Development of PHM methods for NPP components using machine learning

Content

- PHM Concept
- Needs and Business Case
- Requirements
- Projects in the Application of PHM
  - Digital Twin (DT) Integration
  - Control Rod Drive System (CRDS) Monitoring
  - Solenoid Operated Valve (SOV) Monitoring
- Conclusion

PHM Concept

PHM Concept

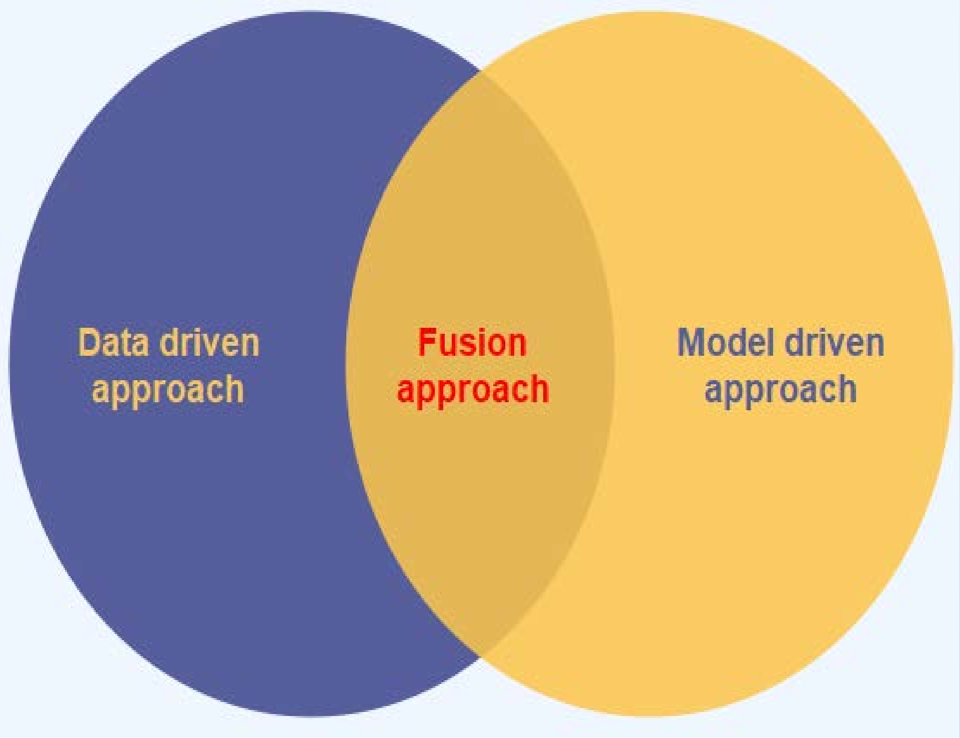
Prognostics and health management (PHM) is an engineering process of failure prevention, diagnosing and prediction of remaining useful life (RUL) of a system.

Benefits of PHM

- Enhanced system availability by extending the maintenance cycles through condition based maintenance
- Reduced failure rate by performing proactive maintenance to forestall failures
- Extension of operational life
- Reduction in inspection costs, required number of skilled labor, system down time and emergency unscheduled maintenance.

