Automated Die Inking through On-line Machine Learning

Constantinos Xanthopoulos*, Arnold Neckermann†, Paulus List†,
Klaus-Peter Tschernay†, Peter Sarson‡1 and Yiorgos Makris*
*Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA
†ams AG, Premstaetten 8141, Austria
‡Dialog Semiconductor LTD, Swindon, SN57UL, United Kingdom

***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. Recently, machine learning has been used to automate and speed-up the inking process. In this work, we employ on-line machine learning to address the practicability limitations of the current state-of-the-art automated inking approach. Effectiveness is demonstrated on an industrial dataset of manually inked wafers.**

I. Introduction

As reliability becomes imperative for industrial and automotive applications, relying on IC testing alone to identify the failure-prone devices has become a significant challenge. To complement testing, several screening techniques are often used, each based on different criteria. The general premise behind these techniques is that any 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) [1] aim to identify the passing die that exhibit marginal test measurements relative to the main distribution of each wafer. Once a wafer has completed wafer sort, the wafer-level distribution of all test measurements is known and robust statistics can be calculated. These statistics are subsequently used to identify any passing outliers which are then discarded or marked for further testing. DPAT is an automated technique which only marginally increases the overall test time.

At the opposite extreme, burn-in testing is a time-consuming technique that allows the detection of manufacturing imperfections. Burn-in stress-tests each device to its operational electrical and temperature limits, in order to accelerate manifestation of latent defects on devices that have passed all previous testing stages. The excessive cost of this process is a result of the addition of one more die-level test insertion and the relatively long time it takes for the stress-test to force the manifestation of any possible defects.

To reduce the number of devices that go through burn-in and to complement the detection capabilities of all other screening methods, manual die inking is used to mark passing die that are located near failing devices on a given wafer. 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. Manual inking is performed by product engineers after the completion of wafer sort and the application of any statistical-based screening methods. Product engineers visually inspect the failure maps for each manufactured wafer, identify clusters of failures for certain critical failure types and then mark the neighboring die as inked. Despite the prescription of generic strategies, inking remains a highly subjective process based on the experience of each product engineer. As a result, manual inking is inconsistently performed between different engineers and often, due to its complexity, even between inspections by the same engineer.

[1] This author was with ams AG when this work took place.

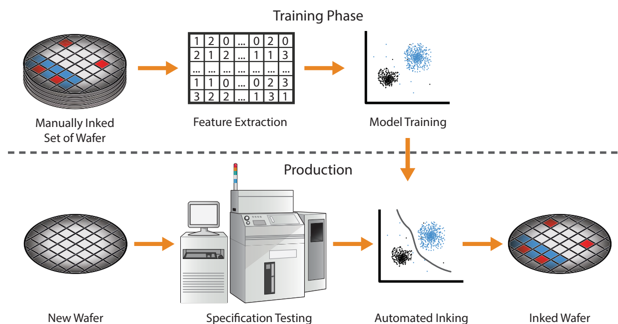

Fig. 1. Flow of the automated inking method proposed in [2]

Theoretically, automation of the inking process can be achieved by developing a rule-based system that utilizes the failure maps. This system requires 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. Designing, maintaining, and adapting such a system to additional products is a mounting challenge that would require multiple development iterations, resulting in a significantly complex system.

Alternatively, authors in [2] proposed a pattern recognition-based approach for the automation of the inking process. In this approach, a machine learning model was trained based on manually inked wafers to decide whether a given die should be inked, or not. Figure 1 shows the two-stage process proposed in [2], wherein a set of wafers is used to extract the model features for training, while the manual inking decision is used

978-1-7281-2490-2/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:22:42 UTC from IEEE Xplore. Restrictions apply.

