Machine/Deep Learning Applications Using the V93000 and Nvidia Jetson TX2

Keith Schaub, Ira Leventhal and Brian Buras - Advantest

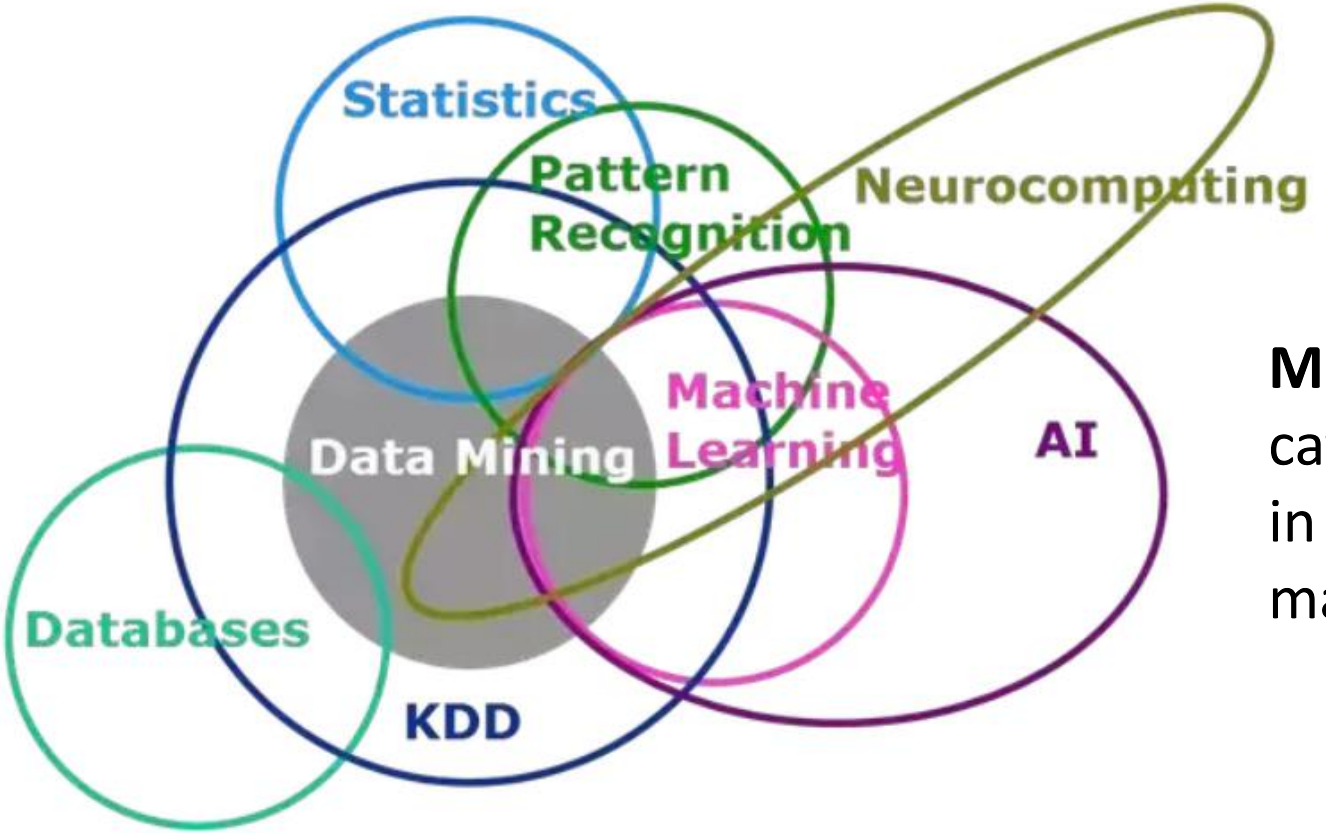
Gerard John - Amkor

Yiorgos Makris - UTD

Outline

- **AI Machine Learning / Deep Learning Overview**
- **Problem Statement**
- **Test Compaction:** Hypothesis 1 – Machine learning algorithms analyze test data to optimize the test list.
- **Dynamic Spatial Testing:** Hypothesis 2 – Machine learning algorithms learn wafer spatial correlations to dynamically optimize test coverage
- **Test Compaction**
  - Process / Data Analysis
  - Results
  - Conclusions
- **Dynamic Spatial Testing**
  - Process / Data Analysis
  - Results
  - Conclusions
- **Summary**
- **Next Steps**
- **Machine Learning Image Classifier Integrated into the V93000 Environment (Kiosk)**
- **Future Considerations**

Machine Learning Overview



Machine Learning is an AI sub-category focused on finding patterns in data and using those patterns to make predictions

Image source: https://www.quora.com/Whats-the-difference-between-the-terms-machine-learning-deep-learning-and-AI

Machine Learning Training

Input, feed a lot of data

Machine Learns patterns in the data

MODEL

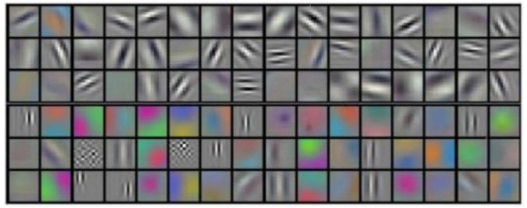
"OK, I see the patterns and understand the data now"

Machine Learning Training Example

Input a bunch of Chihuahuas

Machine Learns to recognize Chihuahua patterns

MODEL



"hmm, ok I learned what Chihuahuas look like"

- Pointed ears
- Small typically dark nose
- Little beady eyes
- …

Disclaimer: No dogs were harmed as part of this presentation

Chihuahua or Muffin?

Input Chihuahuas and “non Chihuahuas”

Algorithm applies Chihuahua model to classify

MODEL

Classification Result

“You didn’t train me what a muffin looks like?!”

Input training data is important!

Puppy or Bagel?

Sheepdog or Mop?

