Automated Die Inking: A Pattern Recognition-Based Approach

Constantinos Xanthopoulos*, Peter Sarson†, Heinz Reiter†, and Yiorgos Makris*

*Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX USA, 75080
†ams AG, Premstaetten, Austria, 8141

***Abstract*—Manual wafer-level die inking is a common procedure for excluding die locations that are likely to be defective. Although this is a more cost-effective process, as compared to the expensive burn-in tests, it remains a labor-intensive step during IC testing. For each manufactured wafer, test engineers have to visually inspect every failure map in order to identify any regions where additional die need to be marked and discarded. Towards reducing this cost, we introduce a novel pattern recognition 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.**

Keywords: Inking, test cost reduction, automation, latent defects, infant mortality

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 greatly improve manufacturability, testing and production yield, the detection of latent defects remains a mounting challenge, both in terms of complexity and in terms of cost.

A common practice for identifying latent defects is the burn-in process, during which the chips are subjected to higher frequencies, voltages, and temperatures. The goal of these stress tests is to accelerate the manifestation of any imminent but latent defects. Although burn-in tests are very effective in identifying such manufacturing imperfections, this incurs increased complexity of test-floor logistics and significant cost overhead, which is prohibitive for high-volume manufacturing.

Based on the observation that manufacturing defects are spatially correlated on the wafer surface [1], manual inking process is being used to filter out the devices that are likely to exhibit latent defects. Figure 1 shows the flow of the 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 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 resemblance to commonly occurring defect patterns. Although inking is a cost effective solution, as compared to the expensive burn-in testing, it is a manual process that requires human intervention. This entails added time, cost and possibly subjective discrepancies between inking decisions.

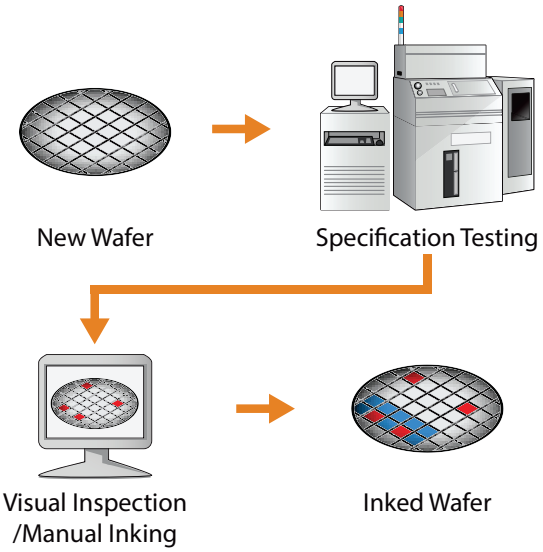


Fig. 1. Current inking process

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) that are generated from the 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 can either originate from the manually inked or the post burn-in 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 and, therefore, readily applicable to all existent products where inking is currently performed with minimal modifications.

The remainder of this paper is organized as follows. After a short introduction of related work in Section II, Section III gives an overview of the proposed approach. In Section IV, we evaluate the accuracy of the automated inking methodology on an industrial dataset of manually inked wafers. Section V

