Wafer Map Failure Pattern Recognition Using Artificial Intelligence

Presenter : Andy Wee

- GLOBALFOUNDRIES

Authors : Lo Wee Tick

- GLOBALFOUNDRIES

Hui Kong Meng

- ISS-NUS Master Student

GLOBALFOUNDRIES

NUS National University of Singapore

ISS Institute of Systems Science

SWTW

IEEE SW Test Workshop

Semiconductor Wafer Test Workshop

June 10 - 13, 2012 | San Diego, California

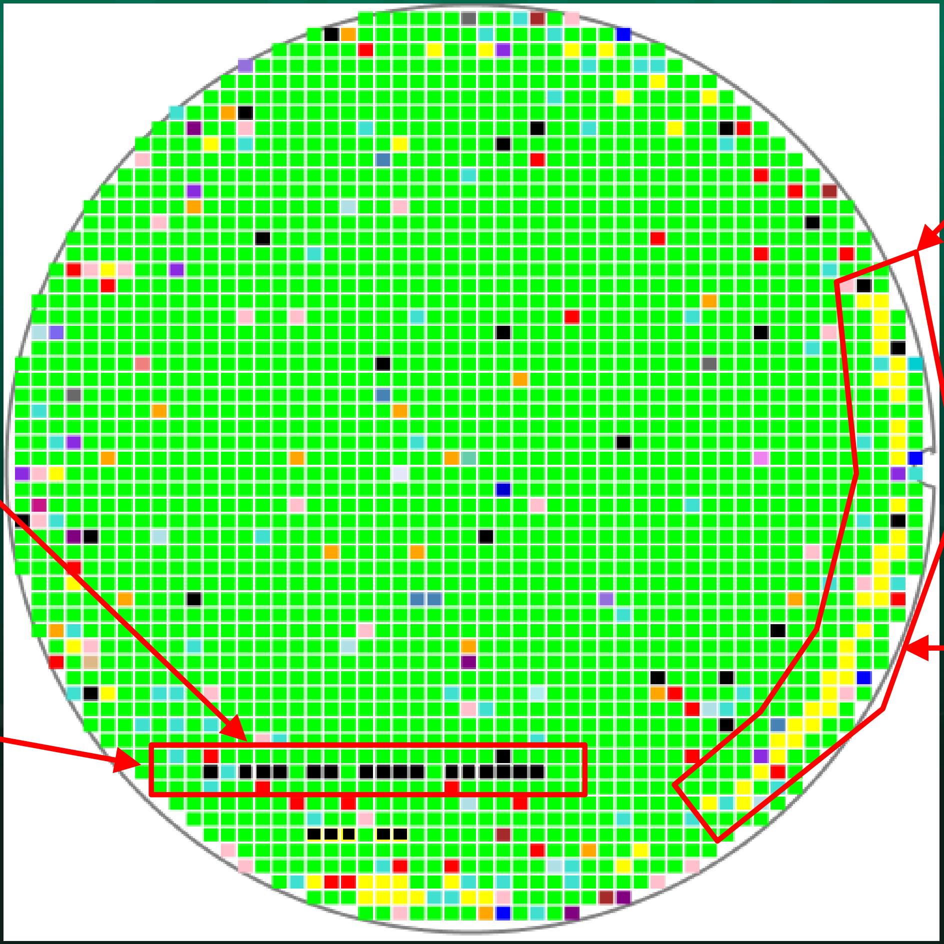
Background

- **Probing methodology getting more complex with :**
  - Increased parallelism
  - Optimized probing path to reduce step indexing time

- **Adds challenges to wafer disposition :**
  - Unnecessary retesting of wafers
    - Leading to loss of test capacity
    - Hardware pre-mature end-of-life (EOL)
    - Added cost to overall test operation
  - Failure to recover overkill dies
    - Revenue leak due to throwing away potential good dies
  - Highly dependent on past experiences
    - Lost domain knowledge
    - Time consuming

Background

- **Disposition challenges**



Single Site
- Possible
probing issues

Multi-Site
- Could be
wafer issues

Single Site
- Possible
wafer issues

Multi-Site
- Could be
probing issues

Background

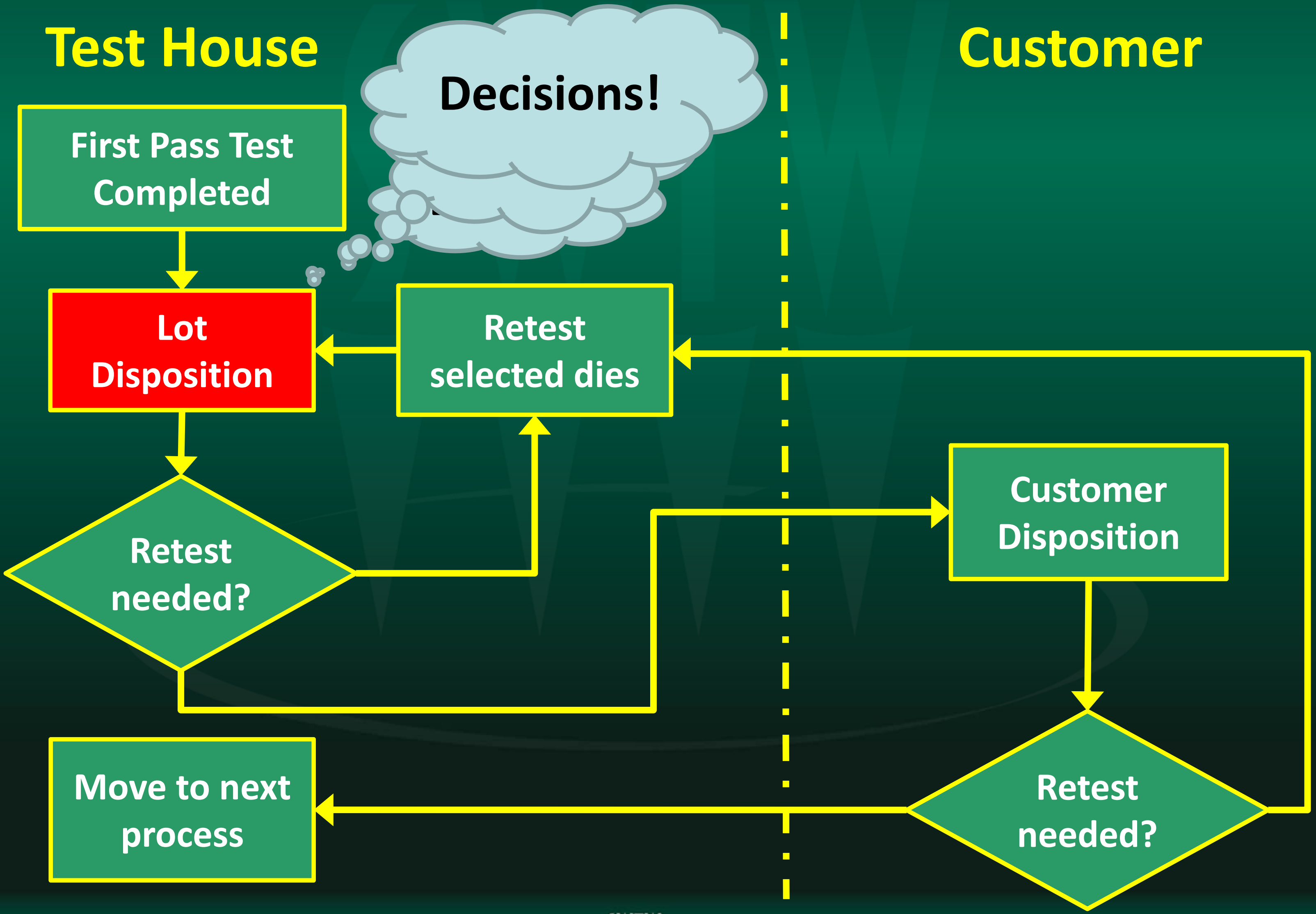
- **What are we trying to solve?**
  - Recognition / Decision problem!!