PHM Concept

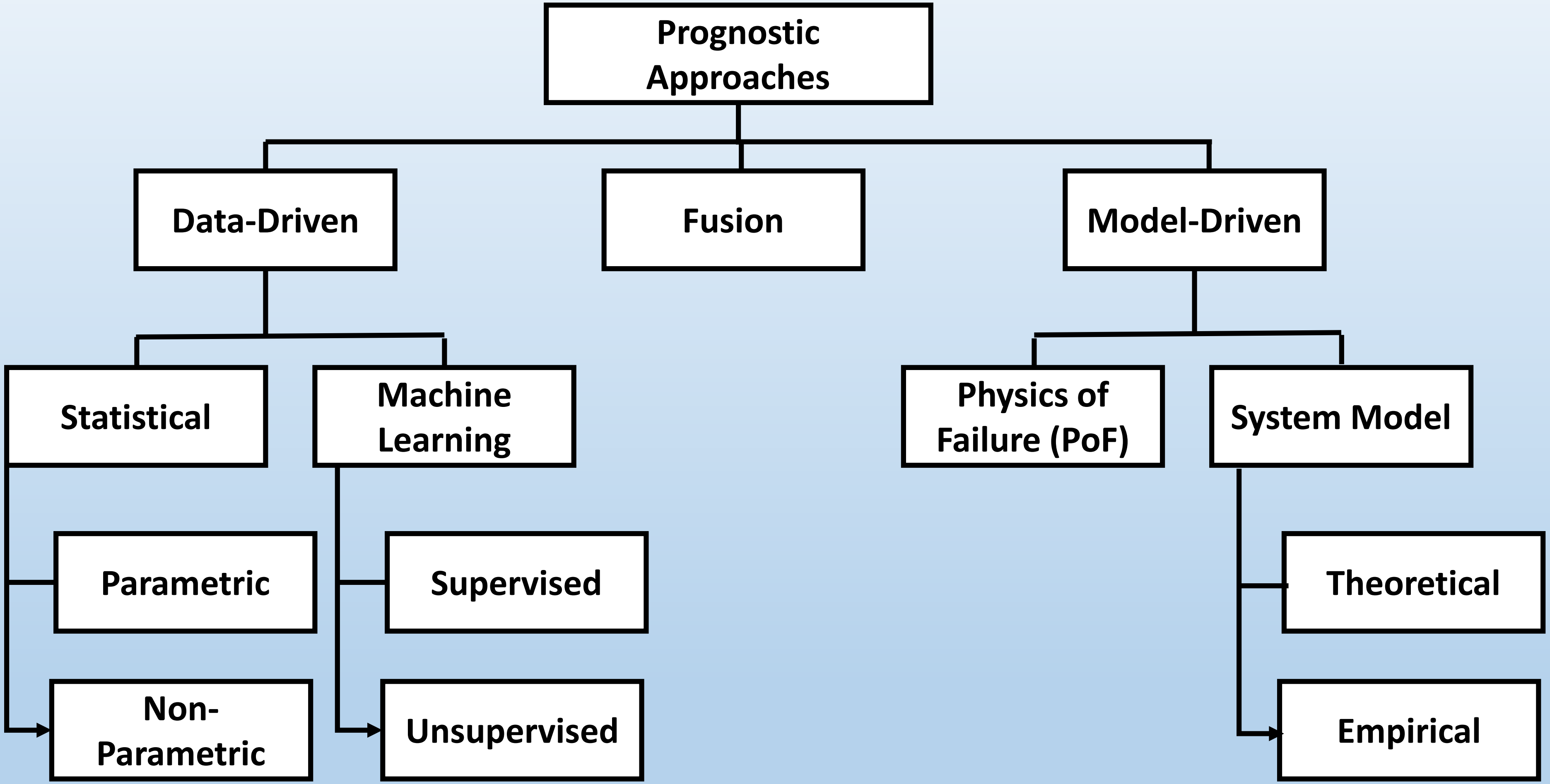
PHM Approaches



- Data driven approach do not need the system models or specific knowledge to perform PHM. They rely on historical and current data to perform prognostics. The major challenge is the availability of historical and empirical data
- Model driven approach uses mathematical equations that predict the physics governing failure , sometimes called Physics of Failure (PoF). They require expert knowledge and this can be a challenge when the system becomes more complex.
- Fusion approach is based on advanced features from both data driven and model driven approaches. This requires accurate mathematical models and enough historical data. This approach is meant to be more accurate than the other models