Labradoodle or fried chicken?

Problem Statement

- **Testing complexity and test cost continues to increase**
  - Quality is the new Cost
  - More testing
  - Multiple domain types and insertions needed
  - Need to avoid longer test times
  - Need to minimize test costs
- **Process variations are not static, yet testing methodologies typically are static**
  - Same tests applied throughout device life cycle
  - Engineers manually adjust
    - Laborious, tedious, “after the fact; late”
      - Negatively impacts quality
      - TONS of data, but humans are not efficient at analyzing it

VOICE 2018

Test Compaction – Hypothesis 1

Can a machine learning algorithm learn measurement correlations to automatically optimize testing metrics?

Pair Name	NE_CORNER	NW_CORNER	1VIA_NETWORK	CRACK_GPU1_NW	CRACK_GPU1_SW	CRACK_GPU1_SE	CONTROLVIA_NETWORK	CRACK_GPU1_NE	TSV_CHAIN_GPU1	10E_Sanity	SE_CORNER	2VIA_NETWORK2	CRACK_GPU2_SE	CRACK_GPU2_NW	2VIA_NETWORK	SW_CORNER	CRACK_GPU2_NE	CRACK_GPU2_SW	10E_Sanity	1K_Sanity	TESTTIME	RESULT
	55	55	55	10	10	10	55	10	100	8	55	55	10	10	55	55	10	10	8	910		
Die #	100	100	100	100	30	30	100	30	350	18	100	100	30	30	100	100	30	30	18	1040		
1	76.47	79.77	89.18	20.14	19.82	20.49	77.94	19.86	267.5	9.86	-12186130	21120700	20.86	20.85	90.26	79.9	20.79	21.56	10.04	1014.67	2.57	FAIL
2	75.84	78.88	88.27	19.97	19.76	20.45	77.19	19.65	269.8	9.85	80.16	89.28	20.69	20.78	90.81	80.73	20.68	21.52	10.07	1014.68	2.54	PASS
3	-34769800	78.15	87.43	19.93	19.58	20.33	-46666836	19.55	269.03	9.84	80.28	89.4	20.59	20.66	90.5	80.31	20.62	21.46	10.06	1014.88	2.57	FAIL
4	76.75	79.5	89.25	20.17	19.52	20.46	78.3	20.03	270.44	9.85	-13827451	-79996080	20.34	20.82	90.22	79.83	20.85	20.8	10.05	1014.57	2.56	FAIL
5	76.61	79.3	88.83	20.08	19.63	20.65	78.24	19.94	271.9	9.88	79.54	89.64	20.63	20.85	90.17	79.56	20.89	21.48	10.07	1014.94	2.54	PASS
6	76.86	79.8	89.41	20.22	19.83	20.62	78.5	19.97	272.13	9.78	80.18	90.66	20.95	21.35	91.63	80.98	20.9	21.6	10.09	1014.71	2.54	PASS
7	76.24	78.73	88.4	20.18	19.62	20.56	77.83	19.86	271.22	9.86	80.04	90.47	20.93	21.08	91.53	80.59	20.82	21.65	10.12	1014.79	2.53	PASS
8	75.5	79.5	88.88	20.25	19.68	20.43	77.15	19.76	271.51	9.83	79.56	89.6	20.71	21.09	89.44	79.6	20.78	21.4	10.08	1014.7	2.52	PASS
9	75.55	79.35	88.33	19.94	19.68	20.3	77.17	19.76	269.55	9.89	78.83	89.39	20.72	20.73	89.4	78.98	20.6	21.31	10.07	1014.69	2.53	PASS
10	75.69	78.87	88.17	19.82	19.05	20.17	76.57	19.44	265.58	9.8	78.6	89.04	20.79	20.49	89.22	79.2	20.4	21.16	10.09	1014.77	2.53	PASS
11	76.49	78.31	87.97	19.97	19.48	20.44	78.03	19.86	273.4	9.87	-130742056	23536378	-11954473	20.72	88.79	78.41	20.69	20.95	10.11	1014.77	2.54	FAIL
12	76.47	78.49	88.21	20.02	19.69	20.47	77.9	19.67	275.08	9.83	80.34	90.82	20.88	20.76	89.97	80.53	20.8	21.52	10.06	1015.03	2.53	PASS
13	76.31	78.04	87.65	20.05	19.72	20.39	77.82	19.77	272.97	9.88	79.92	90.42	20.72	20.78	91	80.55	20.71	21.57	10.01	1014.89	2.54	PASS
14	75.67	78.08	87.6	19.86	19.74	20.35	77.18	19.5	271.74	9.92	80.18	90.74	20.81	20.78	90.5	80.08	20.73	21.5	9.96	1014.75	2.54	PASS
15	74.76	77.68	87.08	19.93	19.69	20.28	76.39	19.45	270.45	9.87	79.73	90.07	20.68	20.72	90.1	79.53	20.58	21.34	10.12	1014.68	2.54	PASS
16	75.7	78.2	87.55	19.77	19.65	20.2	77.26	19.71	270.42	9.86	78.96	89.53	20.71	20.92	89.95	79.4	20.51	21.26	10.16	1014.85	2.53	PASS
17	76.21	78.77	88.28	20.15	19.58	20.36	77.56	19.62	270.31	9.87	80.26	90.68	20.86	20.8	89.62	79.91	20.64	21.37	10.1	1014.63	2.54	PASS
18	75.68	79.17	88.38	20.26	19.75	20.31	77.01	19.63	269.69	9.84	80.22	90.33	20.83	20.89	90.14	79.98	20.62	21.45	10.07	1014.67	2.54	PASS
19	74.13	79.22	88.46	19.98	19.75	20.34	75.84	19.53	266.17	9.86	79.43	89.74	20.69	20.79	89.6	79.17	20.68	21.38	10.12	1014.66	2.54	PASS
20	76.47	78.72	88.86	19.95	19.88	20.57	78.07	19.9	275.3	9.79	80.22	90.67	20.92	20.9	90.08	79.43	20.84	21.59	10.1	1014.8	2.54	PASS
21	76.82	79.54	89.41	20.11	20.06	20.62	78.49	20	274.55	9.86	80.26	90.93	20.96	20.97	90.69	80.51	20.97	21.77	10.04	1014.68	2.54	PASS
22	75.45	78.8	88.59	20.05	19.76	20.33	77.21	19.73	273.26	9.9	80.28	91.08	20.85	20.91	90.3	80.02	20.75	21.68	10.04	1014.62	2.54	PASS
23	74.96	76.93	86.62	19.81	19.79	20.24	76.67	19.65	272.47	9.85	78.79	89.49	20.74	20.89	89.62	79	20.55	21.51	10.07	1014.78	2.53	PASS
24	74.32	77.48	87.22	19.81	19.52	20.08	76.01	19.37	271.18	9.88	77.73	88.6	20.49	20.73	89.27	78.04	20.34	21.14	10.05	1014.8	2.54	PASS

