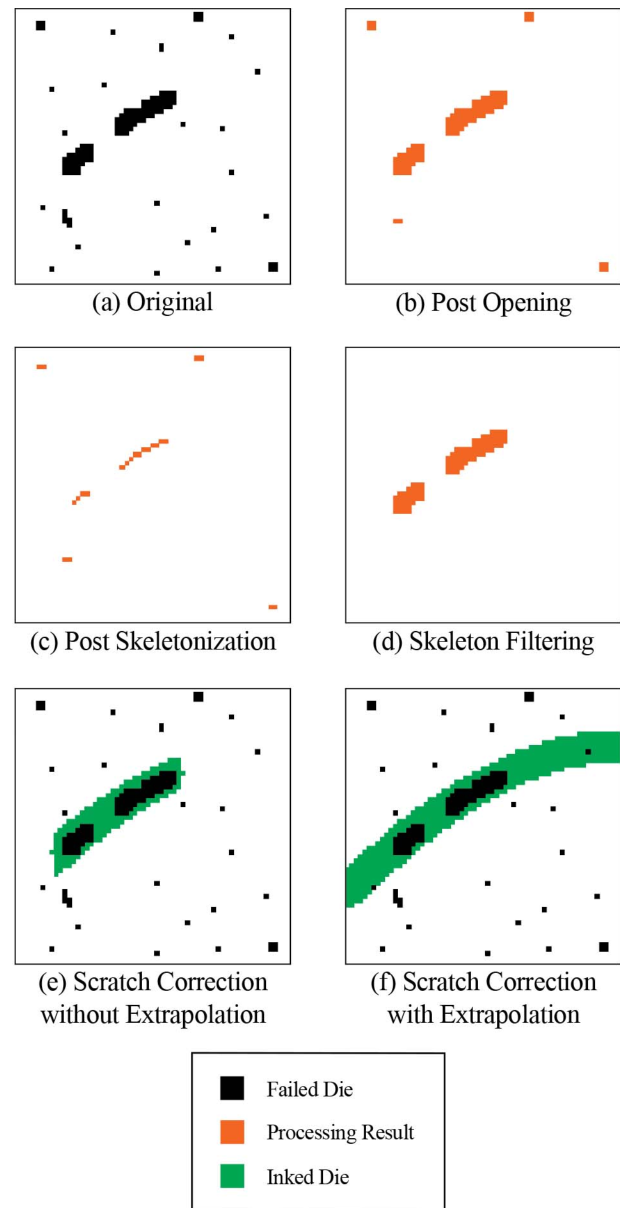


Fig. 5. An example of the proposed scratch correction algorithm with the intermediate steps and final inking result.

The first step above is required in order to subsequently apply image processing filters and transformations. It involves converting the failure wafermaps to binary images where colors indicate passing and failing devices. Once the image representation of a wafer is available, it is essential to remove any individual die or small clusters of die as those are unlikely to be part of any scratch patterns. This operation is performed using the common mathematical morphology called opening, which is a succession of the erosion and dilation operations. Figure 5 shows the result after every step of the proposed scratch detection automated inking algorithm. Figure 5(a) shows the result of the binary image transformation for a wafer

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

rectangular slice. Each black square represents a failed die, while white color squares represent the passing ones. After the second step of the proposed algorithm, namely the opening operation, all the small-sized clusters are removed, as shown in Figure 5(b).

The resulting image is then passed through skeletonization transformation [32], which reduces every cluster to a 1-die wide representation, thus better-highlighting clusters whose shape is closer to lines, as shown in Figure 5(c). During the skeleton analysis, which follows skeletonization, every line segment is filtered based on its co-linearity with other line segments and its size. Small line segments are removed as they represent failure clusters that are not line-shaped, while any long co-linear segments are kept. Figure 5(d) shows the result of filtering and Figures 5(e-f) show the final inking results, where the green-colored squares represent the die that will be automatically inked as they are likely to be affected by a scratch-type wafer defect. The option to extrapolate the detected line to the edge of the wafer allows product engineers to select whether they want to ink die outside the detected line segment. Figure 5(e) shows the result without extrapolation while Figure 5(f) shows how extrapolation would ink the devices that fall on the best fit line. Provided that this option can be enabled on an ad-hoc basis, it can allow product engineers to select the wafers that need to be more aggressively inked, beyond the starting ending points of the detected line. Although this feature can be enabled permanently, its use should be limited only on scratches whose detected length already covers a major portion of the wafer or their true shape, as indicated by failing die, matches the predicted curve. For example, the extrapolated line shown in Figure 5(f), although it represents the best-fitted curve for the two detected clusters, the inking confidence outside the two clusters should be considered low given the extrapolated ink pattern only covers two known failures, which are unlikely caused by a systematic wafer-level scratch defect.