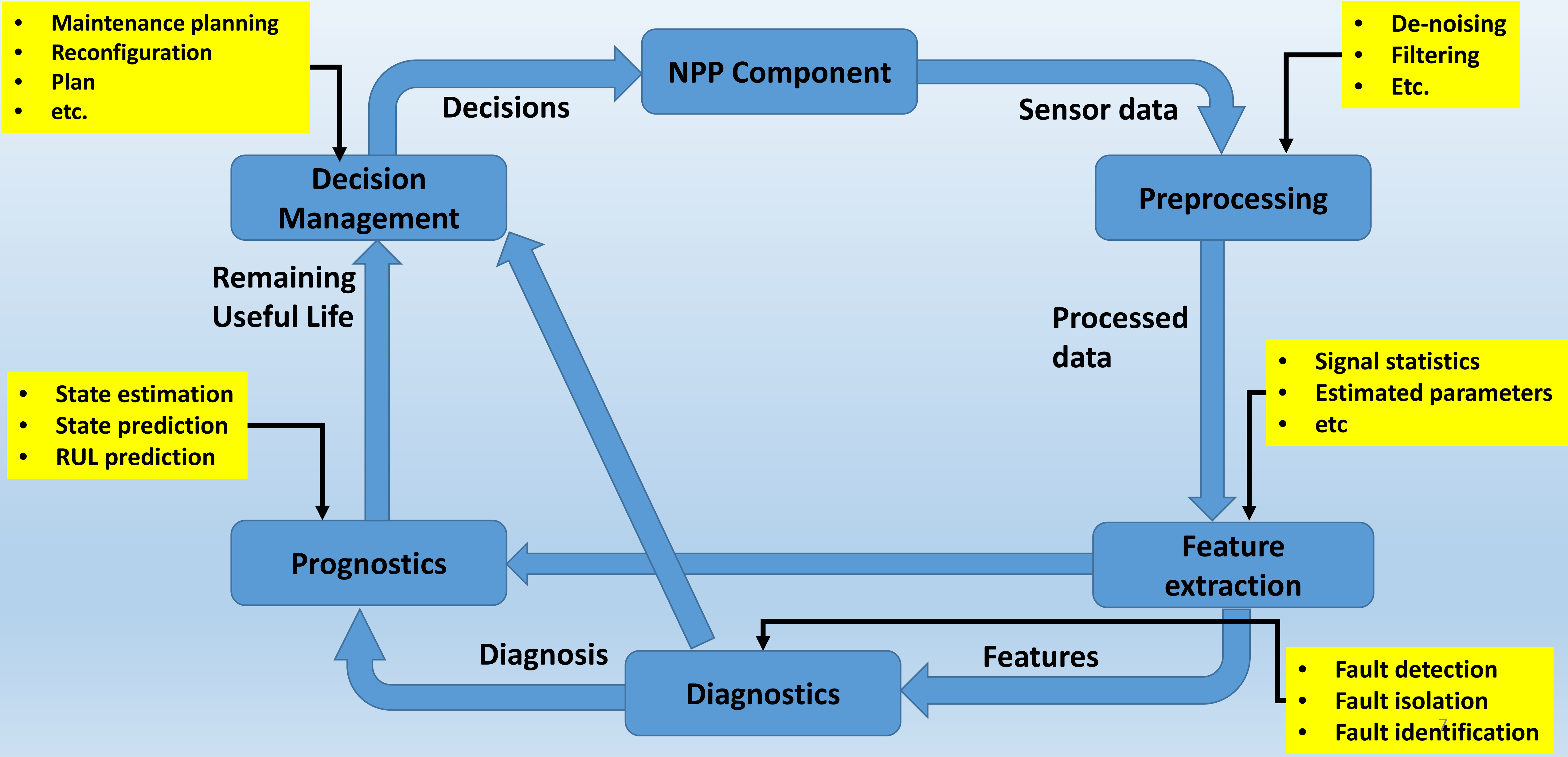
PHM Concept

PHM Approaches



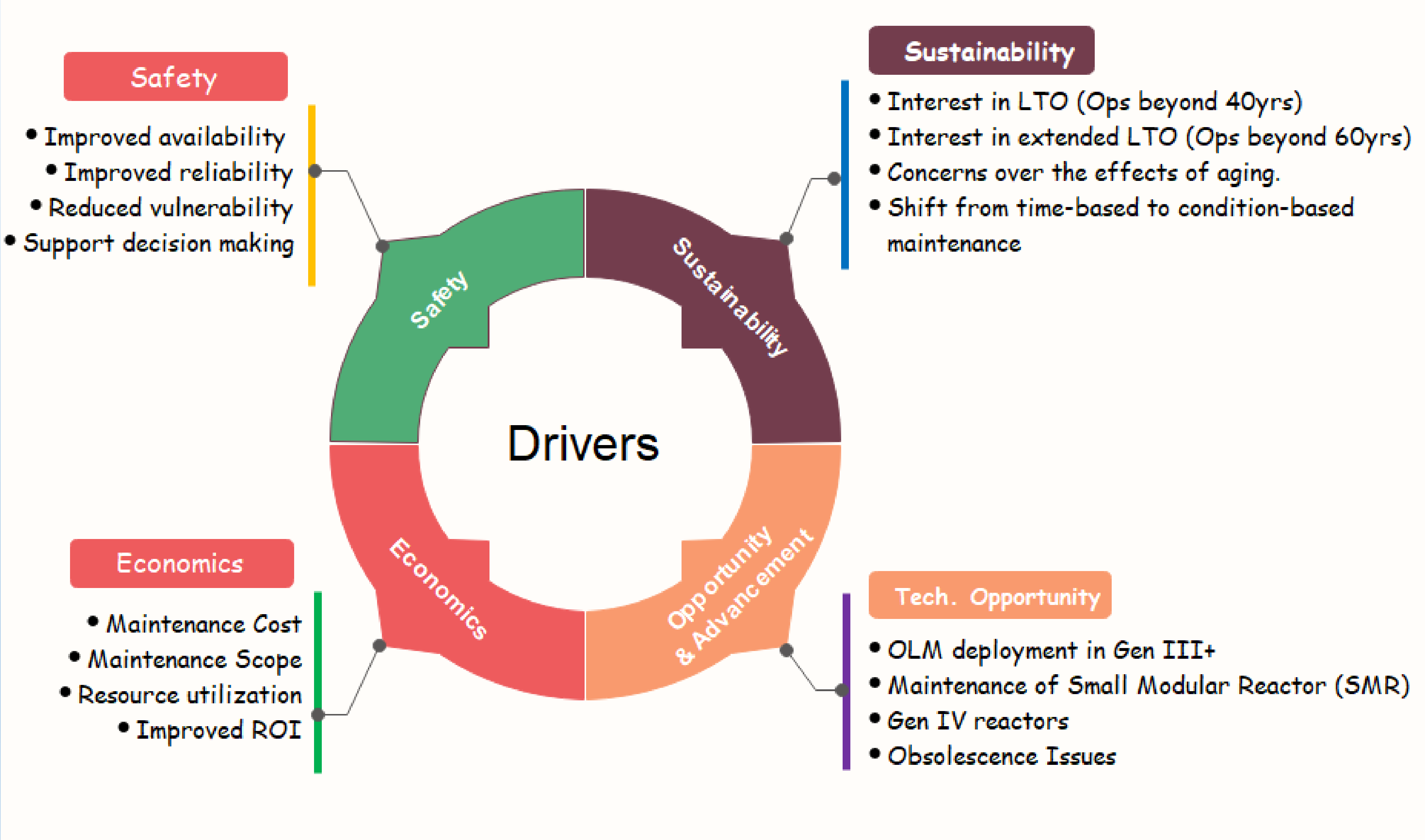
PHM Concept

PHM Overview



Needs and Business Case

Business Case and Needs Analysis



Business Case and Needs Analysis

Other considerations

Maintainers
- Scheduling Mx
- Opportunistic Mx
- System Uptime
- Minimize unnecessary Mx

Logisticians
- Spares Positioning
- Reduced Spares Count
- Logistics Footprint

Engineers
- Requirements Satisfaction
- Robustness
- Design for PHM

Regulatory Bodies
- Safety
- Avoid Catastrophic Failures
- Minimize impact on other (healthy) systems

Fleet Management
- Fleet Health
- Lifecycle Cost
- Mission Capability
- Mission Planning
- Minimize downtime

Program Mgmt
- Meeting customer expectations

Not Just for Maintenance!

MOE & Value Elements

Category	End User	Goals	Metric
Operations	Program manager	Economic Viability of prognosis technology	Cost-benefit metric
	Plant Manager	Resource planning and utilization	Accuracy and precision metrics of prediction or RUL
	Operator	Informed operator action/response	Accuracy and precision based metric for RUL estimation on specific SSCs
	Maintainer	Plan maintenance in advance to reduce downtime and maximize uptime	Accuracy and Precision metrics of RUL
Engineering	Designer	Implement PHM system within requirements and constraints	Reliability based metrics for design evaluation. Computational performance metric
	Researchers	Assess confidence level, uncertainty and error	Accuracy and Precision metrics with uncertainty
Regulatory	Policy Maker	Assess potential risk	Cost-benefit-risk measures, accuracy and precision metrics to establish guidelines