Hunting Season!!!

Objectives

- **Create a tool to help recognize wafer failure pattern using AI (Artificial Intelligence) to help :**
  - Wafer test disposition decision
  - Reduce wafer test results review time
  - Retain knowledge gained
  - Create a more systematic way for wafer disposition process

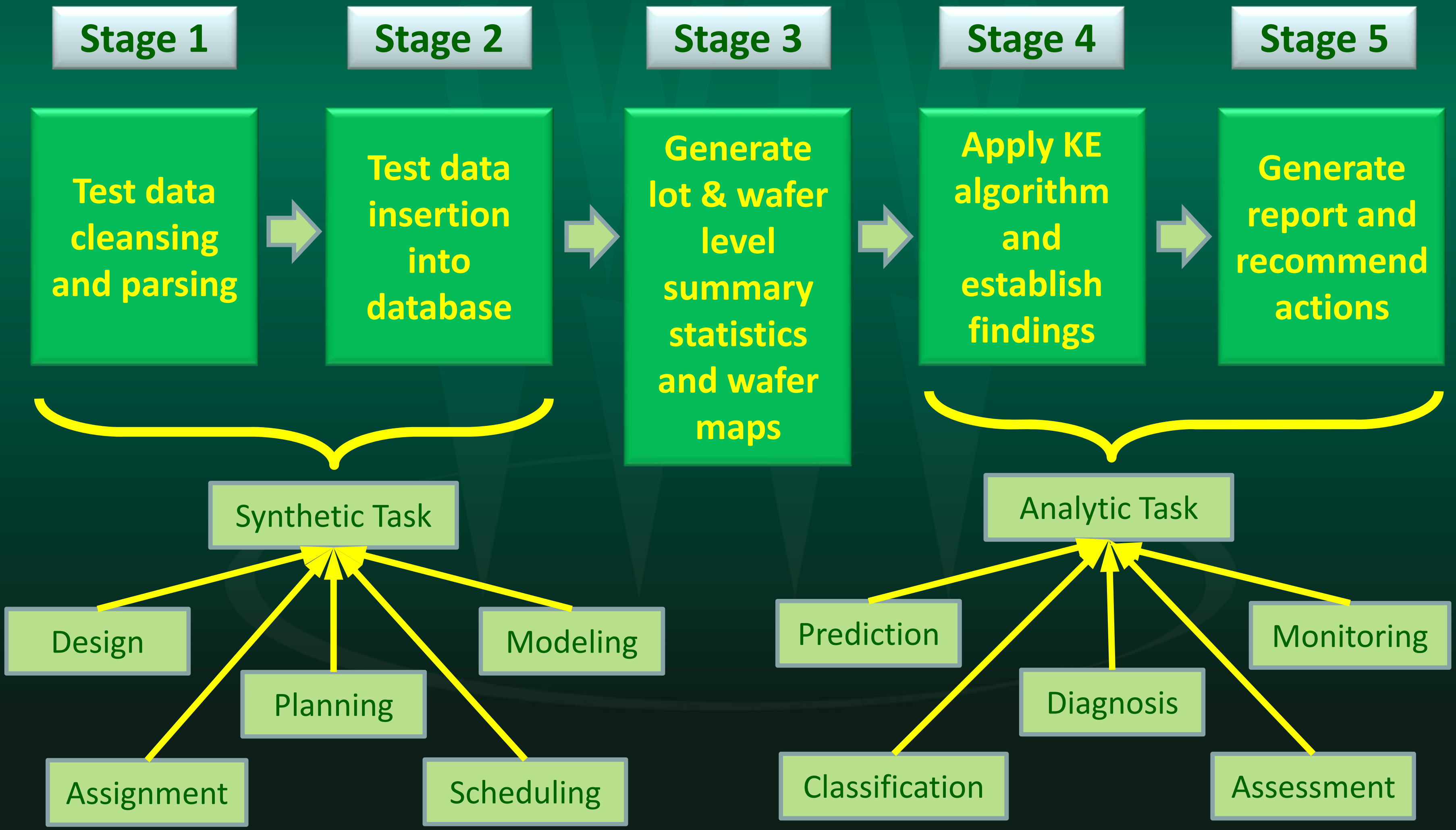
Typical Outsourced Disposition Flow



Approach

- Extract useable parameters from available test data to compute statistical information.
- Create database for data insertion
- Select suitable AI methodology for model learning e.g. NN (Neural-Network) supervised learning, to identify recoverable failures detect various pattern failure scenarios.
- Develop novel algorithms to detect various pattern failure scenarios.
- Develop a rule-based algorithm on logical assessment of information.

Development overview



Stage 1

Test data cleansing and parsing

Stage 2

Test data insertion into database

Stage 3

Generate lot & wafer level summary statistics and wafer maps

Stage 4

Apply KE algorithm and establish findings

Stage 5

Generate report and recommend actions

Synthetic Task

Design

Planning

Modeling

Assignment

Scheduling

Analytic Task

Prediction

Diagnosis

Monitoring

Classification

Assessment

Phase 1 : Data preparation (1)

- **Understanding the data**
  - Decision to use **STDF** (Standard Test Data Format) that is readily available in all major ATEs
  - Decode the STDF binary format to sieve out required data
- **Cleansing the data for database insertion**
  - Filter out flawed data e.g. missing information, misplaced fields etc.
- **Mining the data**
  - Identify critical parameters and compute statistical data as input nodes
  - Select training data sets based on the disposition scenarios
  - Select test data sets for verification

Phase 1 : Data preparation (2)

- **Some example of useful data**
  - To detect system failure, trouble spots pattern

Input	Detection	Verification	Output
1. X-Y coordinates 2. Soft Bins	1. Start with each failure die as seed. 2. Cluster failing dies to determine if there is grouping beyond set a threshold.	1. Repeated signature in some wafers, adjacent wafers. 2. Independent of tester or H/W used, cluster will still fail.	Cluster failure True / False? Quadrant location?

  - To detect "Known Recoverable Bins"

Input	Detection	Verification	Output
Soft Bins	Recoverable list of soft bins	Recovery history on different wafers, first probe, online reprobe etc.	Probability of subjecting to retest

Phase 1 : Data preparation (3)