IV. Experimental Results

A. Dataset Overview

To evaluate the effectiveness of the proposed methodology a dataset of several hundred thousand devices across 120 industrial wafers was used. After specification testing, whereby all failing devices were identified, each wafer was manually inked by product engineers and the locations of the inked die were indicated in the dataset by a specific bin number. Bin numbers corresponding to different failure types were also provided, allowing the proposed model to infer the significance of each failure type with respect to the inking decisions, as summarized in Section III-A.

B. Automated Inking Accuracy

To assess the overall ability of the proposed methodology in correctly identifying the areas on the wafer that require inking, a leave-one-out cross-validation experiment was performed. For this, each wafer was removed from the training dataset and the remaining wafers were used to train the model. Each column of Figure 6 shows the predictions of the automated inking methodology for one of three representative wafers of the dataset. The first row of wafermaps depicts the probability estimates for the positive inking decision. Gray represents 0% probability for those die locations to require inking, while other colors represent probability values in $(0, 100]$ as shown in the colormap on the right side. In the second row, corresponding prediction results are shown for the above three wafers, where passing die are depicted with gray color and blue colored die are the failing ones, with all bins been represented by one color.[1] Moreover, green indicates agreement between product engineers and the automated inking model (i.e., true positive predictions) and red represents manually inked die locations that weren't inked by the proposed methodology (i.e., false negatives). Purple colored die represent locations that were only marked as inked by the methodology but not by the product engineers (i.e., false positives).

As can be observed in Figure 6(a), certain clusters of die locations are selected by the proposed model, based on the failure density and their distance from the edge of the wafer. High probability, as represented by the red colored die, is in the center of every die cluster and decreases progressively the further away a die is located from that center. One of the major benefits of the post-prediction tuning approach using probability estimates is that it allows product engineers to tweak the degree of inking with a simple knob while evaluating the results. This implies that, even during the early stages of industrial integration of the proposed methodology, when the model is still learning, product engineers would save time by being able to quickly adjust the aggressiveness of the model or to use the predicted locations as suggestions, before making any manual alterations to the ink maps.

Figure 6(b) shows how the automated inking compares to the manual inking performed on these wafers. In order to match the aggressiveness of the manual inking, a threshold probability of 10% was chosen for the complete set of wafers. *Wafer A* included a single cluster at the bottom side of the wafer, which was manually inked by the product engineers, and which was correctly identified by the machine learning-based algorithm. At this level of aggressiveness, the algorithm also identified a smaller cluster of die near the center of the wafer. On the other hand, *Wafers B and C* exhibit multiple failure-dense clusters. Similarly to the previous wafer, the proposed model was able to identify the overall location and size of those clusters with minimal disagreement. By contrasting the probability estimates of these wafers in Figure 6(a), one can observe that a higher threshold would have produced a better matching inking pattern. Specifically, a threshold value of 30% for *Wafer B* would have accurately inked the top-right cluster, which was aggressively inked with a 10% threshold, but would have likely missed the smaller cluster on the right. This example illustrates a possible trade-off between different threshold values, which is best addressed by manually correcting any wrong inking decision, instead of merely relying on the inking-degree tuning. In other words, by adjusting the

[1] Detailed binning information may not be released due to an NDA under which the data has been provided to us.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