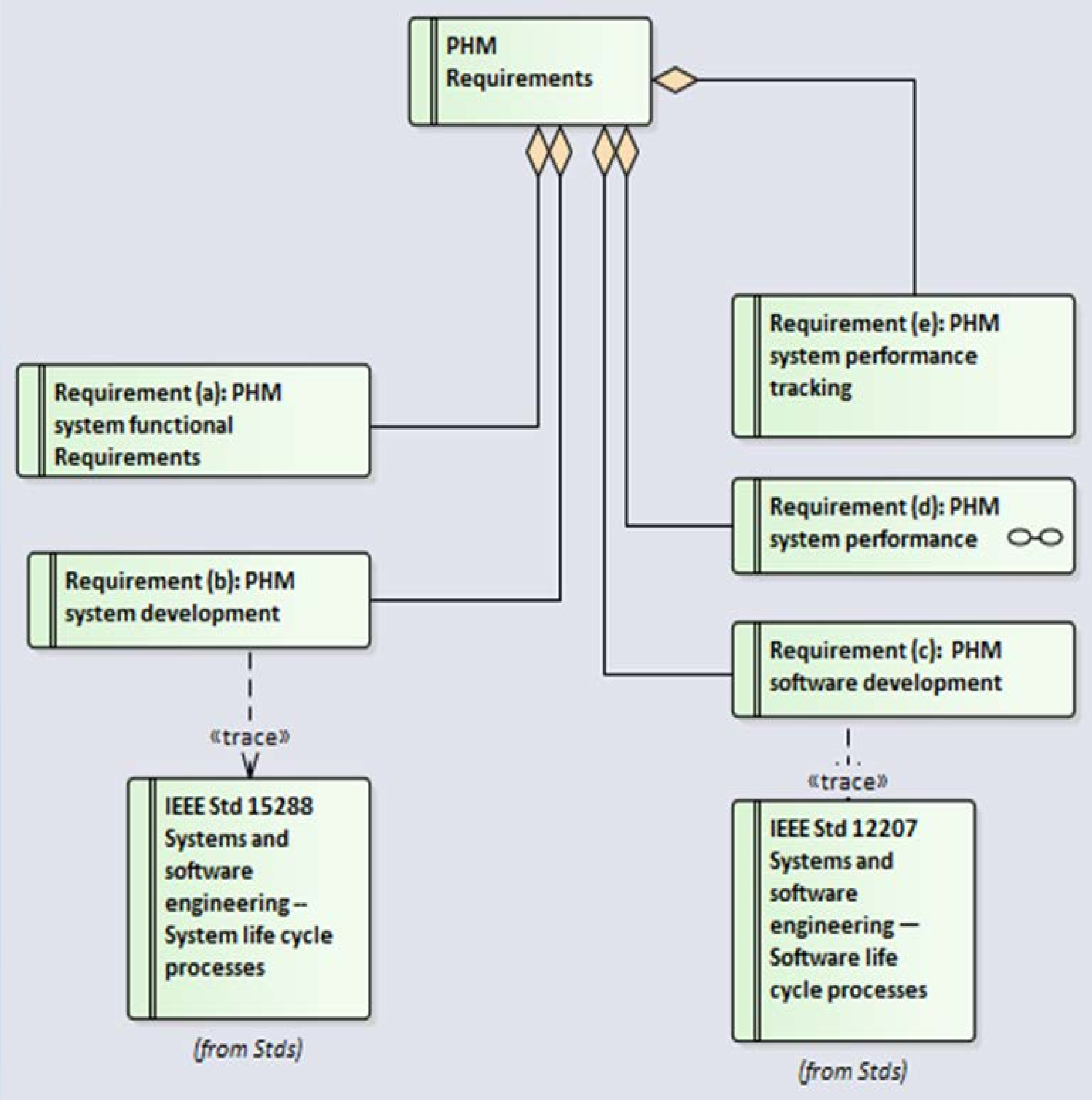
Requirements

Requirements

Functional

- Data Acquisition
- Data management
- Data processing
- Diagnostics, Health state estimation, & prognostics
- Health management