978-1-5386-3413-4/17/$31.00

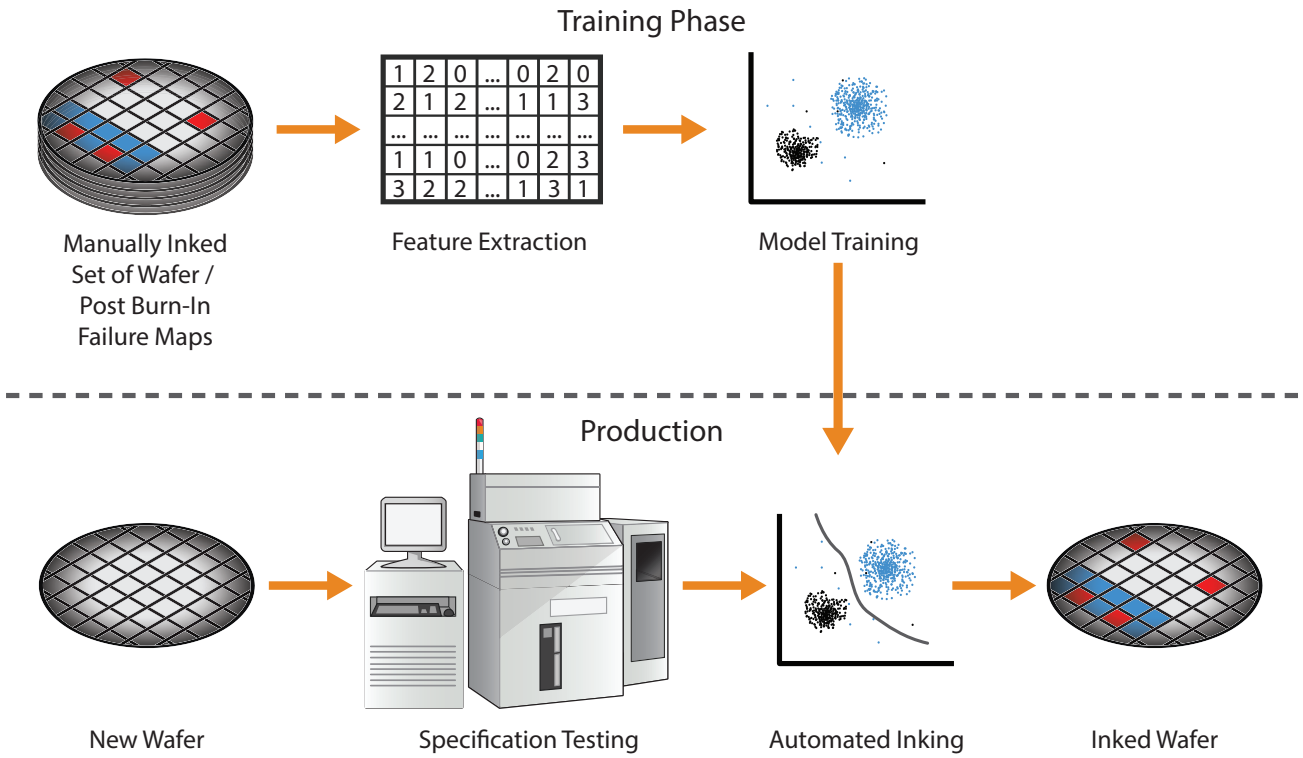


Fig. 2. Flow of proposed approach

provides an outlook of possible future work and conclusions are drawn in Section VI.

II. Related Work

In recent years, several studies have focused on identification and classification of wafer-level failure patterns. Recognition of systematic defect patterns has been shown to assist in the prevention of process-related issues, yield improvement, and yield estimation, as well as in the detection of process excursions.

The authors in [2] laid the groundwork by showing the benefits of visual analysis of the defect patterns, as compared to simple statistical approaches that rely on cumulative defect count statistics. Towards yield improvement, the authors in [3] introduced a machine learning-based pattern identification system to aid in the detection of systematic failure causes. Similarly, in [4], the authors did a comparative study between various classification algorithms with features extracted by the use of Polar Fourier Transform (PFT) and Rotational Moment Invariant (RMI) methods. A denoising algorithm was presented in [5], to automatically detect non-overlapping clusters of defective die on a wafer.

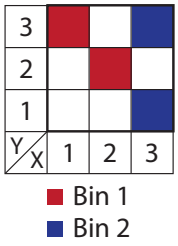
In a different direction, an inter-wafer pattern mining study was performed in [6], where the goal was to identify abnormal failure patterns which are possibly linked to subtle systematic process problems. Similarly, in [7], a methodology for clustering wafer-level spatial signatures was introduced, in order to group the wafers and identify any outliers in the presence of random defect and variation.

III. Proposed Approach

In this work, we introduce a classification-based methodology to emulate the decision process that is usually performed manually by an engineer. Figure 2 shows the flow of the proposed approach. During the training phase, we first extract the features from an initial set of manually inked wafers or a set of post burn-in failure maps. Once these features are extracted, we train a supervised model to learn the relation between the failure signature of each die neighborhood and the corresponding inking pattern. After the model is trained, it can be used to either fully replace or expedite the manual inking process by providing instantaneous suggestions to the supervising engineers.