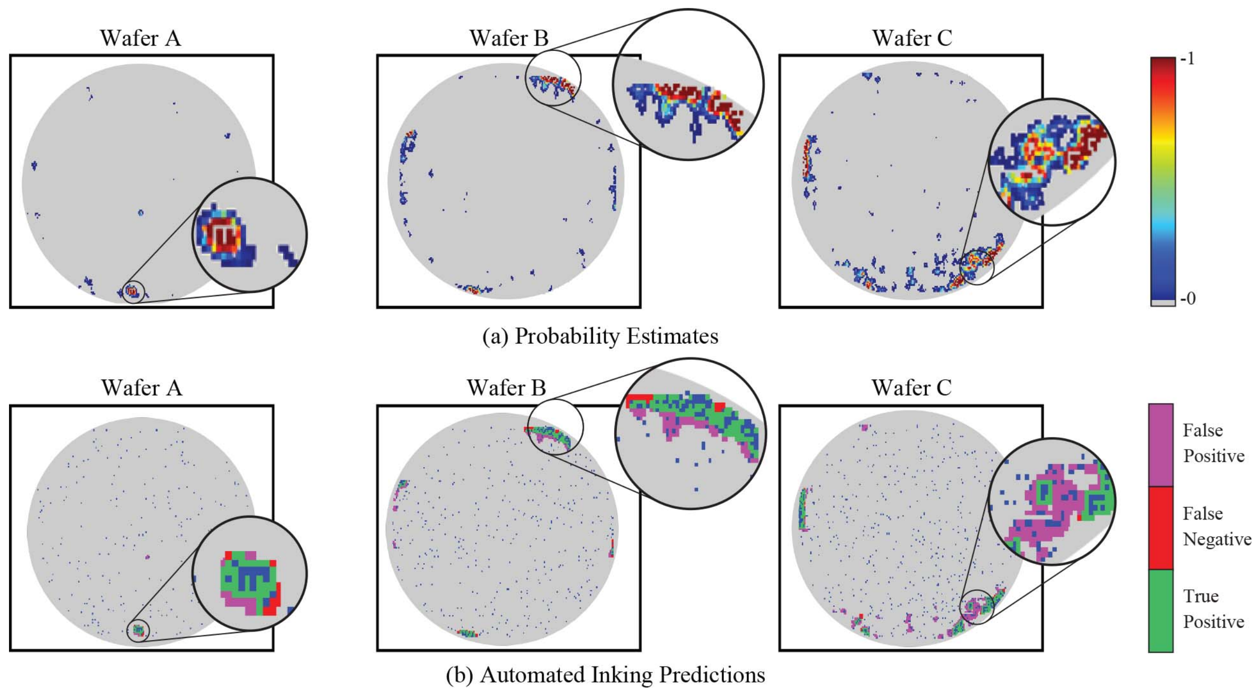


Fig. 6. Probability estimates and predictions for three sample wafers.

threshold, the product engineers affect the degree of inking performed by the proposed automated inking methodology, and not the decision function itself, that needs to be adjusted to eliminate the trade-off.

C. On-Line Machine Learning-Based Modeling

To accurately simulate the actual sequence of wafer arrival, the process of inking prediction, and the training of the proposed inking model, the wafers were sorted based on their manufacturing order, as reflected by their wafer and lot ID. For each wafer in the ordered dataset, the feature extraction step was first performed, using the locations and bins of each failing device. For the initial training of the model, we used only the first wafer. This allows us to better evaluate the learning rate of the proposed model, as it starts with the minimal available information. In practical cases, the model would have been trained with wafers from at least one lot, the wafers of which would have been manually inked. For all remaining wafers, feature extraction was performed in order to predict the inked locations accordingly. Following prediction, the known manual inked locations of the wafer under processing were used to update the proposed online model.

1) Learning Progression: As shown from the previous experiment, the proposed model can learn the inking strategy effectively when provided with all 119 training wafers. To evaluate the progress and the number of wafers needed for the model to produce useful inking decisions, a different experiment was performed, reflecting the proposed on-line learning algorithm. In this experiment, a single sample wafer was chosen as the target, while the algorithm was progressively trained using the remaining wafers. In other words, the selected wafer represents an unseen *newly manufactured wafer* at different learning iterations. At each iteration, the trained model was used to predict the target wafer and was updated with the next wafer from the dataset.