- **Some example of useful data**
  - To detect Parametric signature

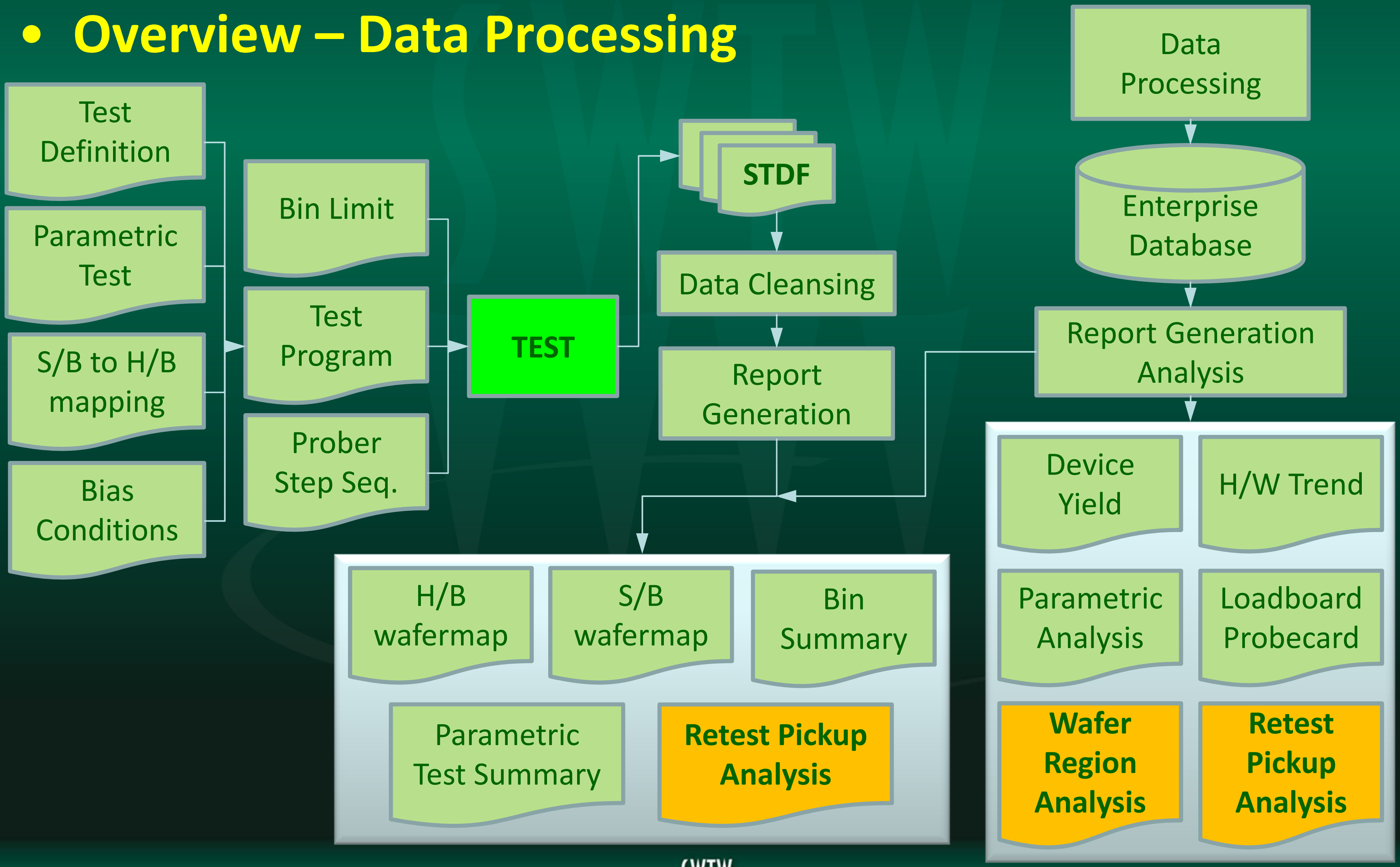
Input	Detection	Verification	Output
1. X-Y coordinates 2. Bin failure 3. Failing parametric value	Map failing parametric value to population (?)	Test values recorded near limits?	Probability of recovery after retest.

Phase 2 : System selection

- Modular system upon which new KE (Knowledge Engineering) techniques can be added.
- Self-learning system upon which new models can be added.
- Configurable with different rules for different devices, different knowledge.
- Configurable thresholds to detect various failure patterns.
- Web-based reports, easier access.
- Database approach for scalable repository.

Phase 3 : System structure (1)

- **Overview – Data Processing**



Test Definition

Parametric Test

S/B to H/B mapping

Bias Conditions

Bin Limit

Test Program

Prober Step Seq.

TEST

STDF

Data Cleansing

Report Generation

H/B wafermap

S/B wafermap

Bin Summary

Parametric Test Summary

Retest Pickup Analysis

Data Processing

Enterprise Database

Report Generation Analysis

Device Yield

H/W Trend

Parametric Analysis

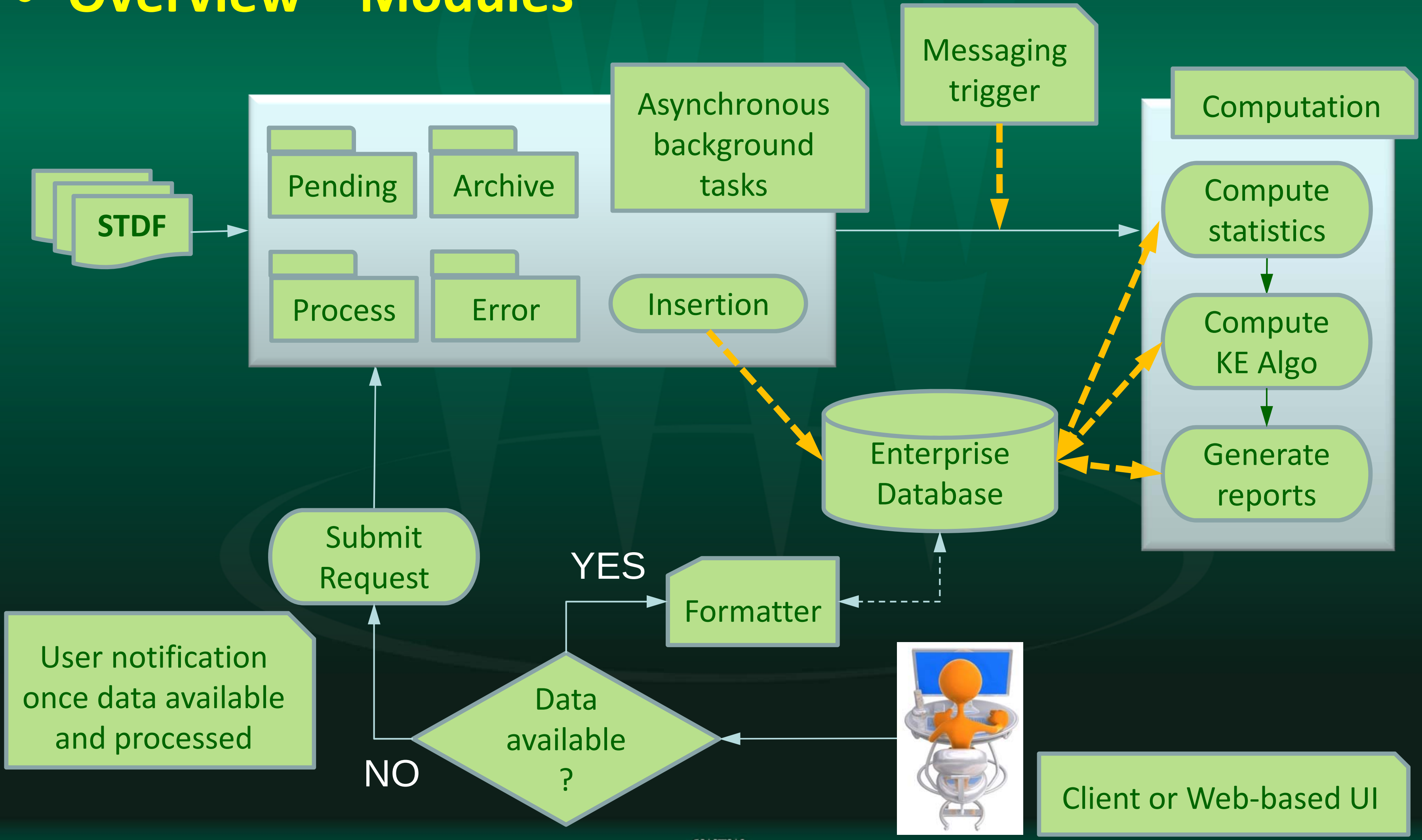
Loadboard Probecard

Wafer Region Analysis

Retest Pickup Analysis

Phase 3 : System structure (2)