A. Feature Extraction

To simplify test-floor logistics, one of the over-arching constraints while developing an automated inking solution was to rely only on the pass/fail decisions of the specification tests and avoid the use of the actual test measurements. As a result, our classification model must be trained by features that can be extracted from the map of failed die locations for each wafer in a given training set. These features must be selected such that they reflect the failing conditions in the neighborhood of each die. To enhance the prediction accuracy of our model, rather than relying only on the pass/fail status of each die, the various binning groups are also considered. Usually, the binning groups denote different types of defects or performance characteristics of each chip; a different identification number, commonly termed bin number, gets assigned by the



a.

		Bin 1		Bin 2	
X	Y	D_1	D_2	D_1	D_2
1	1	0	2	0	1
1	2	2	2	0	0
2	1	1	1	1	1
2	3	2	2	1	1
3	2	1	1	2	2

b.

Fig. 3. Feature extraction example of a wafer segment and its corresponding feature vectors

test program. One advantage of the proposed method is that it handles any discrepancies between the defect severity each bin number reflects, across different products. Static solutions which are based on simple statistical metrics or image processing techniques would require adaptation to each product's test program output in order address these discrepancies. On the other hand, these adaptations are intrinsically reflected by the manually inked wafers or the post burn-in failure maps which are used to train the classification model.

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 all distances up to a predefined maximum neighborhood distance. At the end of this process, 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.

Algorithm 1: Feature Extraction

```
for d in passing die do
    for b in failing bins do
        for k in 1...max_k ;     // max_k: The maximum
        considered distance for a neighborhood
        do
            features = GetNumberOfNeighbors(d, b, k) ;
            // Function that returns the number of
            neighbors within k distance of die d
            that belong to bin b
        end
    end
    features += GetDistanceToEdge(d) ;     // Function
    that returns the distance of die d to the
    nearest edge of the wafer
end
```

The implementation of the above-mentioned *GetNumberOfNeighbors* function was done by using the K-Dimensional Trees [8] 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. As 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 radius limit *max_k*.

As an example Fig. 3.a, shows a 3×3 wafer segment with four failing die locations colored according to their binning group number. The table in Fig. 3.b displays the generated feature vectors for the non-failing die locations, with *max_k* set to 2. Computing the feature vector for die location $d_{2,3}$ is performed as follows:

- For bin group 1 the number of failing neighboring die locations with distance equal or less to 1 is 2, namely $d_{2,2}$ and $d_{1,3}$.
- Similarly for distance equal or less to 2 the number of die is still 2.
- For bin group 2 there is only one die ($d_{3,3}$) that is within distance 1 and consequently within distance 2.

B. Classifier Training

As shown in Figure 2, following the feature extraction, a binary classifier is trained to learn the relation between the extracted features and the inking decision. Many non-linear classification algorithms such as the Nearest Neighbors, Support Vector Machines (SVM), Gaussian Processes, Decision Trees and Neural Networks can learn this relation, with minor differences in their accuracy. In this work, we propose the use of a binary Support Vector Machine (SVM) [9] classifier. The SVM was selected due to its ability to create non-linear classification boundaries and also handle high dimensional input data while being computationally more efficient than the above-mentioned alternative algorithms. An SVM is a supervised learning algorithm that aims to create a maximum-margin hyperplane, which separates the samples of the two classes. To achieve this, the algorithm uses a non-linear mapping of the inputs into a high-dimensional feature space, an operation commonly known as "kernel-trick". The choice of a kernel depends on the nature of the input data and classification task. In this work, we employ the Radial Basis Function Kernel which has the following format:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$$

where, $\mathbf{x}$ and $\mathbf{x}'$ represent the feature vectors of two samples, while $\|\mathbf{x} - \mathbf{x}'\|$ is their Euclidean distance.