Figure 7 shows the comparison between the automated and manual inking decisions of the target wafer at different learning iterations. The first wafermap shows the initial prediction when only one wafer was used to train the model. As expected, the model has only learned that not inking any die location is a preferable strategy in terms of overall accuracy, as the number of non-inked die is significantly larger than the number of inked ones. After 21 wafers, it appears that the model has already learned that specific failure density related features, as well as the distance from the edge, are essential; thus, it correctly inks some of the die on the left and right sides of the wafer. This remains true and even shows minor improvements after a total of 34 wafers have been incrementally used for training. Although the size and location of the two die clusters that were selected by the product engineers have been correctly identified, the model appears to have weighted the distance from the edge more than it should, resulting in some false positive predictions shown on the left. Finally, after incremental learning has been performed with a total of either 70 or 114 wafers, the model has very accurately learned the manual inking strategy employed by the engineers for this product. Table I show the false positive false negative rates of the inking decision, when all test wafers have been processed and no adjustment has been made to the inking aggressiveness. As it can be observed, the overall accuracy of the model is very high, and it is more likely to over-ink a wafer compared to the manual inking results, the degree of which can be easily reduced using the post-prediction tuning explained in Section III-C.

Alternatively, the F_1*-score* can be used to summarize the progression of the proposed incremental learning methodology

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

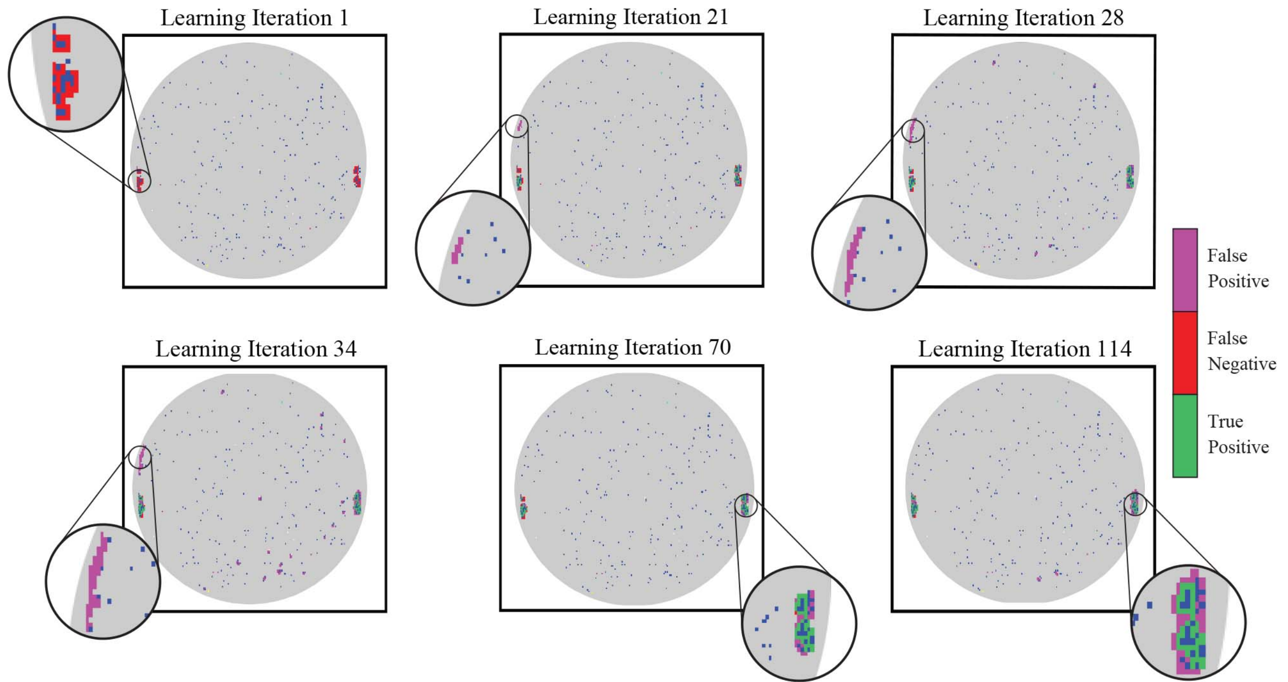


Fig. 7. Training progress as demonstrated on an unseen sample wafer at different learning iterations.