IEEE Std. 1856:
Framework for PHM of Electronic Systems



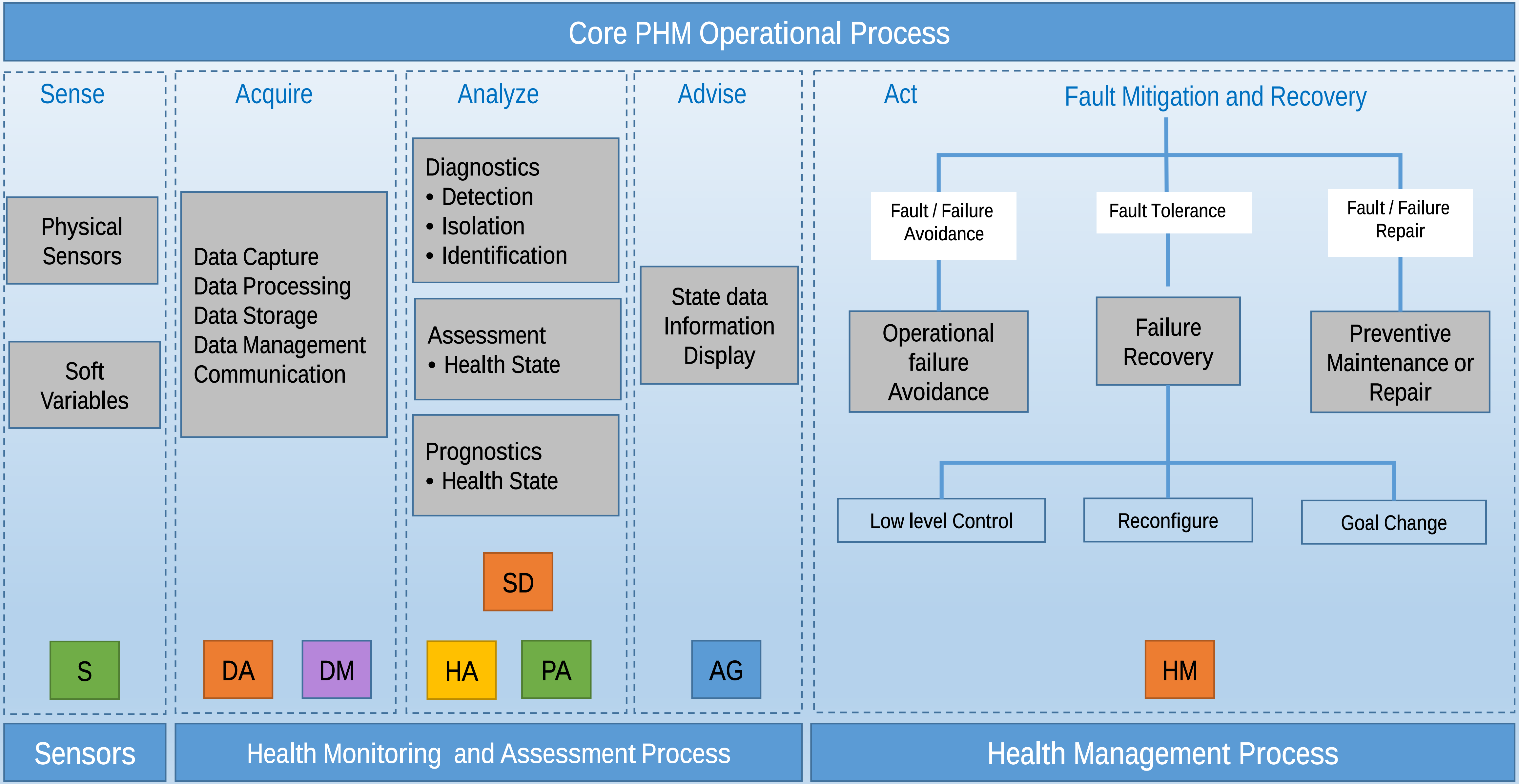
Performance

- Accuracy
- Timeliness
- Confidence
- Effectiveness

PHM Functional Model

Core PHM Operational Processes	Functional Block	Description
Act	Health Management (HM)	This function uses the information generated in the AG to institute actions to return the systems to a “health state”
Advise	Advisory Generation (AG)	This function provides actionable information to operational and maintenance personnel or external systems
Analyze	Prognostic Assessment (PA)	This function provides future state of health, performance life remaining, or remaining useful life indicators
Analyze	Health Assessment (HA)	This function provides information to determine the current state of health of the systems
Analyze	State Detection (SD)	This function evaluates equipment state conditions against normal operating profiles and generates normal or abnormal condition indicators.
Acquire	Data Manipulation (DM)	This function processes and transforms the sensor data and health state information collected by the DA
Acquire	Data Acquisition (DA)	This function acquires and records the sensor data and health state information from the systems' internal monitors, data recorder or system data bus
Sense	Sensors (S)	This involves physical sensors and any ‘soft’ system performance variable available within the system.

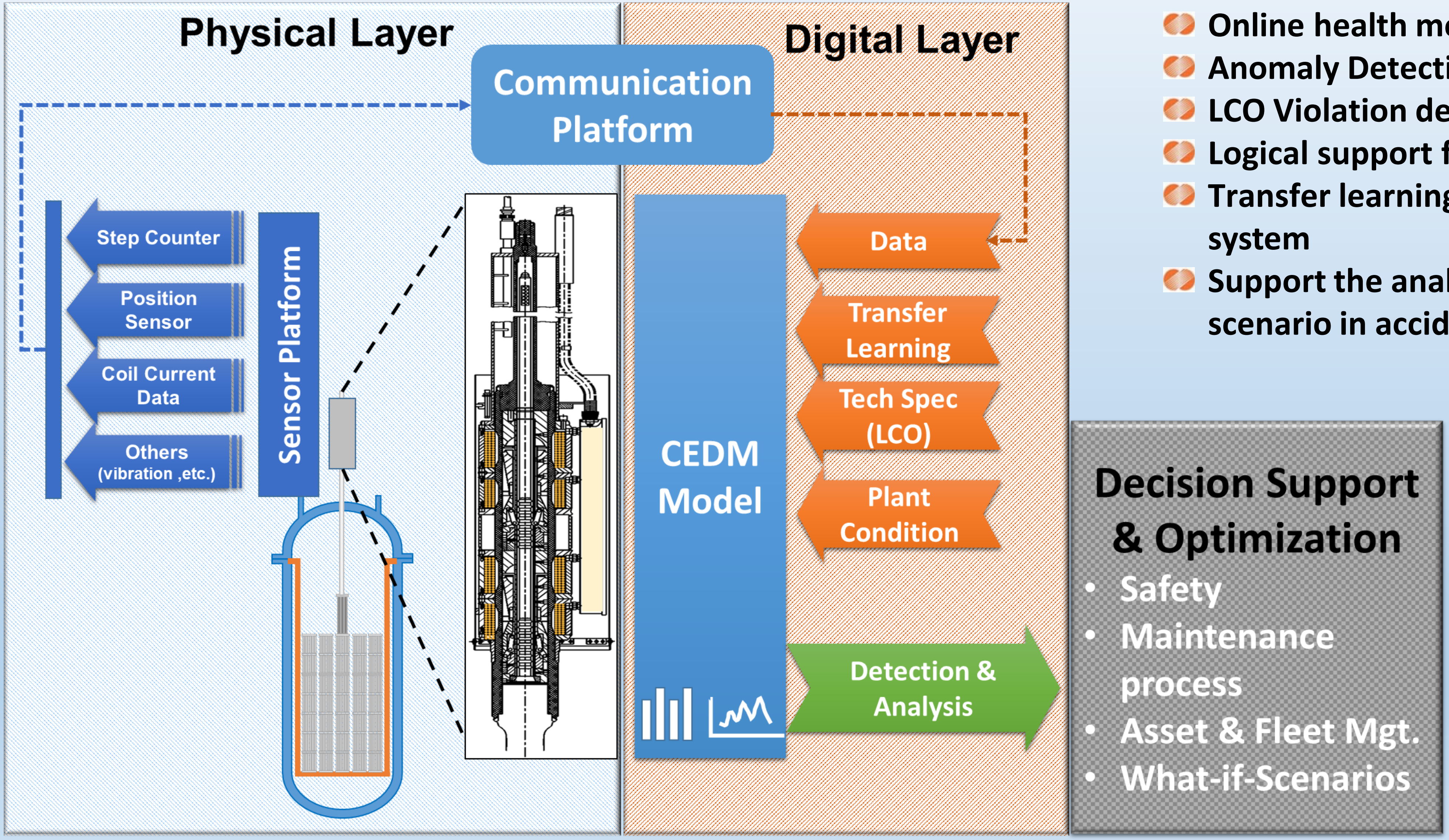
PHM Operational Process



Projects

- Conceptual framework for digital twin (DT) in PHM application
- Anomaly detection in Control Rod Drive System
- PHM of Solenoid Operated Valve (SOV)