C. Post-Prediction Processing

To further improve the accuracy of the automatically generated inking pattern, a few adjustments can be made after its prediction. These adjustments aim to tune the size of each inked region to either more conservative or more aggressive results, by increasing or decreasing it, respectively. Herein, we propose the use of the simple erosion/dilation morphological

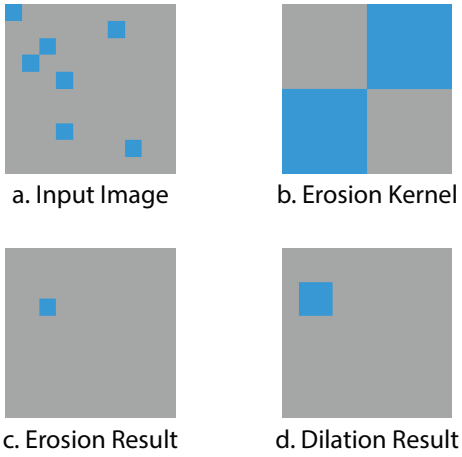


Fig. 4. Erosion and dilation output on a inking pattern

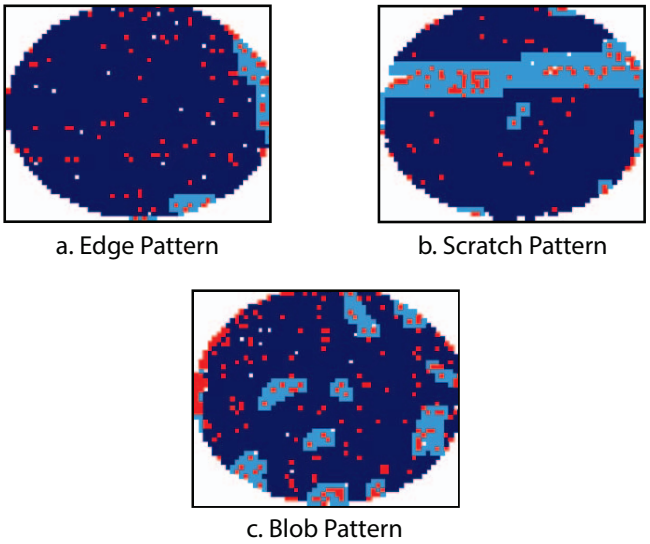


Fig. 5. Examples of each failure pattern type

operations [10]. These operations are commonly applied in computer vision and image processing applications. Both of these actions are binary operators using different kernels, depending on the nature of the application.

During erosion, the selected kernel slides through the provided image and only the pixels of the binary image that are matching the ones in the kernel are kept, while the rest are eliminated. Effectively, the result of this operation is to remove any small patterns that are considered to be noise.

Dilation is the opposite operation of erosion and is used to increase the marked area of the processed image. This is achieved by relaxing the condition for the kernel to be applied, such that only one of the pixels in the kernel needs to match the image pixels.

Usually, these operations are executed consecutively in order to first remove any random noise and then highlight the relevant patterns of the image. An example of the erosion and dilation operations is shown in Figure 4. The first image is an example of a binary input while the second shows the selected 2×2 kernel. The result of the erosion operation is presented in the third image, wherein it can be seen that the original pattern has been simplified. The erosion-produced image is then fed to the dilation operator with a 2×2 filled kernel, in order to highlight the area where the desired pattern is located.

IV. Experimental Results

A. Dataset overview

In order to experimentally evaluate the effectiveness of the proposed pattern inking methodology, we use a set of industrial wafers exhibiting various failure patterns. The data consists of 123 wafers across several lots, each with more than 2,000 devices. Along with specification testing, manual inking has been performed by different engineers, and a unique bin number has been used to identify each marked die. The dataset has been selected such that it includes all types of failure patterns, as well as wafers with no inked die. The failure patterns are divided into blob type, where continuous regions of the wafer have been inked, edge type where the majority of inked die are along the edge of the wafer or close to it, and, finally, scratch patterns where a line of failed devices exists. Table I summarizes the distribution of the above types in our dataset.