- **Overview – Modules**



Messaging trigger

Asynchronous background tasks

Computation

STDF

Pending

Archive

Process

Error

Insertion

Compute statistics

Compute KE Algo

Generate reports

Enterprise Database

Submit Request

YES

Formatter

User notification once data available and processed

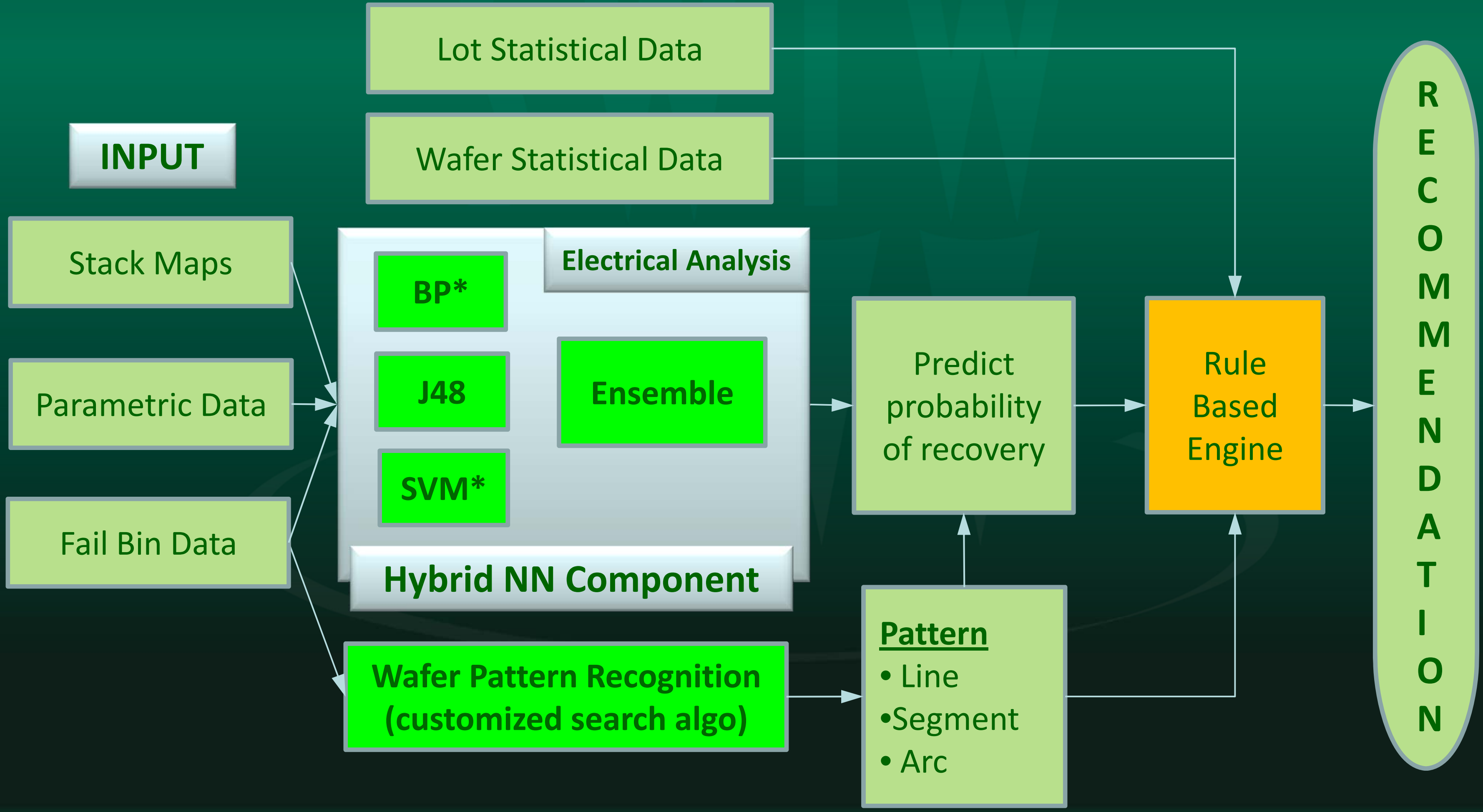
Data available ?

NO

Client or Web-based UI

Phase 3 : System structure (3)

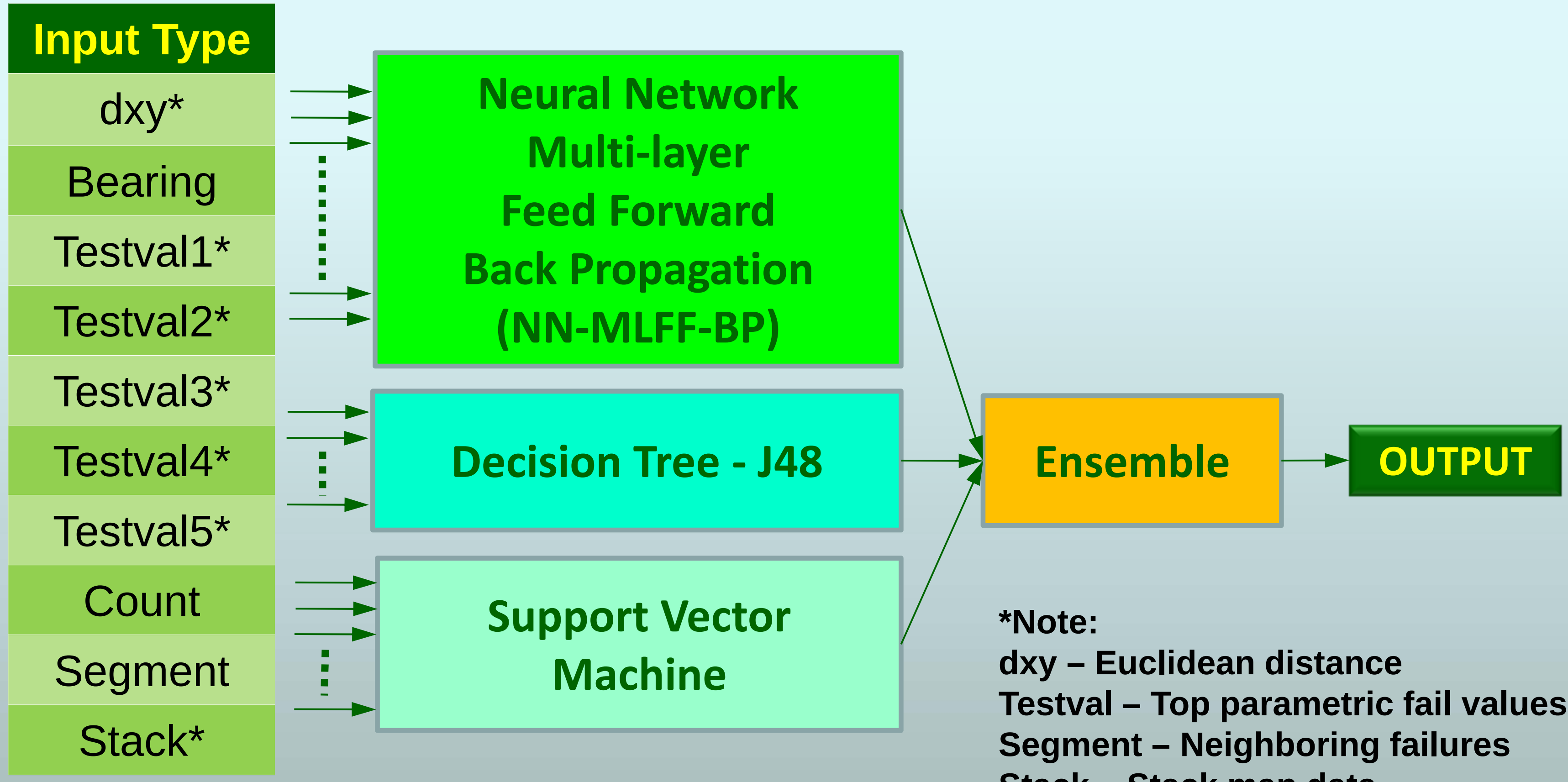
- **Overview – Hybrid Intelligence System**