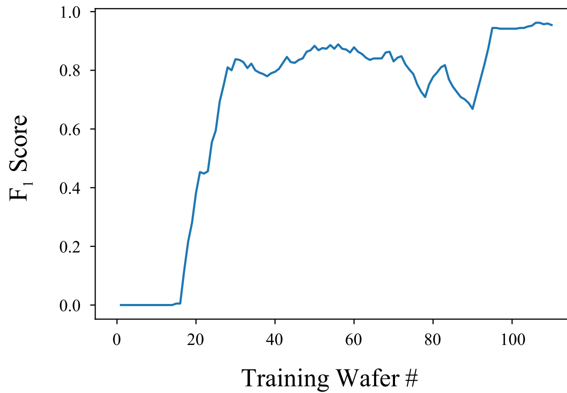


Fig. 8. F_1-score improvement during incremental learning.

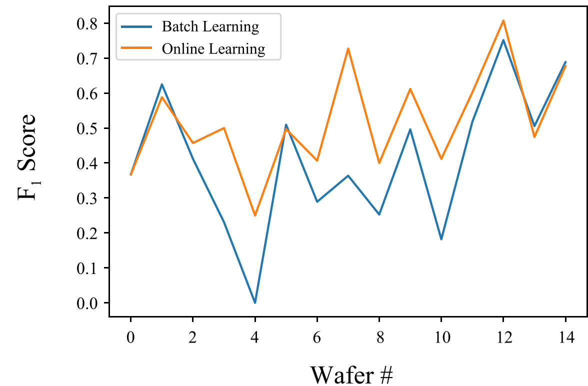


Fig. 9. Model accuracy comparison between batch and on-line learning.

TABLE I
ERROR RATES

False positives	**False Negatives**
1%	0.12%

at each iteration step. The F_1-score is defined as the harmonic mean of precision and recall, as shown in Equation 1. Precision, in a classification context, is defined as the proportion of positive identifications that were correctly classified, while recall is the proportion of actual positive classifications, in our case inked devices, that are correctly identified. Figure 8 shows how the F_1-score changes when predicting the target wafer during the execution of the proposed on-line learning methodology. As demonstrated, the F_1-score remains zero for the first few wafers, yet within the first lot it increases to above 0.8. Then, for two lots there is no significant improvement until there is a sudden decrease when the fourth lot is processed. This sudden fluctuation is caused by a low recall score, due to less conservative inking, which is then quickly corrected with the arrival of more wafers that provide a more robust understanding of the significance of the model features. With a relatively small number of wafers (less than four lots), the model is capable of learning the inking strategy and can be used to reduce the need for manual inking significantly. Furthermore, to avoid any previously unobserved patterns affecting the reliability of the devices sold, the monitoring technique explained in Section III-C2 can guarantee that a manual review will occur, and the model will get updated.

$$F_1 = 2 \cdot \frac{precision * recall}{precision + recall}. \quad (1)$$

2) Comparison to Batch Machine Learning-Based Modeling: Ideally, to effectively compare the proposed approach with the current state-of-the-art batch learning approach, a dataset containing multiple distinct inking strategies should have been used. These dissimilar strategies