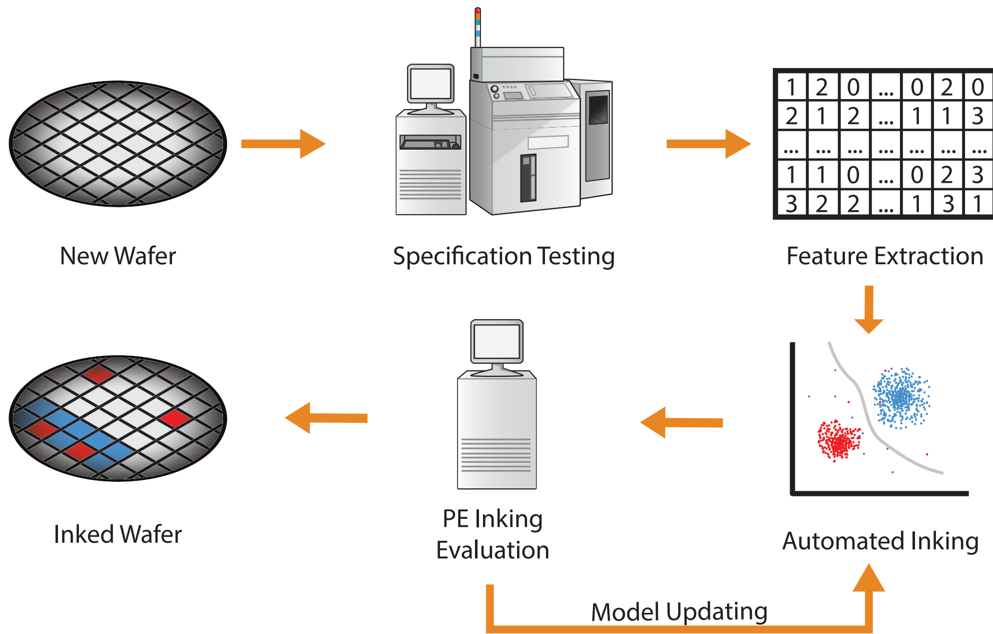

Fig. 2. Flow of the proposed on-line learning automated inking method

as the prediction target. After training, the model predicts the inking decision for every newly manufactured wafer based on its failure map, eliminating the need for visual inspection and manual inking. Despite the model's high accuracy, which demonstrated its ability to learn the underlying manual inking strategy, several practicability related limitations emerged. Such a two-phase system, where training and production deployment are fully separated, requires availability of a relatively large number of wafers in order to train a model capable of supporting all different inking strategies. Moreover, any future process shifts that might lead to a supplementary inking policy will force a complete re-training of the model. Finally, one more limitation of the modeling approach in [2] is that even subtle changes to the inking decisions must be accumulated over time and added to the initial training set to improve the overall inking accuracy.

To address these limitations, in this work we propose an on-line machine learning-based approach which allows for continuous updating of the learned inking strategy. The remainder of the paper is organized as follows. Section II gives an overview of the proposed approach. In Section III, we evaluate the on-line learning-based automated inking methodology on an industrial dataset of manually inked wafers and in Section IV we draw conclusions.

II. Proposed Approach

To address the above-mentioned practical drawbacks, we propose the shift to the on-line machine learning paradigm. The current state-of-the-art is based on a batch-learning type of classifier, which takes a collection of manually inked wafers to learn the binary decision boundary. In contrast, on-line learning does not separate the training phase from the production phase; instead, it creates a unified flow which contains a feedback loop responsible for updating the model as needed. Figure 2 shows the proposed approach, where for each newly manufactured and fully tested wafer the failure map is used to extract the features needed by the model to predict the location of the inked die. Once these wafers have been automatically inked, an evaluation step for assessing 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 removing 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 methods by a committee of engineers.

This continuous feedback allows for more control over how inking is being performed, especially during ramp-up where the model learns the inking strategy for each new product. During these early wafers, the model will serve as a suggestion tool for the engineers, pointing the most likely locations that need inking. In addition, as the automated inking model is getting adjusted by the PE's corrections, its accuracy will progressively get increased and, as a result, less time will be required for evaluation. After the model has been trained with a sufficient number of wafers, high prediction confidence will allow manufacturers to auto-approve the automated inking decisions, thus eliminating the need for visual inspection.