TABLE I
Distribution of failure types

Blob	Edge	Scratch	None
40%	29%	3%	28%

Figure 5 shows a representative wafer from our dataset for each failure pattern type. The red colored die have failed the specification tests, while the dark blue have passed. Light blue denotes the die that have been manually inked by the engineers. While the failing die in our dataset are associated with a unique failing bin number, that information cannot be disclosed. In Figure 5.a, the inked die form clusters near the edge of the wafer, while in 5.b, the majority of inked die form a line. Figure 5.c shows an example of the most common and generic formation, namely that of irregularly shaped blobs of die.

B. Feature Extraction

Before training the classification model, we need to derive a set of features for each die that represent its neighborhood. As mentioned in section III, the length of the feature vector depends on the number of failing bins generated by the test program, as well as the considered neighborhood sizes. In the available dataset, we identified three distinct bins, each representing a different defect type. To generate the feature vectors, we first need to determine what is the maximum size (k) of our neighborhoods, by evaluating the accuracy of a classification model in a small hold-out set of wafers while increasing k. Figure 6 shows the results of this experiment. As we can observe, the knee of the curve corresponds to a neighborhood of size 6. This indicates that by further

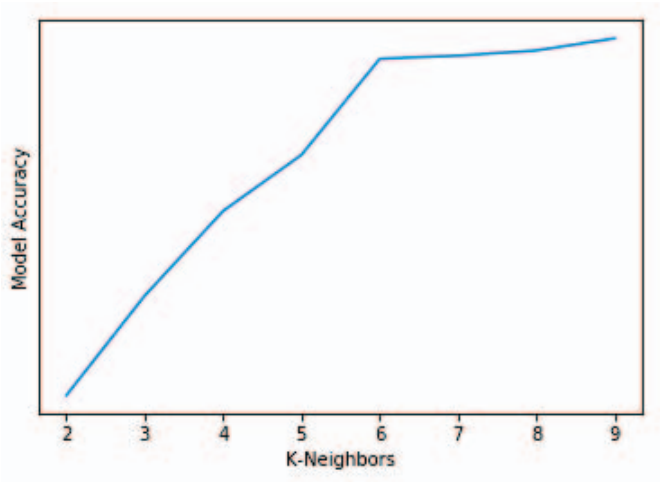


Fig. 6. Model accuracy vs size of neighborhood

increasing the maximum size of the neighborhood, the model does not improve significantly.

Given the maximum size of the neighborhood, as well as the number of distinct bins, we can now generate the feature vector by calculating the $k = 1 \ldots 6$ nearest neighbors of each die and for each failure bin. After including the distance from the wafer center, our resulting vector has a length of 19 features.

C. Prediction results

Once the feature vector has been generated, we can train an SVM classifier to learn the inking patterns. Due to the limited number of wafers in our dataset, we performed several leave-p-out cross-validation experiments by randomly splitting our dataset. In each iteration, we used 75% of the wafers for training and 25% for testing.

Figure 8 shows the failed die locations of three wafers as well as the comparison between the manual and the automated inking process. As can be observed, the automated inking process predicts the locations that the engineers have selected in Wafer A, with high accuracy. This wafer is representative of the majority of the predictions in our dataset, where there are only a few missclassified locations usually at the perimeter of the manually inked area. In more detail, the automated process in Wafer A missclassifies only 69 die locations, by conservatively estimating the size of the inked area.

Wafer B shows one of the few examples where the automated process does not adequately predict the manually inked pattern. By visually inspecting all the wafers where the prediction does not match the engineer's inked pattern, we discovered that they all exhibit scratch-type failure patterns. This can be attributed to the fact that only 3% of our training set includes this type failing patterns.