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

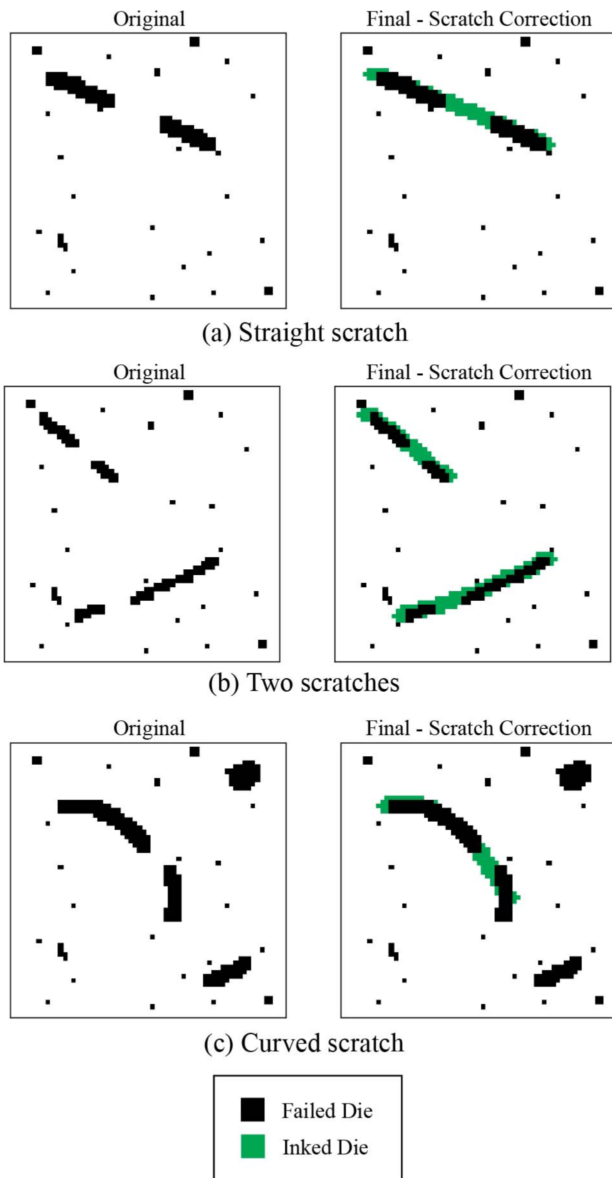


Fig. 10. Examples of Scratch correction for different simulated scratch shapes.

would showcase the primary benefit of the proposed approach to incrementally learn and accommodate them, compared to the static batch learning-based method. To simulate such a dataset, we re-ordered the industrial dataset described before, so that wafers exhibiting significant edge defects are pushed last. Assuming that this was the wafer manufacturing order, we compare the performance of the two approaches in predicting the last 15 wafers. In this experiment, both models have been initially trained using the same set of 105 wafers. The proposed model continues to be trained incrementally with every new wafer. Figure 9 shows the F_1-score for the two methods for each new wafer. As expected, the F_1-score for the first predicted wafer is the same for both methods, since they have been trained using the same wafers. After the ink maps for the first and second wafers are corrected, the on-line learning model learns that the edge-distance-based feature bears more significance and weighs it accordingly. For all remaining wafers, the on-line learning based model either outperforms or matches the accuracy of the batch learning model.

D. Scratch Correction

Figure 10 shows three more simulated examples of scratch correction without the intermediate steps, to illustrate the capabilities of the proposed methodology. The first pair of images shows another example of a straight scratch with a larger empty space between the two co-linear clusters and at a different angle, compared to the one used in Figure 5. The pair in Figure 10(b) shows that the algorithm presented has the ability to detect and ink more than one scratches by interpolating the filtered skeletons by groups according to their co-linearity. Finally, in Figure 10(c), the example of a more rare curved scratch is demonstrated, where in the fourth and fifth step of or proposed algorithm, the two skeletons generated are within the definitions of a line-shaped cluster and are co-linear enough to be considered part of the same scratch and have them properly inked.

V. CONCLUSION

Wafer-level failing patterns have been shown to be significant indicators for the existence of systematic defects and process shifts. Although research has been focused on automatically identifying and classifying such patterns, most decisions are still performed manually by product engineers. Marking of the devices that have a high probability for early-life defect manifestation remains a wearisome non-automatic procedure. In this work, we introduced an online learning-based methodology for predicting the inking patterns, by only utilizing the failure wafermaps. Using industrial data, the methodology was able to generate inking patterns matching those produced manually by product engineers but with greater consistency. Moreover, the proposed methodology allows the inking of systematic patterns at the lot level, as well as the post-prediction inking degree tuning and inking of scratch-type patterns, covering all the needs for its industrial integration.

REFERENCES

[1] E. A. Elsayed, "Overview of reliability testing," *IEEE Trans. Rel.*, vol. 61, no. 2, pp. 282–291, Jun. 2012.
[2] S. Bahukudumbi and K. Chakrabarty, *Wafer-Level Testing and Test During Burn-In for Integrated Circuits*. Boston, MA, USA: Artech House, 2010.
[3] M. F. Zakaria, Z. A. Kassim, M. P. Ooi, and S. Demidenko, "Reducing burn-in time through high-voltage stress test and Weibull statistical analysis," *IEEE Design Test Comput.*, vol. 23, no. 2, pp. 88–98, Mar./Apr. 2006.
[4] W. R. Mann, "Wafer test methods to improve semiconductor die reliability," *IEEE Design Test Comput.*, vol. 25, no. 6, pp. 528–537, Nov./Dec. 2008.
[5] R. Madge, M. Rehani, K. Cota, and W. R. Daasch, "Statistical post-processing at wafersort-an alternative to burn-in and a manufacturable solution to test limit setting for sub-micron technologies," in *Proc. 20th IEEE VLSI Test Symp.*, Monterey, CA, USA, 2002, pp. 69–74.
[6] *Guidelines for Part Averaging Testing*. Accessed: Dec. 9, 2011. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q001_Rev_D.pdf
[7] E. J. Marinissen *et al.*, "Adapting to adaptive testing," in *Proc. Design Autom. Test Eur. Conf.*, Dresden, Germany, 2010, pp. 556–561.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