Note:
BP - Backward Propagation
SVM – Support Vector Machine

Phase 3 : System structure (4)

- **Overview – AI System Modelling and Ensemble**



Input Type

- dxy*
- Bearing
- Testval1*
- Testval2*
- Testval3*
- Testval4*
- Testval5*
- Count
- Segment
- Stack*

Neural Network Multi-layer Feed Forward Back Propagation (NN-MLFF-BP)

Decision Tree - J48

Support Vector Machine

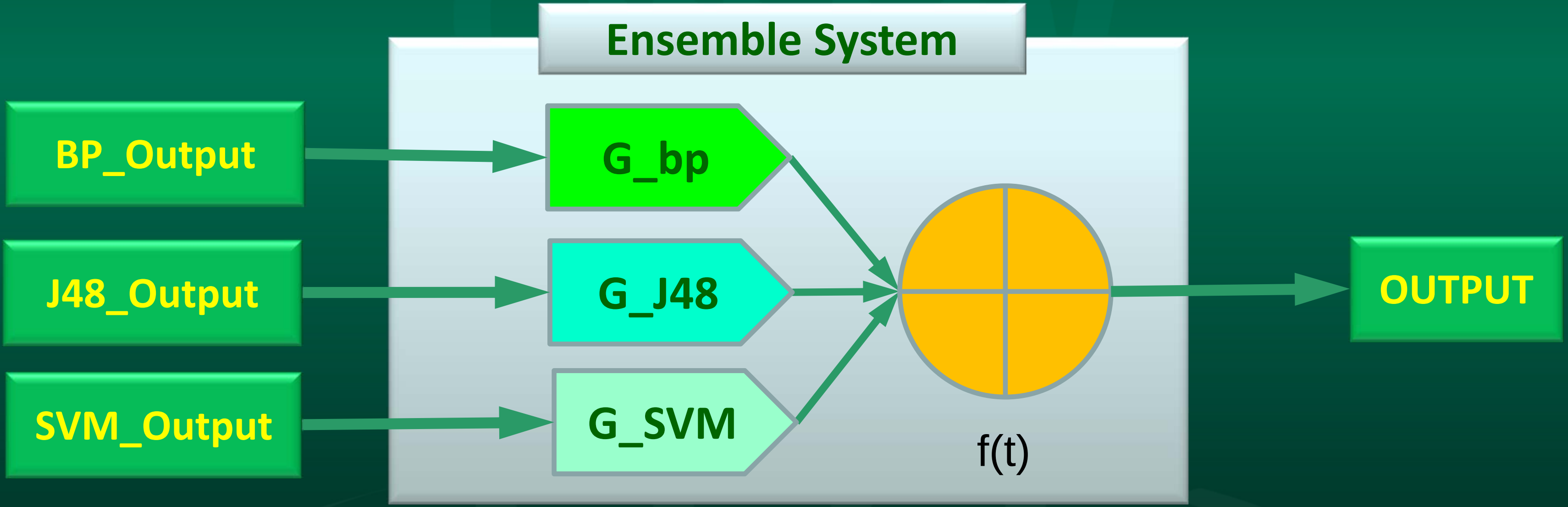
Ensemble

OUTPUT

***Note:**
dxy – Euclidean distance
Testval – Top parametric fail values
Segment – Neighboring failures
Stack – Stack map data

Phase 3 : System structure (5)

- **AI Modeling and Ensemble – How it integrate**



Trained Values		
BP	J48	SVM
75.00%	84.00%	81.00%

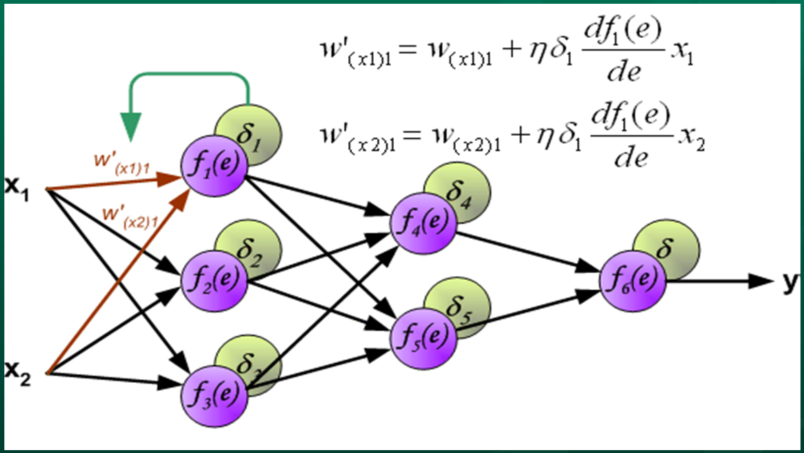
G_bp = 75/(75+84+81) = 0.3125
G_c45 = 84/(75+84+81) = 0.35
G_svm = 81/(75+84+81) = 0.3375
x = Bp_output * G_bp + J48_output * G_j48 + SVM_output * G_svm

f(t) x >= 0.6666,
0 otherwise

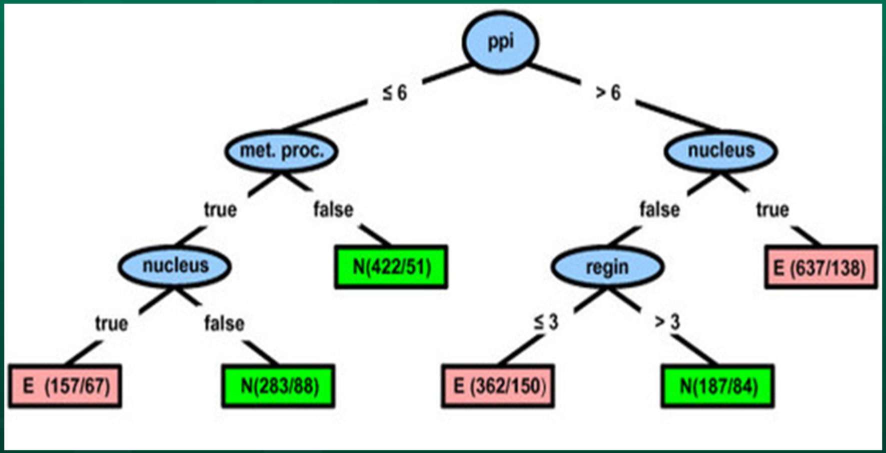
Phase 3 : System structure (6)