A. On-line Machine Learning

Supervised machine learning aims to build a model that expresses the desired output as a function of the input features. This is traditionally achieved with a training dataset consisting of $(\mathbf{x}, y)$ pairs of input vectors and target values. This approach

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:22:42 UTC from IEEE Xplore. Restrictions apply.

is commonly referred to as batch learning since it requires such collection of training pairs to exist for the model to be trained. In many practical applications, the availability of a large training dataset cannot be easily satisfied as samples arrive over time rather than all being available up front. To address this limitation, as well as others that stem from finite computational resources (e.g., run-time and storage memory), the idea of incremental learning has been introduced. Incremental or on-line learning refers to the ability of a model to be trained with smaller training sets, either in the form of mini-batches or even with a single instance at a time. This ability provides multiple benefits for practical applications where data arrive in streams.

B. Classification

Similarly to the study in [2], a binary classifier model needs to be trained in order to infer the inking decision based on the wafer failure maps. Several models [3] have the ability to be incrementally learned, some of which include Multilayer Perceptron [4], On-line Random Forests [5], Incremental Support Vector Machines [6], Naive Bayes [7], and Learn++ [8]. In this work, we use a Multilayer Perceptron (MLP) classifier with Stochastic Gradient Descent (SDG) [9], [10].

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.

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 on-line learning paradigm.

C. Feature Extraction

Both training and prediction require a die-level feature vector that is indicative of the likelihood of a die to be inked. As described in [2], these features are based on the distance of each die from the edge of the wafer, as well as the failure density of the neighborhood for each failure type, with the various failure types being denoted by different bin numbers.

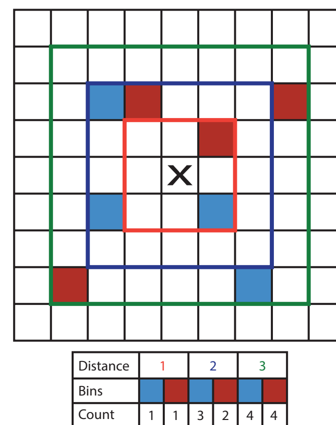

Fig. 3. Feature extraction example for a maximum die distance of three

Figure 3 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, as proposed in [2], counts the number of failures at certain distances away from the target die, as shown by the *count* row in Figure 3, 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.

Likewise, measuring the distance from the edge allows the distinction between failure-dense areas near the center of the wafer, as compared to the ones near the edge. This distinction is important, as wafers tend to be more sensitive near their edge and, therefore, fewer failures are required for a positive inking decision to be made.

D. Post-prediction 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 [2]. During that step, the size of the automatically inked areas was reduced or increased by a pre-determined and hard-coded degree.

In this work, we propose an alternative approach for post-prediction tuning of the inking result based on the confidence

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:22:42 UTC from IEEE Xplore. Restrictions apply.