Machine Learning Algorithm Trains

Determine if some of these are highly correlated!

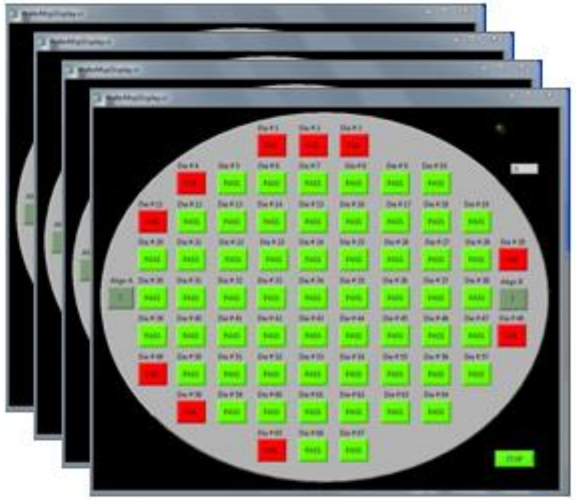
Pair Name	NE_CORNER	NW_CORNER	1VIA_NETWORK	CRACK_GPU1_NW	CRACK_GPU1_SW	CRACK_GPU1_SE	CONTROLVIA_NETWORK
	55	55	55	10	10	10	55
Die #	100	100	100	100	30	30	100
1	76.47	79.77	89.18	20.14	19.82	20.49	77.94
2	75.84	78.88	88.27	19.97	19.76	20.45	77.19
3	-34769800	78.15	87.43	19.93	19.58	20.33	-46666836
4	76.75	79.5	89.25	20.17	19.52	20.46	78.3
5	76.61	79.3	88.83	20.08	19.63	20.65	78.24
6	76.86	79.8	89.41	20.22	19.83	20.62	78.5
7	76.24	78.73	88.4	20.18	19.62	20.56	77.83
8	75.5	79.5	88.88	20.25	19.68	20.43	77.15
9	75.55	79.35	88.33	19.94	19.68	20.3	77.17

Use the newly organized subset

Show test results just as accurately
Show same quality
Show reduced cost impact

Dynamic Spatial Machine Learning of Wafer Testing – Hypothesis 2

Can a machine learning algorithm learn spatial correlations to automatically optimize testing per die?



Machine Learning Algorithm Trains

Trained Model

Predicted test results

Apply Model to predict result

Specification Test Compaction Concept

T: Total set of n tests

S: S $\subset$ T (Subset of k tests)