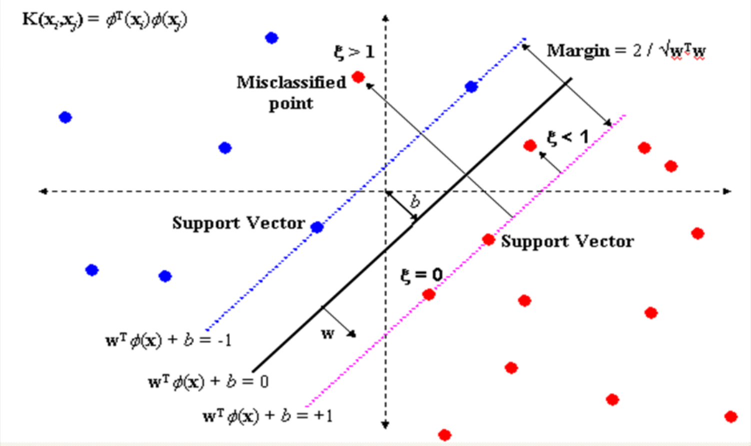
- **Details of the AI model employed**

NN-MLFF-BP



J48 Decision Tree

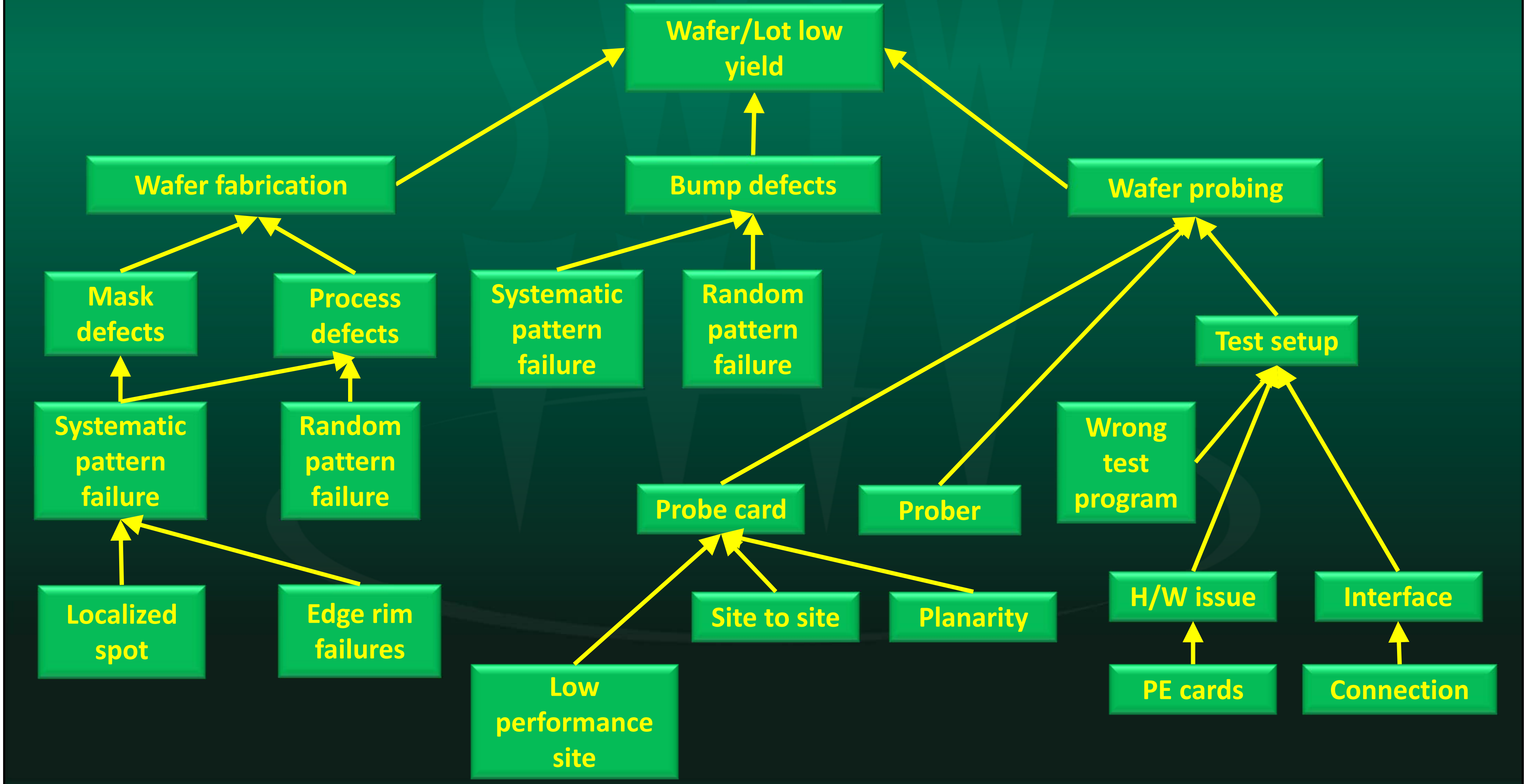




SVM

Phase 3 : System structure (7)

- Domain Knowledge



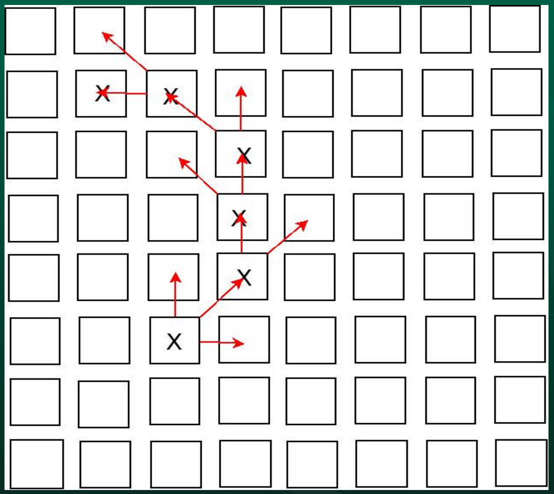
Phase 3 : System structure (8)

- Rule-based Engine Logic (translated Domain Knowledge)

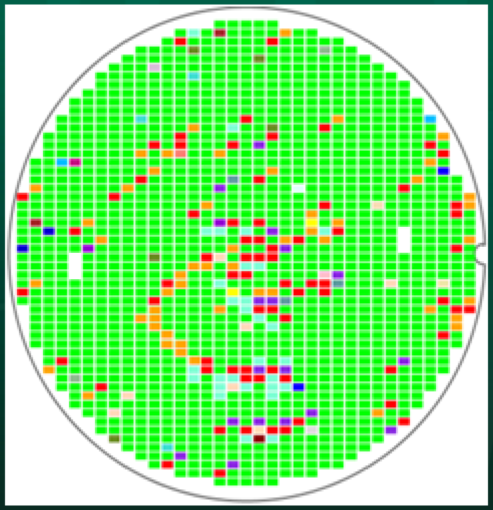
Yield		Yield		NN			Pattern			Message
Ave, 1st Pass	Ave, Auto Recovery	1st Pass Yield	Auto Reprobe Recovery	Ave. Auto Reprobe Recovery	Recovery Predict Count	NN Predict Yield Uncertainty	Line Count	Arc Count	Segment Count	
H		L	L							FP:L, AR:L
		M	L							FP:M,AR:L
		L	M							FP:L,AR:M
		M	M							FP:M,AR:M
		L	H							FP:L,AR:H
		M	H							FP:M,AR:H
					H	M				NN:H,UC:M, RT:F
					H	L				NN:H,UC:L, RT:F
							>0			Line
								>0		ARC
									>0	Segment

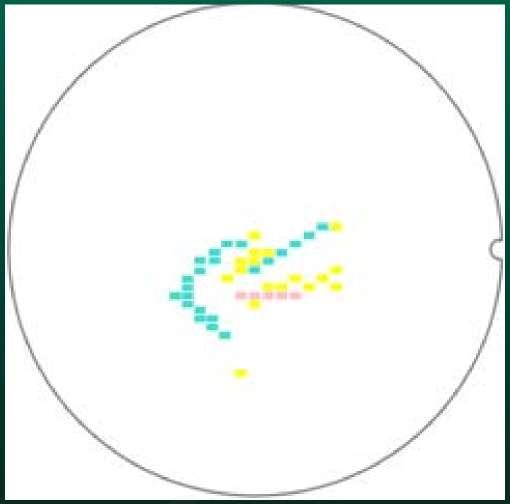
Phase 4 : Pattern Recognition Methodology (1)