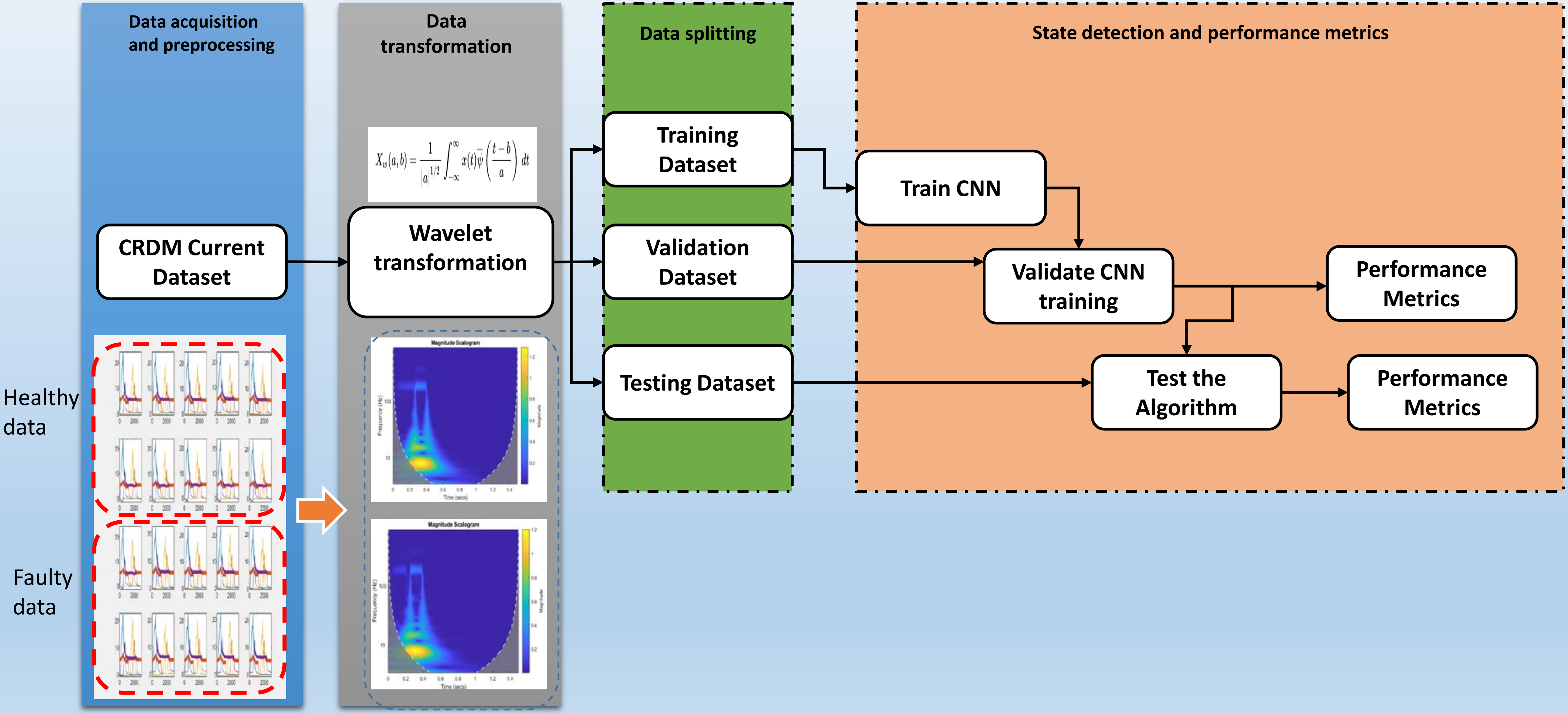
Conceptual framework for DT in PHM application



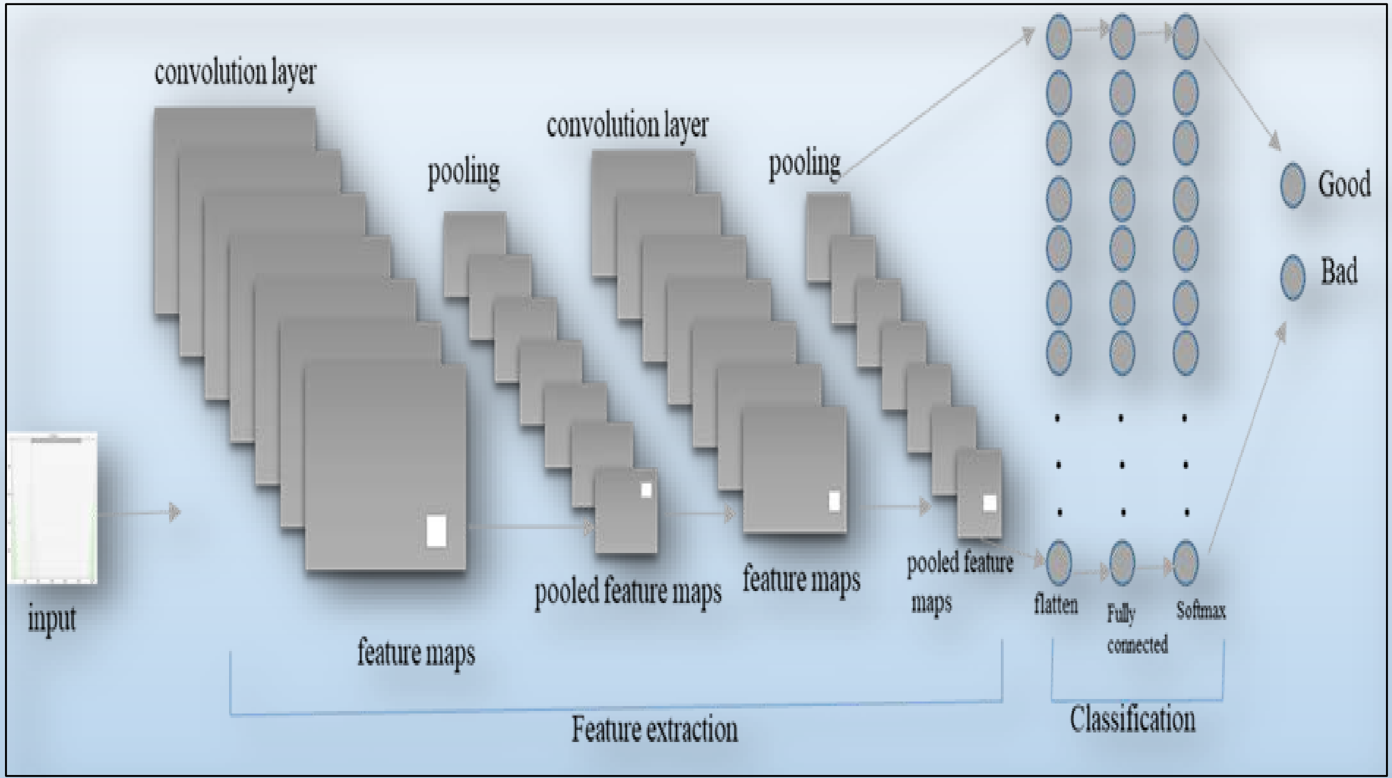
- Online health monitoring
- Anomaly Detection
- LCO Violation detection and Alert
- Logical support for LCO restoration
- Transfer learning from similar system
- Support the analysis of postulated scenario in accident situation

CEDM fault detection with CNN

Deep Learning approach



CNN Architecture



CNN approach applied for this research was transfer learning.