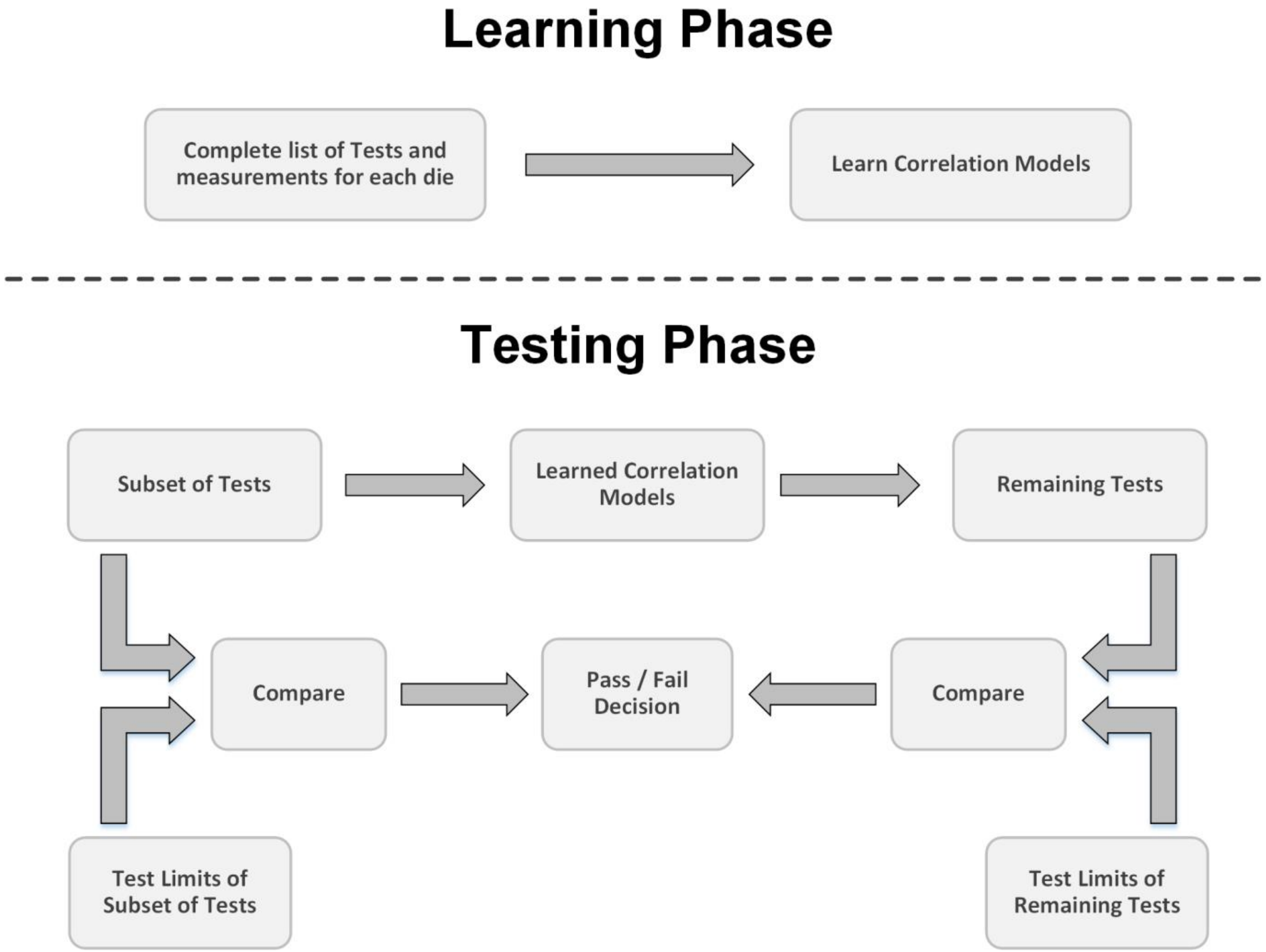
Et: Number of test escapes for test t

The objective is to minimize the size of S while maintaining a low

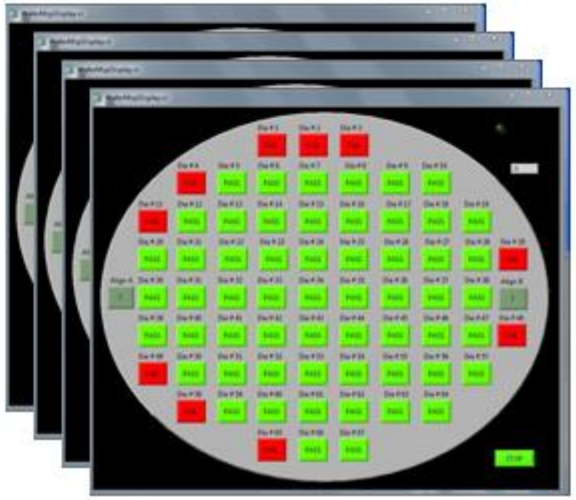
$$\sum_{i=0}^{k} E_i$$

where Ei is the test escape for i^{th} test in S.

Different sizes of S can be produced depending on what the acceptable escape rate is.



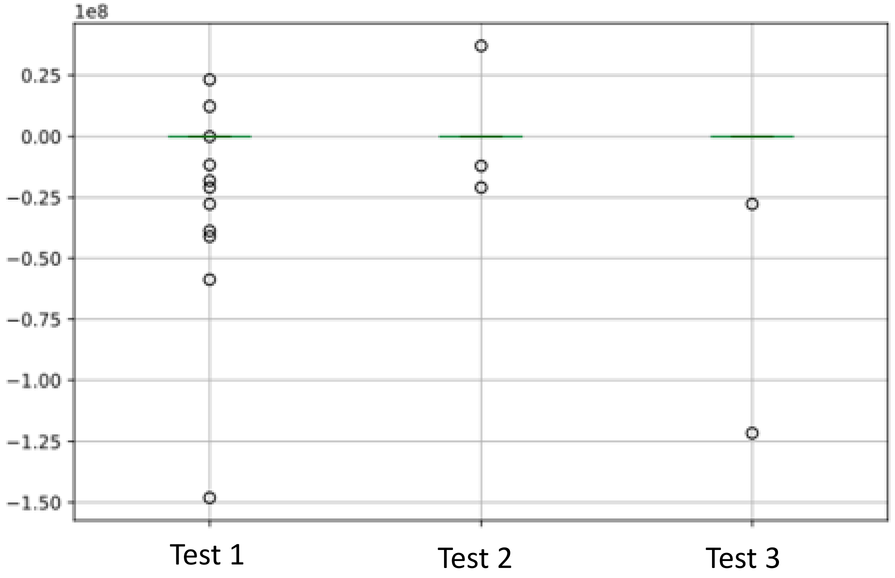
Test Compaction – Data Description & Idiosyncrasies



- Dataset contains 6 wafers, with 20 test measurements
- There are 30 failing die out of 402
- Small number of die locations per wafer
- No test groups or test times