[8] M. P. L. Ooi, E. K. J. Sim, Y. C. Kuang, L. Kleeman, C. Chan, and S. Demidenko, "Automatic defect cluster extraction for semiconductor wafers," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, Austin, TX, USA, 2010, pp. 1024–1029.
[9] R. Madge, B. Benware, R. Turakhia, R. Daasch, C. Schuermyer, and J. Ruffler, "In search of the optimum test set—Adaptive test methods for maximum defect coverage and lowest test cost," in *Proc. Int. Conf. Test*, Charlotte, NC, USA, 2004, pp. 203–212.
[10] S. Benner and O. Boroffice, "Optimal production test times through adaptive test programming," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 908–915.
[11] A. Ahmadi, C. Xanthopoulos, A. Nahar, B. Orr, M. Pas, and Y. Makris, "Harnessing process variations for optimizing wafer-level probe-test flow," in *Proc. IEEE Int. Test Conf.*, Fort Worth, TX, USA, 2016, pp. 1–8.
[12] M. J. Moreno-Lizaranzu and F. Cuesta, "Improving electronic sensor reliability by robust outlier screening," *Sensors*, vol. 13, no. 10, pp. 13521–13542, 2013.
[13] W. Dobbelaere *et al.*, "Analog fault coverage improvement using final-test dynamic part average testing," in *Proc. IEEE Int. Test Conf.*, Fort Worth, TX, USA, 2016, pp. 1–9.
[14] H. Stratigopoulos and C. Streitwieser, "Adaptive test with test escape estimation for mixed-signal ICs," *Trans. Comput-Aided Design Integr. Circuits Syst.*, vol. 37, no. 10, pp. 2125–2138, Oct. 2018.
[15] A. Nahar, R. Daasch, and S. Subramaniam, "Burn-in reduction using principal component analysis," in *Proc. IEEE Int. Conf. Test*, Austin, TX, USA, 2005, p. 10.
[16] N. Sumikawa, J. Tikkanen, L. Wang, L. Winemberg, and M. S. Abadir, "Screening customer returns with multivariate test analysis," in *Proc. IEEE Int. Test Conf.*, Anaheim, CA, USA, 2012, pp. 1–10.
[17] L. Fang, M. Lemnawar, and Y. Xing, "Cost effective outliers screening with moving limits and correlation testing for analogue ICs," in *Proc. IEEE Int. Test Conf.*, Santa Clara, CA, USA, 2006, pp. 1–10.
[18] W. R. Daasch, K. Cota, J. McNames, and R. Madge, "Neighbor selection for variance reduction in I/sub DDQ/ and other parametric data," in *Proc. IEEE Int. Test Conf.*, Baltimore, MD, USA, 2002, pp. 92–100.
[19] A. Nahar, K. M. Butler, J. M. Carulli, and C. Weinberger, "Quality improvement and cost reduction using statistical outlier methods," in *Proc. IEEE Int. Conf. Comput. Design*, Lake Tahoe, CA, USA, 2009, pp. 64–69.
[20] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, Sep. 1975.
[21] C. Xanthopoulos, P. Sarson, H. Reiter, and Y. Makris, "Automated die inking: A pattern recognition-based approach," in *Proc. IEEE Int. Test Conf.*, Fort Worth, TX, USA, 2017, pp. 1–6.
[22] V. V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, Jan. 2018.
[23] G. E. Hinton, "Connectionist learning procedures," *Artif. Intell.*, vol. 40, pp. 185–234, Sep. 1989.
[24] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Proc. Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, 2009, pp. 1393–1400.
[25] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. 13th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2001, pp. 388–394.
[26] C. D. Manning, P. Raghavan, and H. Schuetze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.
[27] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001.
[28] L. Bottou, "Online learning and stochastic approximations," in *On-Line Learning in Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Stat. (COMPSTAT'2010)*, 2010, pp. 177–187.
[30] H. Zaragoza and F. Alché-Buc, "Confidence measures for neural network classifiers," in *Proc. 7th Conf. Inf. Process. Manag. Uncertainty Knowl. Based Syst.*, 1998, pp. 886–893.
[31] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, 1950.
[32] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, "Building skeleton models via 3-D medial surface axis thinning algorithms," *Graph. Model Image Process.*, vol. 56, no. 6, pp. 462–478, 1994.