Wafer C in Figure 8 presents a rare case in our dataset, where although the input wafer has not been subjected to manual inking, our prediction model does generate an inking pattern. Upon further evaluation from reliability experts, the automatically generated inking pattern, while more aggressive, marks the areas of the wafer that are likely to contain latent

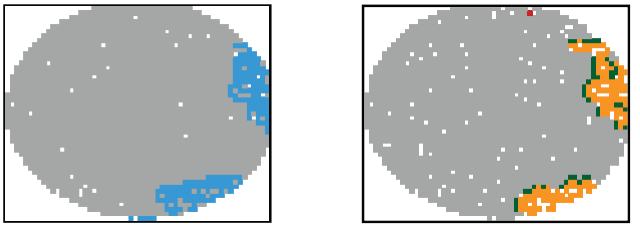


Fig. 7. Erosion-dilation example performed on a automatically predicted ink pattern.

defects and is more preferable than the manual decisions in our dataset.

The mean accuracy and error rates of the classification model, as compared to the manually inked patterns, are summarized in Table II. The error is divided further to false positive and false negative rates. As shown, the automated inking model has a mean accuracy of 96.1%, for the wafers in the test sets. Furthermore, false negative predictions constitute the majority of the missclassified die, with a rate of 3.07%. In these results, false negative predictions represent conservative inking decisions, where the model under-estimates the number of die locations that require inking, as compared to the manual process.

TABLE II
MEAN ACCURACY AND ERROR RATES

Accuracy	False positives	False Negatives
96.1%	0.83%	3.07%

As mentioned above, the overall location of the inked die clusters is accurately predicted for the majority of wafers. This observation allows us to use the erosion and dilation methods, in order to bias our predictions towards a more aggressive or more conservative inking strategy. An example of this post-prediction processing methodology is shown in Figure 7, where the manually and automated ink patterns are presented. In Figure 7b., the automatically inked locations are depicted in orange, the die which have been added to the predicted pattern after the erosion-dilation process are depicted in green, and the single die that has been correctly removed from the pattern due to erosion is depicted in red.

V. FUTURE WORK

Towards improving the accuracy of this pattern recognition-based approach, we intend to further investigate the following directions:

- Identify and enhance the training features in order to capture the rarest failure types.
- Enrich the training dataset with more scratch type patterns in order to enable their accurate recognition.
- Leverage the actual specification test values in order to further improve model accuracy.
- Perform burn-in tests on a number of wafers in order to improve the quality of the learned model, as well as to better evaluate the effectiveness of the proposed method