Wafer #	Pass count	Fail count
1	58	9
2	60	7
3	57	10
4	65	2
5	67	0
6	65	2

Test Compaction – Pre-filter

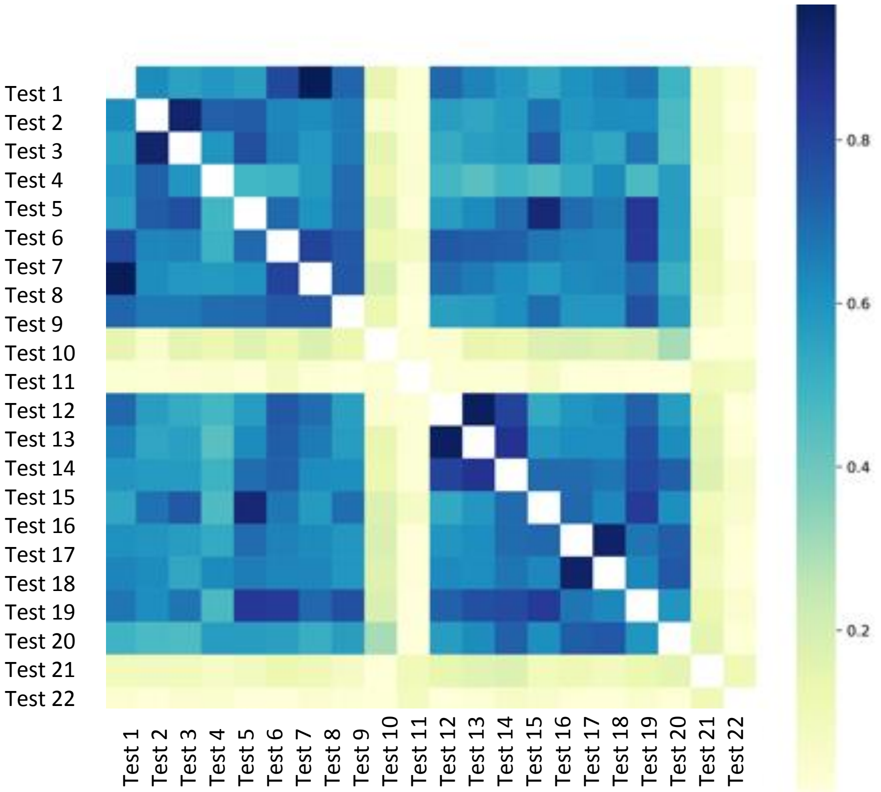


Clean the Data (labradoodle or fried chicken?)
An important pre-step to training the model is to clean up the data before it is fed to the training algorithm.

We removed outliers before training the algorithm

Test Correlation

High correlation



- Bi-variate correlation of all test pairs using absolute values of Pearson Correlation Coefficients (PCC).
  - This shows the degree by which two variables co-vary
- Multi-variate non-linear regression modeling is a more suitable technique for discovering correlations between tests.

Low correlation

Correlation Results

Yellow = low correlation

Blue = high correlation

Many measurements are highly correlated!

Test Correlations

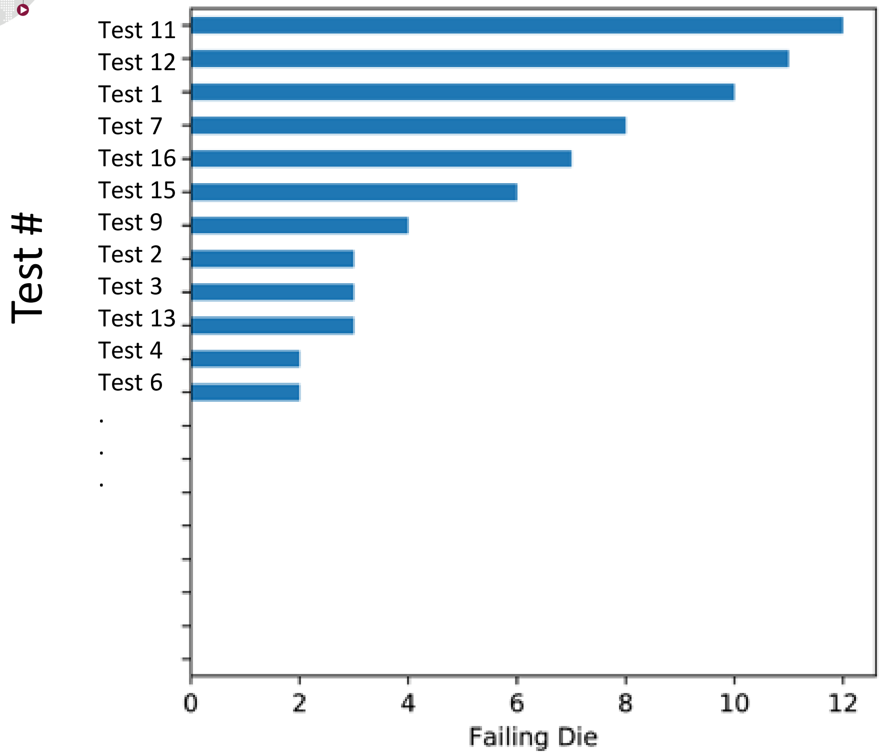
- Multi-variate Adaptive Regression Splines (MARS)[1] is a non-linear regression analysis methodology
- Training consists of two phases that aim to select the optimal number of features:
  - **Forward pass**: Starting with the intercept term and progressively adds a basis function that minimizes the prediction error. This usually generates an overfit model
  - **Backward pass**: This stage prunes the basis functions using a metric that penalizes the model based on the number of features

[1] Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". *The Annals of Statistics*.

Test Correlations

- Description of the MARS-based experiment:
  - Train a MARS model for every test in the dataset and calculate the accuracy of the model using a hold-out set of wafers
  - Identify the most accurately modeled tests based on the prediction error
- Most accurately predicted tests: Test 1, Test 2, Test 3, Test 7 , Test 11 Test 15, Test 16
- For this experiment the python implementation of MARS (pyearth) was used

Test Compaction & Reordering – Trained algorithm suggests subset of tests



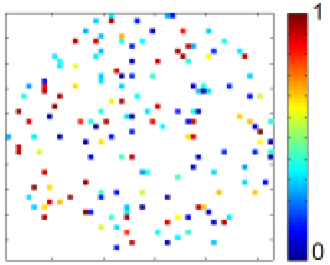
- Greedy Algorithm for test compaction:
  - Start by including the test that captures the most failing devices. Test 11 in our dataset
  - Iteratively add the test that minimizes the test-escapes. This can skip tests based on the overlap
- e.g. tests that capture all 30 failing die are: Test 1, 3, 5, 8, 7
- Algorithm suggests to use these 5 tests
- Algorithm could automatically re-order tests to optimize test flow (i.e. learn and apply most efficient tests and optimize test flow)
- Test time savings reduces cost

Other algorithm examples: Support vector machines, decision trees, neural networks