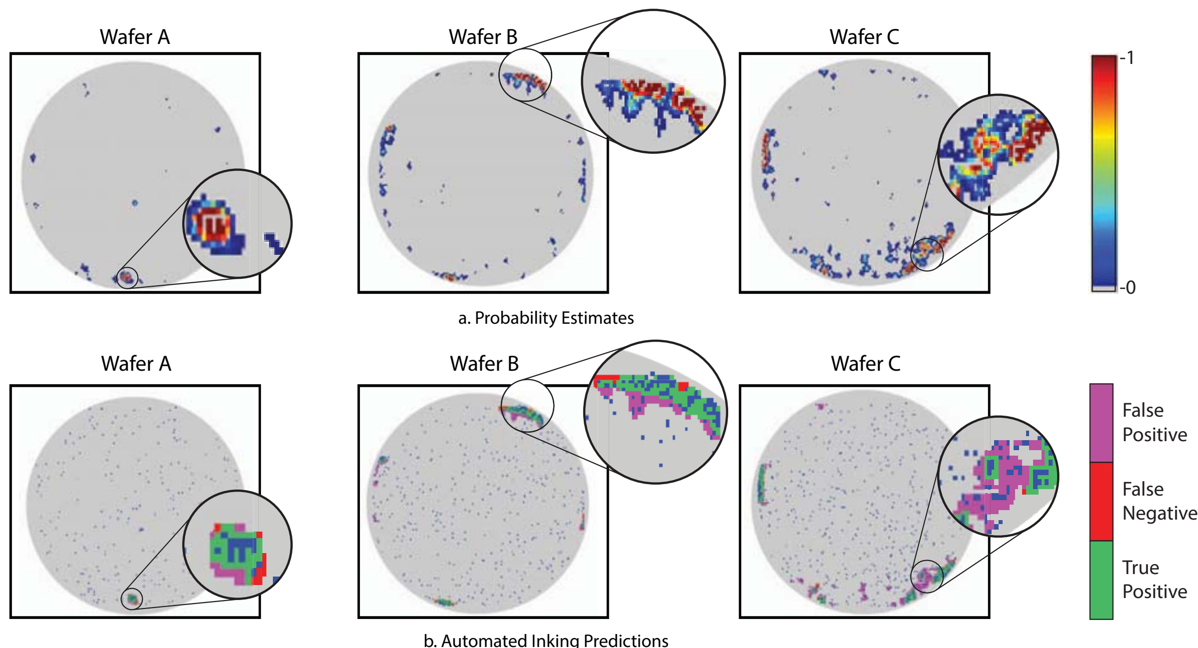

Fig. 4. Probability estimates and predictions for three sample wafers

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 [11]. This technique allows for a dynamic adaptation of the degree of inking performed by the proposed methodology, by adjusting the decision threshold. 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.

III. 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 II-C.

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 4 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 4.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

[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 22:22:42 UTC from IEEE Xplore. Restrictions apply.

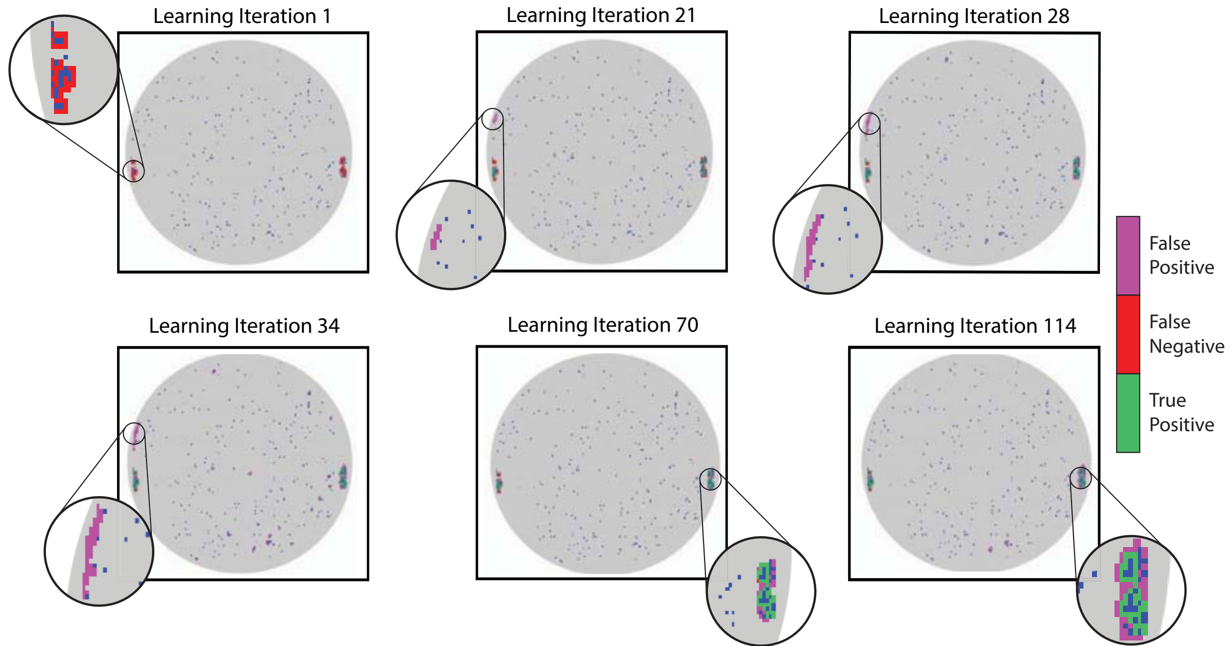

Fig. 5. Comparison between automated and manual inking at different learning iterations

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 4.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 4.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.