- **Googlenet**

GoogLeNet is a pretrained convolutional network that is 22 layers deep. This number does not include the pooling layers and the independent blocks used for constructing the network. This network has been used to classify up to 1000 images. This network uses a boosting approach for prediction

- CNN is a deep neural networks.
- Commonly used for image classification and video recognition

Layers:

- ***Feature learning layer***
- ***Classification layer***

Feature Extraction Layer consist of the convolution layer, with an activation function and a pooling layer.
The classification layer consist of the Flatten, Fully connected and the Activation function layer.

CNN Architecture

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

GoogLeNet structure

The **GoogLeNet** was modified for the purpose of classifying the CRDM states. For the CRDM, there were two output classes and the GoogLeNet was developed for 1000 classes. The output classes were modified and the dropouts as well.

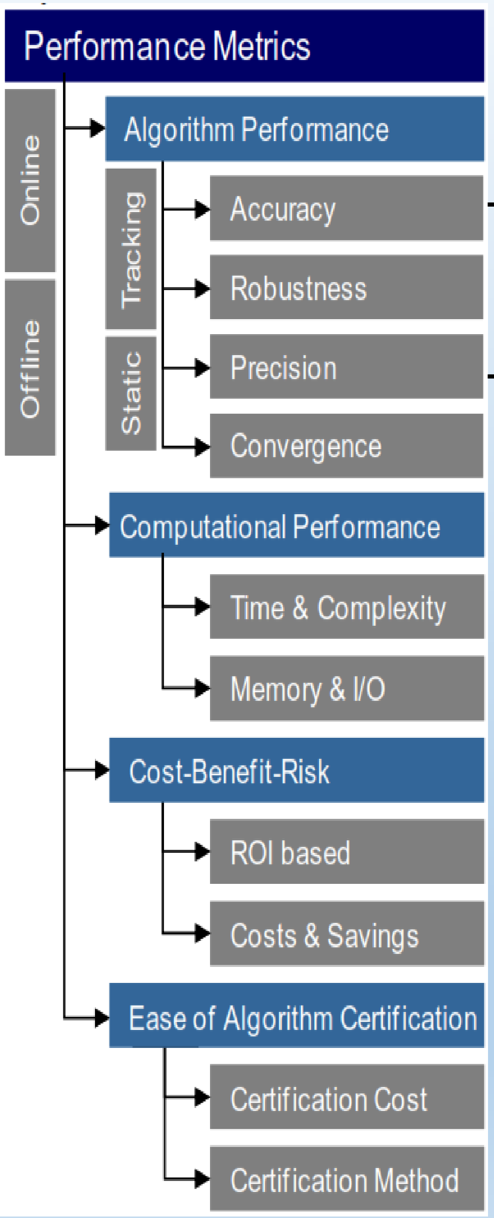
The total number of layers for the implemented architecture was **144** and is shown in the next slide.