Dynamic Spatial Machine Learning of Wafer Testing

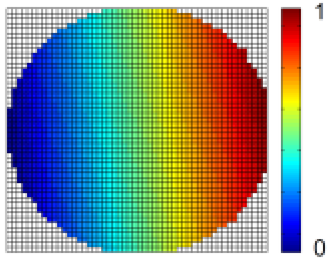
Spatial decomposition of wafer measurements

$$g(x,y) = \alpha_1 b_1(x,y) + \cdots + \alpha_{n_b} b_{n_b}(x,y)$$

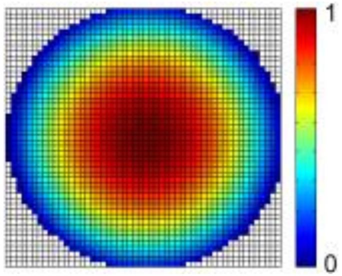


=

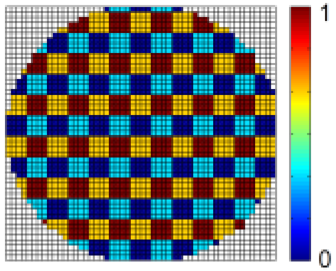
α_1



$+\alpha_2$



$\ldots + \alpha_{n_b} b_{n_b}(x,y)$



$$b_1(x,y) = ax + by$$

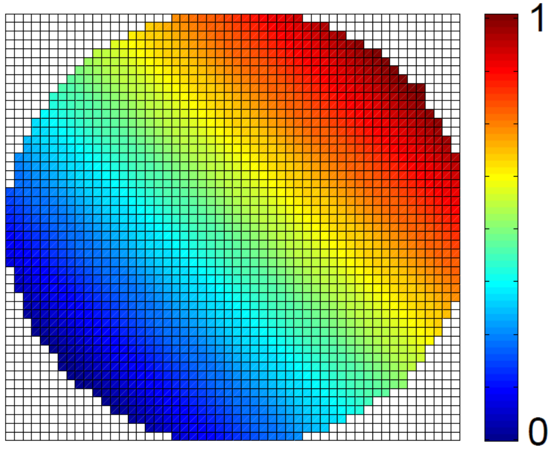
$$b_2(x,y) = \cos\left(\frac{n2\pi}{d_u}\right) r$$

$$b_n(x,y) = \left(\frac{1}{k}\right) * m_k$$

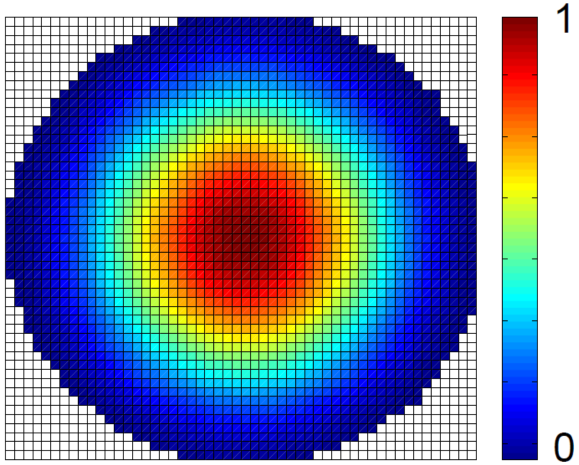
Learn these functions from the data…

*K. Huang, N. Kupp, J. Carulli, and Y. Makris, "Process Monitoring through Wafer-level Spatial Variation Decomposition," ITC 2013