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 5 shows the comparison between the automated and manual inking decisions of the target wafer at different learning iterations. The first wafer map 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

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:22:42 UTC from IEEE Xplore. Restrictions apply.

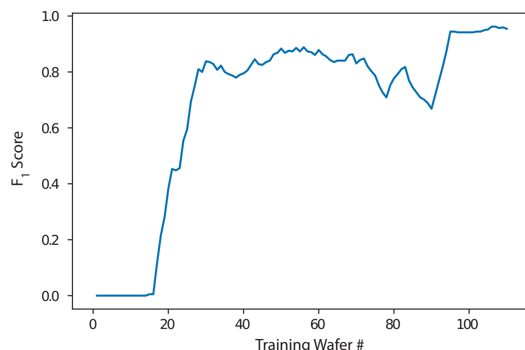

Fig. 6. F_1-score improvement during incremental learning

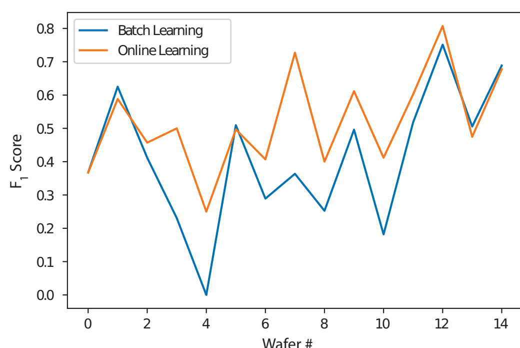

Fig. 7. Model accuracy comparison between batch and on-line learning

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.

Alternatively, the *F_1-score* can be used to summarize the progression of the proposed incremental learning methodology at each iteration step. The F_1-score is defined as the harmonic mean of precision and recall. Figure 6 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 for the significance of the model features.

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 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 7 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.

IV. Conclusions

In this work, we sought to improve the state-of-the-art automated inking algorithm, by employing an on-line machine learning-based methodology. This paradigm shift allows for smooth integration with current industrial environments, by balancing the trade-off between the level of human interaction and confidence towards the process. This is achieved through the ability to monitor, calibrate and continuously update the inking prediction model . High accuracy and fast learning rate were demonstrated on an industrial dataset with more than two million devices.

References

[1] "Guidelines for Part Averaging Testing," Automotive Electronics Council (AEC), AEC-Q001 Rev-D, 2011.

[2] C. Xanthopoulos, P. Sarson, H. Reiter, and Y. Makris, "Automated die inking: A pattern recognition-based approach," in *International Test Conference (ITC)*, 2017.

[3] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, 2018.

[4] G. E. Hinton, "Connectionist learning procedures," *Artificial Intelligence*, vol. 40, 1989.

[5] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *International Conference on Computer Vision*, 2009.

[6] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in neural information processing systems*, 2001.

[7] C. D. Manning, P. Raghavan, and H. Schuetze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[8] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An Incremental Learning Algorithm for Supervised Neural Networks," *Systems Man and Cybernetics*, vol. 31, 2001.

[9] L. Bottou, "Online Learning and Stochastic Approximations," in *On-line learning in neural networks*. Cambridge University Press, 1998.

[10] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Computational Statistics (COMPSTAT)*, 2010.

[11] H. Zaragoza and F. d'Alché Buc, "Confidence Measures for Neural Network Classifiers," in *Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1998.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on August 02,2020 at 22:22:42 UTC from IEEE Xplore. Restrictions apply.