- **Arc Detection**



Definition + Training

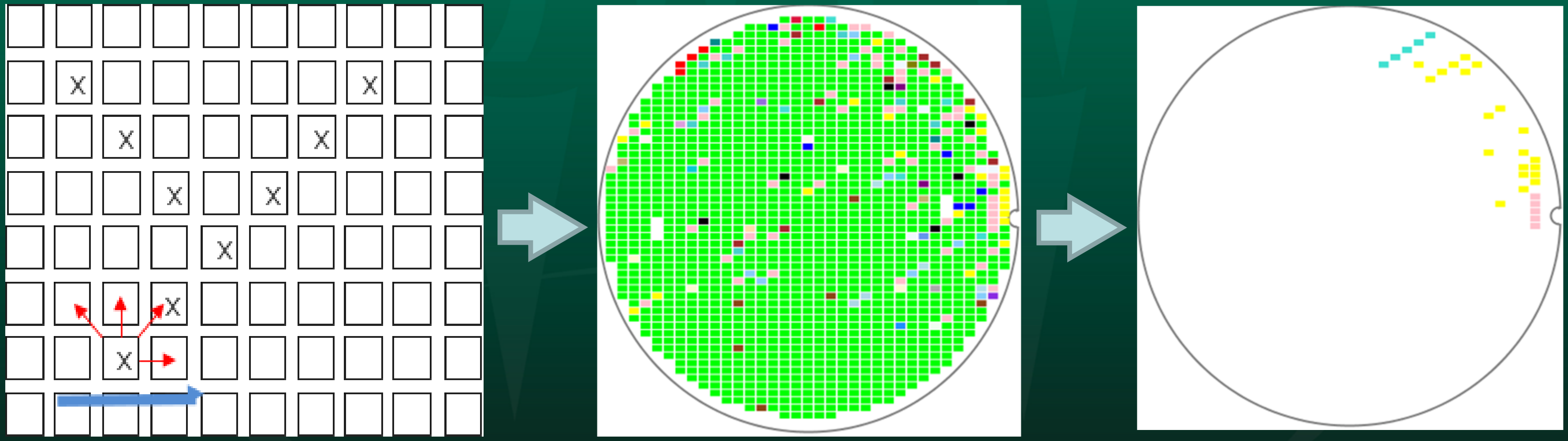




Filter results
+
Identification

Phase 4 : Pattern Recognition Methodology (2)

- **Line Detection**

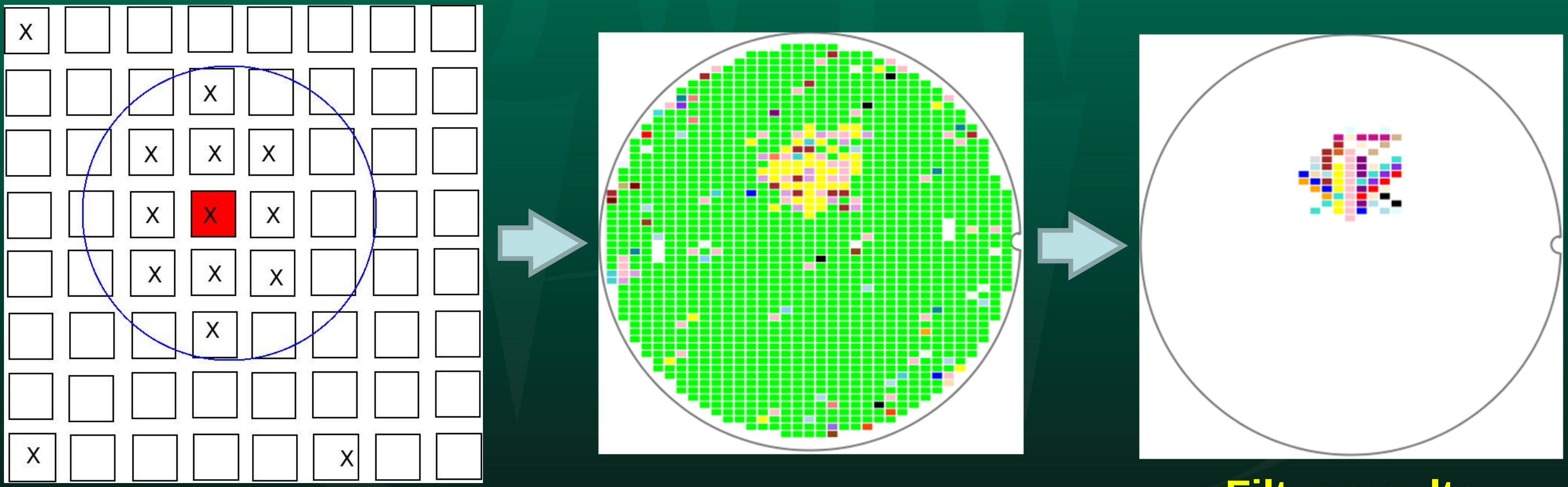


Definition + Training

Filter results
+
Identification

Phase 4 : Pattern Recognition Methodology (3)

- **Segment Detection**



Definition + Training

Filter results
+
Identification

Phase 5 : Testing and Results (1)

- Results from AI Model (Training Sets)

Items	NN-MLFF-BP	J48	SVM
Total Instances	38,122	38,122	38,122
Correctly Classified Instances	27,391 (71.8509 %)	30,932 (81.1395 %)	29,907 (78.4508 %)
Incorrectly Classified Instances	10,731 (28.1491 %)	7,190 (18.8605 %)	8,215 (21.5492 %)
Kappa statistic	0.4008	0.6032	0.539
Mean absolute error	0.3494	0.2584	0.2155
Root mean squared error	0.423	0.373	0.4642
Relative absolute error	72.6676 %	53.7533 %	44.8202 %
Root relative squared error	86.2723 %	76.0688 %	94.6788 %
Coverage of cases (0.95 level)	99.6695 %	98.0562 %	78.4508 %
Mean rel. region size (0.95 level)	94.233 %	83.5646 %	50 %

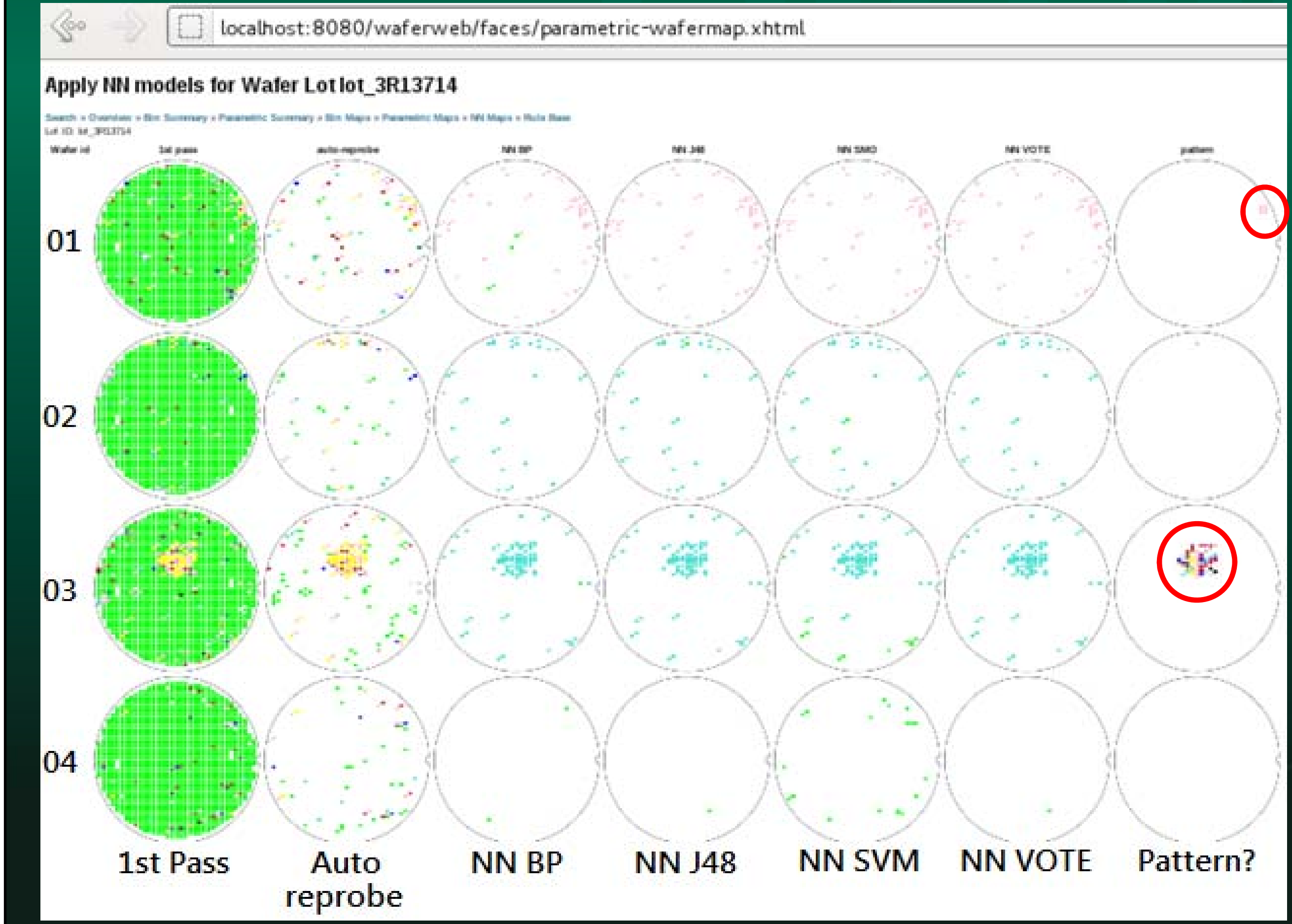
Phase 5 : Testing and Results (2)