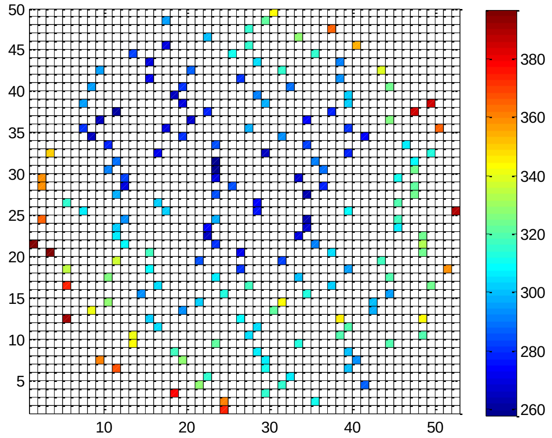
Examples of spatial basis functions

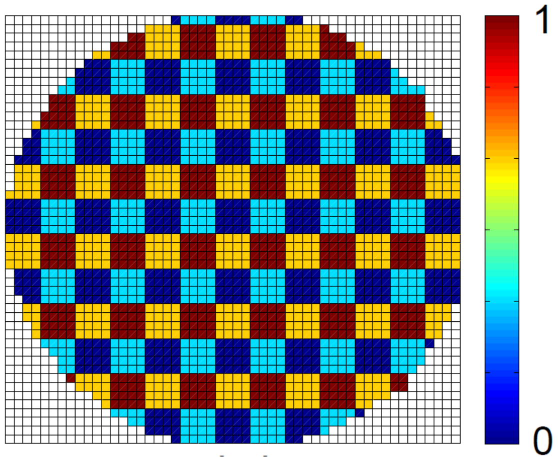


Linear

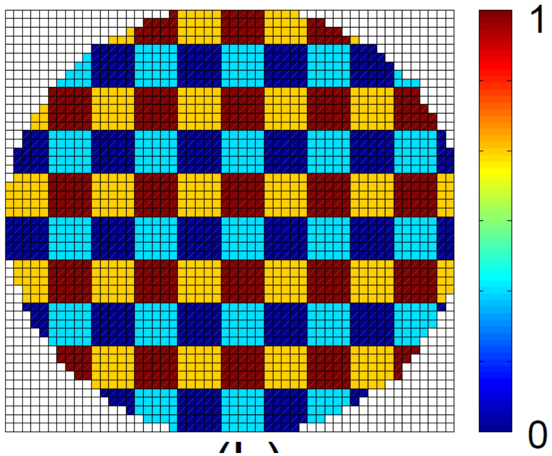


Radial





Checkerboard #1



Checkerboard #2

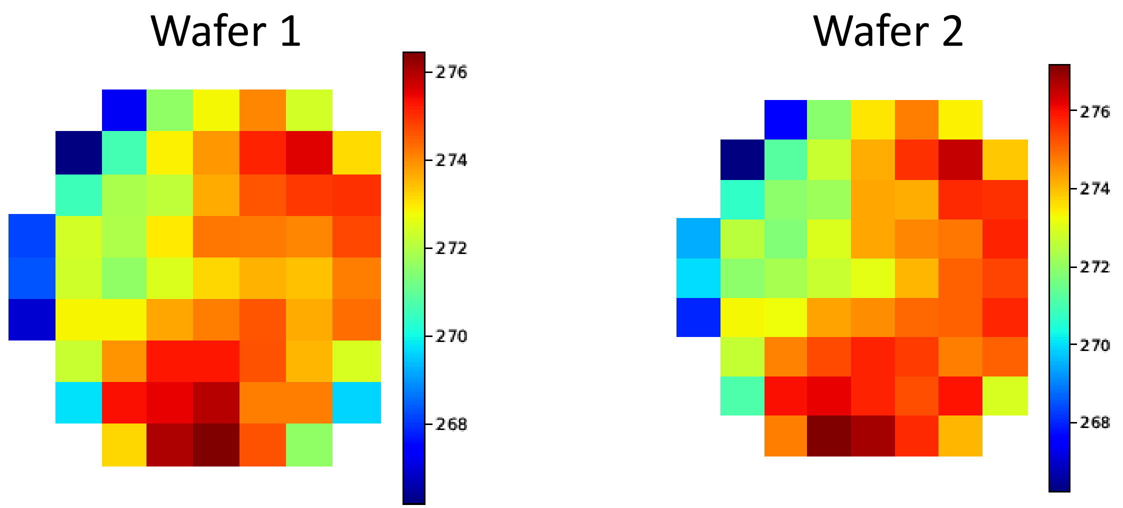
$$A = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]\ ?$$

Basis function learned using domain-specific knowledge

VOICE 2018

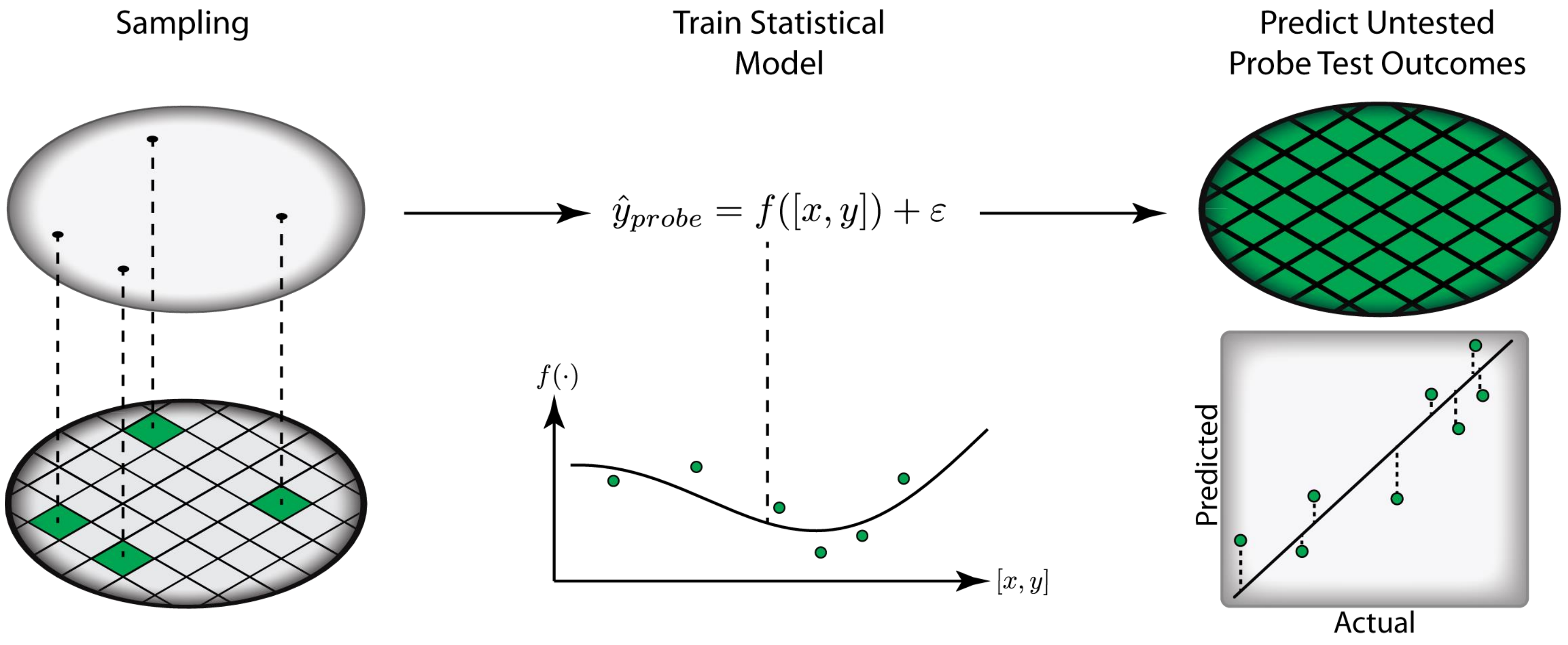
Algorithm Learns Spatial Correlation Pattern

- Spatial correlation refers to the relationship that certain test measurements have as a function of the die locations
- One way to identify such wafer-level spatial correlations is to perform visual inspection on the wafer maps of each test.