Constantinos Xanthopoulos (Student Member, IEEE) received the B.S. degree in computer science from the University of Piraeus, Greece, in 2012, and the M.S. degree in computer engineering from the University of Texas at Dallas in 2015, where he is currently pursuing the Ph.D. degree in computer engineering, as a member of the Trusted and RELiable Architectures Laboratory. His research interests focus on the application of statistical learning theory and machine learning to problems in analog test.

Arnold Neckermann received the B.S. degree in electronics and technology management and the M.S. degree in advanced electronic engineering from the University of Applied Sciences Joanneum Kapfenberg, Austria, in 2014 and 2017, respectively. Afterwards, he started working in the semiconductor industry with ams AG, Premstätten, Austria, as a Product Engineer. He is responsible for gain-cost improvements on Automotive and Costumer products. Thereby he is specialized in advanced data as well as laboratory analysis.

Paulus List received the B.S. degree in technical physics and the M.S. degree in advanced material science of semiconductor process technology and nanotechnology from the Graz University of Technology, Austria, in 2013 and 2016, respectively. He started to work in the semiconductor industry with NXP and is currently working as a Product Engineer with ams AG focused on industrialization and optimization of the production flow of ASICs used as MEMS interfaces.

Klaus-Peter Tschernay received the Dipl.-Ing. degree in electrical and electronic engineering from the Technical University of Graz in 1984. Since 1984, he has been working with ams AG, an Austrian supplier of integrated circuits, focused on sensors. After several roles in product engineering and product engineering management he currently holds the position of the Director Innovation of the Ear and Automotive Solutions Division. His main focus is on innovative industrialization concepts for sensor solutions.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.

Peter Sarson (Senior Member, IEEE) was born in Sheffield, U.K., in 1976. He received the B.Eng. degree (Hons.) from the University of Sheffield, U.K., in 1998, where he specialized in telecommunications. He is currently pursuing the Ph.D. degree with Gunma University. He has 17 years experience in the field of semiconductor test. He started his career as an Applications Engineer with Xcerra Corporation (formally, LTX) in 2000. He has worked with ams AG, Austria, for the Full Service Foundry Division, where he lead the Test Development Organization. He was previously the Principal Engineer with the Corporate Test Development Team, ams AG. He received the Chartered Manager status form the Chartered Management Institute in 2014.

Yiorgos Makris (Senior Member, IEEE) received the Diploma degree in computer engineering from the University of Patras, Greece, in 1995, and the M.S. and Ph.D. degrees in computer engineering from the University of California at San Diego, in 1998 and 2001, respectively. After spending a decade on the faculty of Yale University, he joined UT Dallas, where he is currently a Professor of electrical and computer engineering, leading the Trusted and RELiable Architectures Research Laboratory, and the Safety, Security and Healthcare thrust leader for Texas Analog Center of Excellence. His research focuses on applications of machine learning and statistical analysis in the development of trusted and reliable integrated circuits and systems, with particular emphasis in the analog/RF domain. He was a recipient of the 2006 Sheffield Distinguished Teaching Award, the Best Paper Awards from the 2013 IEEE/ACM Design Automation and Test in Europe conference, the 2015 IEEE VLSI Test Symposium, as well as the Best Hardware Demonstration Awards from the 2016 and the 2018 IEEE Hardware-Oriented Security and Trust Symposia (HOST'16 and HOST'18). He serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and has served as an Associate Editor for the IEEE INFORMATION FORENSICS AND SECURITY and the *IEEE Design & Test of Computers Periodical*, and as a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:18:05 UTC from IEEE Xplore. Restrictions apply.