- Results from AI Model (Test Sets)

Items	NN-MLFF-BP	J48	SVM
Total Instances	3997	3997	3997
Correctly Classified Instances	69.85245%	69.9775%	71.6537%
Incorrectly Classified Instances	30.1476%	30.0225%	28.3463%
Kappa statistic	0.118	0.3124	0.2893
Mean absolute error	0.4004	0.3319	0.2835
Root mean squared error	0.449	0.447	0.5324
Relative absolute error	NA	NA	NA
Root relative squared error	NA	NA	NA
Coverage of cases (0.95 level)	99.9249%	97.298%	71.6537%
Mean rel. region size (0.95 level)	NA	NA	NA

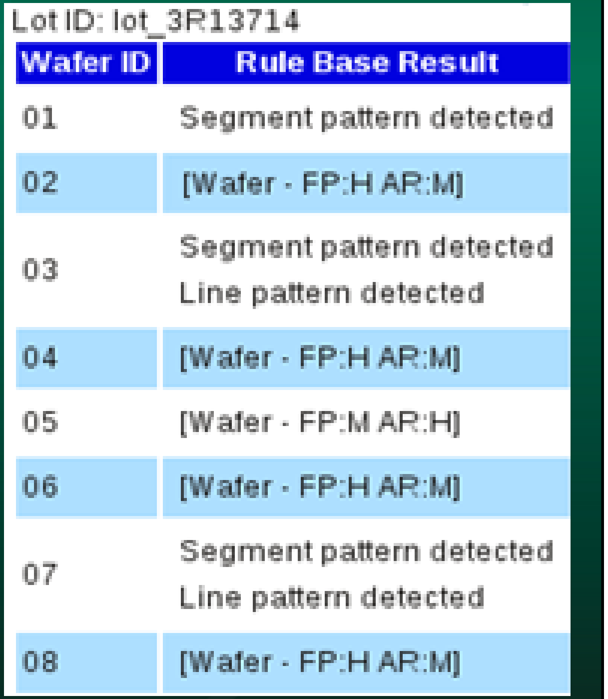
Phase 5 : Testing and Results (3)

- **Results from Rule-based from WEB UI**



localhost:8080/waferweb/faces/parametric-wafermap.xhtml

Apply NN models for Wafer Lot lot_3R13714



Lot ID: lot_3R13714

Wafer ID	Rule Base Result
01	Segment pattern detected
02	[Wafer - FP:H AR:M]
03	Segment pattern detected Line pattern detected
04	[Wafer - FP:H AR:M]
05	[Wafer - FP:M AR:H]
06	[Wafer - FP:H AR:M]
07	Segment pattern detected Line pattern detected
08	[Wafer - FP:H AR:M]

Other Considerations

- **Features**
  - Dynamic DB creation per lot. Ex.. lot_<lotnumber>
  - Childlots insert into parent lot DB.
  - Each STDF is tracked.
  - Traceable first pass, auto-reprobe, offline retest.
  - Each PTR records has its own table: ptr_<ptrnumber>
  - Lot DB be dropped to conserve space.
  - Redundancy for performance. XY coordinates included every table
- **Performance (Centrino 2 core 2.1Ghz, 4GRam laptop)**
  - Size: 30STDFs (121MB), DB size: ~ 4GB, 250 tables.
  - Insertion duration: Approx ~ 10-15mins per STDF.
  - Sql query : 10.22M records, 16850 results, 32 sec

List of tools used (1)

Tools	Info
MYSQL database	Widely available robust database with extensive technical help and online information available. Support in many platforms, particularly for Windows and various Linux distributions. Test data on database makes it easy to extract and access from most software development language and tools. For this project, a prototype database will be build to store test data, later to be part of the overall system. The database can then be switched to the sponsors' internal database system via an interface. link: http://www.mysql.com/
Weka Data-mining	Weka is a datamining software written in Java. It has a comprehensive set of analysis tools for data exploration and understanding. Weka java API can be access and called to perform data analysis functions and return results. We have successfully tested Weka's connection to MYSQL database. link: http://www.cs.waikato.ac.nz/ml/weka/
Oracle Netbeans	Netbeans IDE for Java application development and complementary to Glassfish. link: http://netbeans.org/

List of tools used (2)

Tools	Info
Eclipse	An OpenSource IDE for Java application development. Its flexible plugin framework allows integration of other tools and support multi languages. link: http://www.eclipse.org/
LibSTDF	C++ stdf-reader project derived from FreeSTDF project . Link: http://freestdf.sourceforge.net/
Rule-based Development tools and libraries	JBOSS DROOLS link - http://www.jboss.org/drools
Neura Network	Using Weka decision tree and NN functions.

List of tools used (3)

Tools	Info
Oracle GlassFish Application server	Oracle Glassfish application server community version is an Open Source application server that implements JAVAEE specification and is a potential good platform to build on. Glassfish provides a web container for web application, dependency injection to link services to business objects (KE algorithms), message queue for notification, asynchronous operations, timer service for cron jobs and ease of use among many other features. Glassfish allows us to build a proof of concept system that is flexible to change in later part of the project development. link: http://glassfish.java.net/

Conclusion

- It is concluded that the AI system deployed can help detect and flag-out underlying wafer process or test related problems. However, the accuracy of the prediction to decide on the retest is currently not able to replace human judgment. Nonetheless, it is a very useful tool to help minimize over-sights and definitely a good tool for knowledge retention. With further fine tuning to the system and by employing improved AI algorithm, it is hoped that the prediction accuracy can improve and plays a bigger complimentary role in the disposition process.

Future Work

- More training sets with new patterns to refine pattern detection
- Possible upstream and downstream integration : Etest data input, Final Test data input
- Explore other KE techniques, models.

Thank You

- We would like thank the following teams for the successful completion of the project :

ISS-NUS KE22 FYP Team :

Supervisors: Miss Fan Zhen Zhen, Dr Famming, Charles Pang
Team members: Tran The Anh, Tai JiaJun, Chew Poh Kwee
http://www.iss.nus.edu.sg/iss/browse.jsp?cid=7&pid=45&rid=45&type=9&name=Our_Staff

GLOBALFOUNDRIES Team :

Sponsor: Yeo Chak Huat
Team members: Lee Eu Jin, Zhong Jie