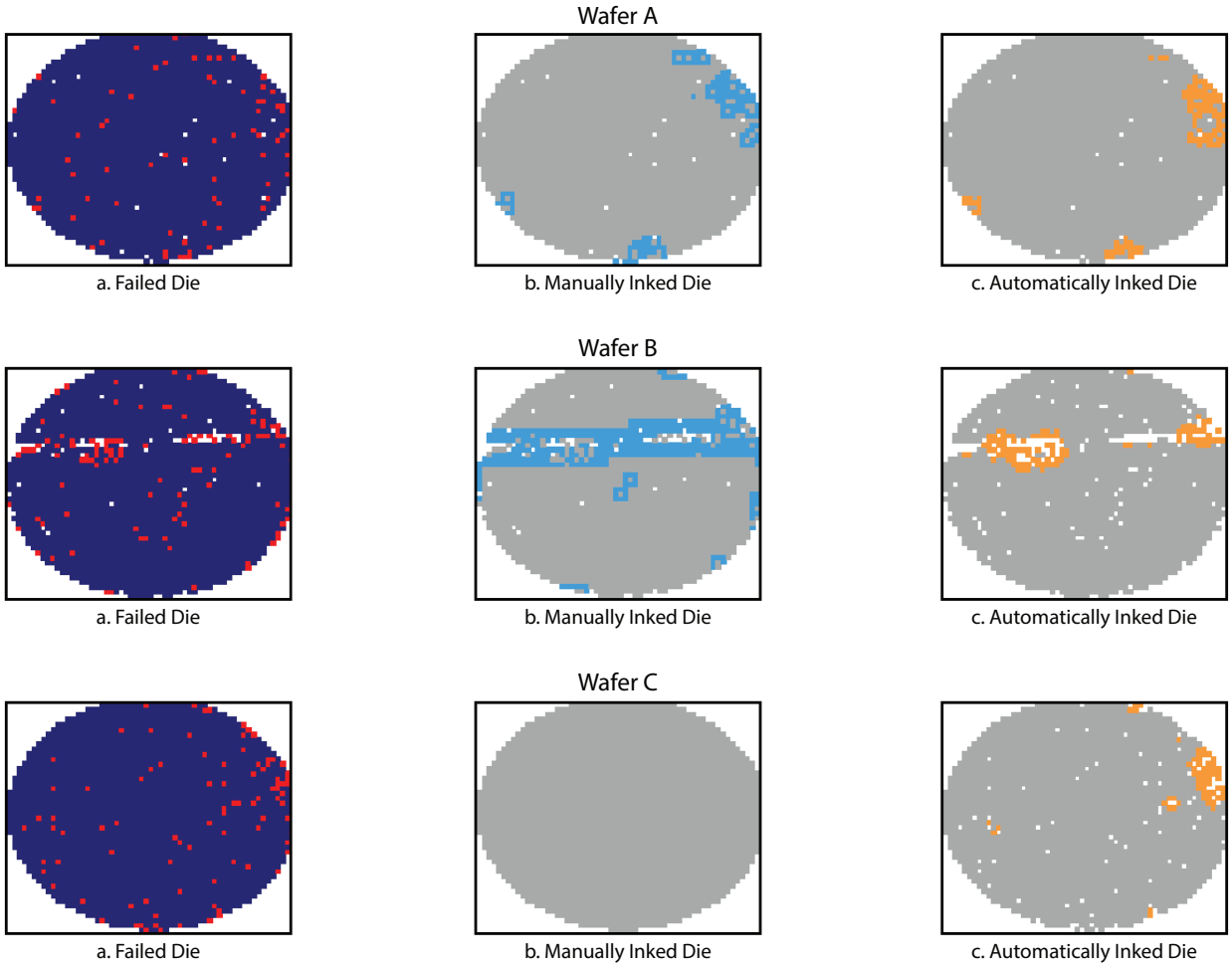


Fig. 8. Manual and automated wafer inking example

against ground truth information regarding die that are subject to infant mortality.

VI. Conclusions

Wafer-level failing patterns have been shown to be significant indicators of the existence of systematic defects and process shifts. While recent research has considered automatic identification and classification of such patterns, most decisions are still performed manually by process engineers. Marking of the devices that have a high probability for early-life defect manifestation remains a laborious non-automated procedure. Towards alleviating this burden, in this work we introduced a pattern recognition-based methodology for predicting the inking patterns, by only utilizing the failure wafer maps. Using industrial data, we successfully trained an SVM classifier and achieved highly accurate pattern predictions.

References

[1] Y. C. Kuang M. P. L. Ooi, E. Kwang Joo Sim, C. Chan L. Kleeman, and S. Demidenko, "Automatic defect cluster extraction for semiconductor wafers," *IEEE Instrumentation and Measurement Technology*, 2010.

[2] S. P. Cunningham and S. MacKinnon, "Statistical methods for visual defect metrology," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 1, pp. 48–53, 1998.

[3] F. L. Chen and S. F Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 3, pp. 366–373, 2000.

[4] C. Chan Y. C. Kuang J. W. Cheng, M. P. L. Ooi and S. Demidenko, "Evaluating the performance of different classification algorithms for fabricated semiconductor wafers," *IEEE International Symposium on Electronic Design, Test, and Application*, pp. 360–366, 2010.

[5] W. Kuo T. Yuan and S. J. Bae, "Detection of spatial defect patterns generated in semiconductor fabrication processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 3, pp. 392–403, 2011.

[6] L. C. Wang N. Sumikawa and M. S. Abadir, "A pattern mining framework for inter-wafer abnormality analysis," *IEEE International Test Conference (ITC)*, pp. 1–10, 2013.

[7] F. Wang M. B. Alawieh and X. Li, "Identifying systematic spatial failure patterns through wafer clustering," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 910–913, 2016.

[8] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[10] J. C. Russ, *Image Processing Handbook, Fourth Edition*, CRC Press, Inc., 4th edition, 2002.