Spatial Correlation Modeling

- In our experiments we performed spatial-correlation modeling using Gaussian processes[2]

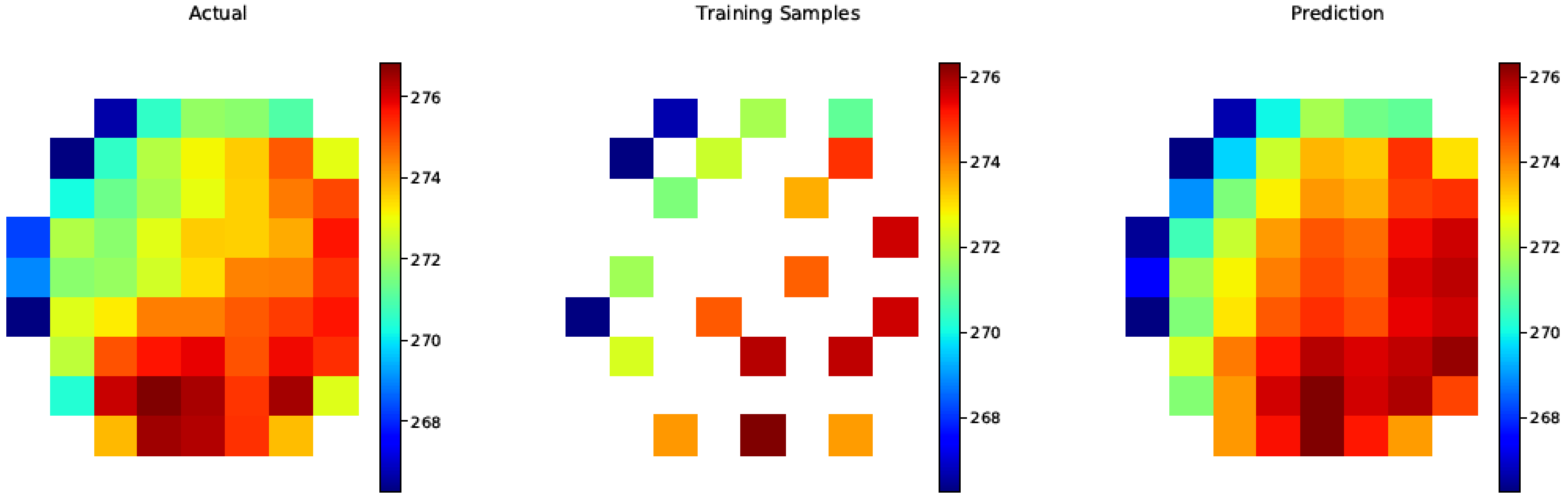


[2] N. Kupp, K. Huang, J. Carulli, Y. Makris, "Spatial Estimation of Wafer Measurement Parameters Using Gaussian Process Models", *Proceedings of the IEEE International Test Conference (ITC)*

Spatial Correlation Accuracy Results

- Spatial correlation modeling example on Test 9
- Relative prediction error = 0.4%

$$Relative\ Error = \left| \frac{(acutal\ - predicted\ )}{actual} \right|$$



[2] N. Kupp, K. Huang, J. Carulli, Y. Makris, "Spatial Estimation of Wafer Measurement Parameters Using Gaussian Process Models", *Proceedings of the IEEE International Test Conference (ITC)*

Summary

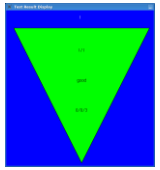
- Both hypothesis were shown to be true
  - Machine learning algorithms can automatically learn test optimization techniques by analyzing the data
    - They can learn which tests are most important
    - They can automatically generate the relevant/sub-set test list
    - They can automatically optimize the test flow by re-organizing the test list
  - Machine learning algorithms
    - Can find correlations and dependencies in the data
    - Use that information to optimize testing and lower test cost
    - Example: the foreknowledge could be used to eliminate re-testing

Next Steps

- Apply same methods to multiple and larger data sets
- Integrate machine learning technique into the SmarTest environment
- Develop an AI V93000 demonstration using Nvidia's Jetson[3] 256 core AI environment
  - Kiosk Demo – AI ML Jetson 2 TX - operating within smartest that classifies smartphone display images

[3] https://developer.nvidia.com/embedded/buy/jetson-tx2

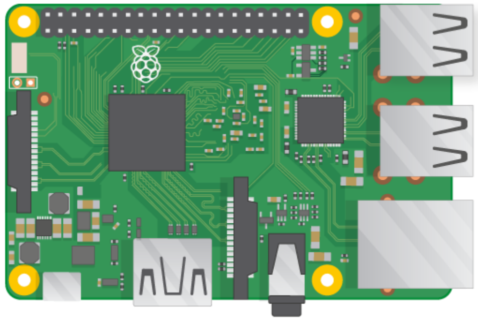
Machine Learning V93000 Environment



V93000 testflow selects image
Compares to classification.

radb control select image

Send classification
To V93000 workstation



Jetson2 camera identifies image

```
// Classify
int imageNet::Classify( float* rgba, uint32_t width, uint32_t height, float* confide
{
    for( size_t n=0; n < mOutputClasses; n++ )
    {
        const float value = mOutputs[0].CPU[n];

        if( value >= 0.4f ) {
            printf("class %04zu - %f  (%s)\n", n, value, mClassDesc[n].c_str());
            sendResult(n,value, mClassDesc[n]);
```

Visit **AI Kiosk** for demo

VOICE 2018

Future Considerations

- Develop Machine Learning APIs for the SmarTest that customers could use from a library
- Develop similar APIs for the Nvidia Jetson II AI environment that could be controlled from SmarTest environment
  - Customers would have a 256 core AI environment that they can build their own models

Measure the Connected World

And Everything in It SM

Thank You.

VOICE 2018

ADVANTEST®

V93000-379-HT - Machine/Deep Learning Applications Using the V93000 and Nvidia Jetson TX2

San Diego

VOICE 2018