CNN Architecture

1	'data'	Image Input	224x224x3 images with 'zerocenter' normalization
2	'conv1-7x7_s2'	Convolution	64 7x7x3 convolutions with stride [2 2] and padding [3 3 3 3]
3	'conv1-relu_7x7'	ReLU	ReLU
4	'pool1-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
5	'pool1-norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
6	'conv2-3x3_reduce'	Convolution	64 1x1x64 convolutions with stride [1 1] and padding [0 0 0 0]
7	'conv2-relu_3x3_reduce'	ReLU	ReLU
8	'conv2-3x3'	Convolution	192 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
9	'conv2-relu_3x3'	ReLU	ReLU
10	'conv2-norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
11	'pool2-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
12	'inception_3a-1x1'	Convolution	64 1x1x192 convolutions with stride [1 1] and padding [0 0 0 0]
13	'inception_3a-relu_1x1'	ReLU	ReLU
14	'inception_3a-3x3_reduce'	Convolution	96 1x1x192 convolutions with stride [1 1] and padding [0 0 0 0]
15	'inception_3a-relu_3x3_reduce'	ReLU	ReLU
16	'inception_3a-3x3'	Convolution	128 3x3x96 convolutions with stride [1 1] and padding [1 1 1 1]
17	'inception_3a-relu_3x3'	ReLU	ReLU
18	'inception_3a-5x5_reduce'	Convolution	16 1x1x192 convolutions with stride [1 1] and padding [0 0 0 0]
19	'inception_3a-relu_5x5_reduce'	ReLU	ReLU
20	'inception_3a-5x5'	Convolution	32 5x5x16 convolutions with stride [1 1] and padding [2 2 2 2]
21	'inception_3a-relu_5x5'	ReLU	ReLU
22	'inception_3a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
23	'inception_3a-pool_proj'	Convolution	32 1x1x192 convolutions with stride [1 1] and padding [0 0 0 0]
24	'inception_3a-relu_pool_proj'	ReLU	ReLU
25	'inception_3a-output'	Depth concatenation	Depth concatenation of 4 inputs
26	'inception_3b-1x1'	Convolution	128 1x1x256 convolutions with stride [1 1] and padding [0 0 0 0]
27	'inception_3b-relu_1x1'	ReLU	ReLU
28	'inception_3b-3x3_reduce'	Convolution	128 1x1x256 convolutions with stride [1 1] and padding [0 0 0 0]
29	'inception_3b-relu_3x3_reduce'	ReLU	ReLU
30	'inception_3b-3x3'	Convolution	192 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
31	'inception_3b-relu_3x3'	ReLU	ReLU
32	'inception_3b-5x5_reduce'	Convolution	32 1x1x256 convolutions with stride [1 1] and padding [0 0 0 0]
33	'inception_3b-relu_5x5_reduce'	ReLU	ReLU
34	'inception_3b-5x5'	Convolution	96 5x5x32 convolutions with stride [1 1] and padding [2 2 2 2]
35	'inception_3b-relu_5x5'	ReLU	ReLU
36	'inception_3b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
37	'inception_3b-pool_proj'	Convolution	64 1x1x256 convolutions with stride [1 1] and padding [0 0 0 0]
38	'inception_3b-relu_pool_proj'	ReLU	ReLU
39	'inception_3b-output'	Depth concatenation	Depth concatenation of 4 inputs
40	'pool3-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
41	'inception_4a-1x1'	Convolution	192 1x1x480 convolutions with stride [1 1] and padding [0 0 0 0]
42	'inception_4a-relu_1x1'	ReLU	ReLU
43	'inception_4a-3x3_reduce'	Convolution	96 1x1x480 convolutions with stride [1 1] and padding [0 0 0 0]
44	'inception_4a-relu_3x3_reduce'	ReLU	ReLU
45	'inception_4a-3x3'	Convolution	208 3x3x96 convolutions with stride [1 1] and padding [1 1 1 1]
46	'inception_4a-relu_3x3'	ReLU	ReLU
47	'inception_4a-5x5_reduce'	Convolution	16 1x1x480 convolutions with stride [1 1] and padding [0 0 0 0]
48	'inception_4a-relu_5x5_reduce'	ReLU	ReLU
49	'inception_4a-5x5'	Convolution	48 5x5x16 convolutions with stride [1 1] and padding [2 2 2 2]
50	'inception_4a-relu_5x5'	ReLU	ReLU
51	'inception_4a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
52	'inception_4a-pool_proj'	Convolution	64 1x1x480 convolutions with stride [1 1] and padding [0 0 0 0]
53	'inception_4a-relu_pool_proj'	ReLU	ReLU
54	'inception_4a-output'	Depth concatenation	Depth concatenation of 4 inputs
55	'inception_4b-1x1'	Convolution	160 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
56	'inception_4b-relu_1x1'	ReLU	ReLU
57	'inception_4b-3x3_reduce'	Convolution	112 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
58	'inception_4b-relu_3x3_reduce'	ReLU	ReLU
59	'inception_4b-3x3'	Convolution	224 3x3x112 convolutions with stride [1 1] and padding [1 1 1 1]
60	'inception_4b-relu_3x3'	ReLU	ReLU
61	'inception_4b-5x5_reduce'	Convolution	24 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
62	'inception_4b-relu_5x5_reduce'	ReLU	ReLU
63	'inception_4b-5x5'	Convolution	64 5x5x24 convolutions with stride [1 1] and padding [2 2 2 2]
64	'inception_4b-relu_5x5'	ReLU	ReLU
65	'inception_4b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
66	'inception_4b-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
67	'inception_4b-relu_pool_proj'	ReLU	ReLU
68	'inception_4b-output'	Depth concatenation	Depth concatenation of 4 inputs
69	'inception_4c-1x1'	Convolution	128 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
70	'inception_4c-relu_1x1'	ReLU	ReLU
71	'inception_4c-3x3_reduce'	Convolution	128 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
72	'inception_4c-relu_3x3_reduce'	ReLU	ReLU
73	'inception_4c-3x3'	Convolution	256 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
74	'inception_4c-relu_3x3'	ReLU	ReLU
75	'inception_4c-5x5_reduce'	Convolution	24 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
76	'inception_4c-relu_5x5_reduce'	ReLU	ReLU
77	'inception_4c-5x5'	Convolution	64 5x5x24 convolutions with stride [1 1] and padding [2 2 2 2]
78	'inception_4c-relu_5x5'	ReLU	ReLU
79	'inception_4c-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
80	'inception_4c-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
81	'inception_4c-relu_pool_proj'	ReLU	ReLU
82	'inception_4c-output'	Depth concatenation	Depth concatenation of 4 inputs
83	'inception_4d-1x1'	Convolution	112 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
84	'inception_4d-relu_1x1'	ReLU	ReLU
85	'inception_4d-3x3_reduce'	Convolution	144 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
86	'inception_4d-relu_3x3_reduce'	ReLU	ReLU
87	'inception_4d-3x3'	Convolution	288 3x3x144 convolutions with stride [1 1] and padding [1 1 1 1]
88	'inception_4d-relu_3x3'	ReLU	ReLU
89	'inception_4d-5x5_reduce'	Convolution	32 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
90	'inception_4d-relu_5x5_reduce'	ReLU	ReLU
91	'inception_4d-5x5'	Convolution	64 5x5x32 convolutions with stride [1 1] and padding [2 2 2 2]
92	'inception_4d-relu_5x5'	ReLU	ReLU
93	'inception_4d-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
94	'inception_4d-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
95	'inception_4d-relu_pool_proj'	ReLU	ReLU
96	'inception_4d-output'	Depth concatenation	Depth concatenation of 4 inputs
97	'inception_4e-1x1'	Convolution	256 1x1x528 convolutions with stride [1 1] and padding [0 0 0 0]
98	'inception_4e-relu_1x1'	ReLU	ReLU
99	'inception_4e-3x3_reduce'	Convolution	160 1x1x528 convolutions with stride [1 1] and padding [0 0 0 0]
100	'inception_4e-relu_3x3_reduce'	ReLU	ReLU
101	'inception_4e-3x3'	Convolution	320 3x3x160 convolutions with stride [1 1] and padding [1 1 1 1]
102	'inception_4e-relu_3x3'	ReLU	ReLU
103	'inception_4e-5x5_reduce'	Convolution	32 1x1x528 convolutions with stride [1 1] and padding [0 0 0 0]
104	'inception_4e-relu_5x5_reduce'	ReLU	ReLU
105	'inception_4e-5x5'	Convolution	128 5x5x32 convolutions with stride [1 1] and padding [2 2 2 2]
106	'inception_4e-relu_5x5'	ReLU	ReLU
107	'inception_4e-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
108	'inception_4e-pool_proj'	Convolution	128 1x1x528 convolutions with stride [1 1] and padding [0 0 0 0]
109	'inception_4e-relu_pool_proj'	ReLU	ReLU
110	'inception_4e-output'	Depth concatenation	Depth concatenation of 4 inputs
111	'pool4-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
112	'inception_5a-1x1'	Convolution	256 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
113	'inception_5a-relu_1x1'	ReLU	ReLU
114	'inception_5a-3x3_reduce'	Convolution	160 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
115	'inception_5a-relu_3x3_reduce'	ReLU	ReLU
116	'inception_5a-3x3'	Convolution	320 3x3x160 convolutions with stride [1 1] and padding [1 1 1 1]
117	'inception_5a-relu_3x3'	ReLU	ReLU
118	'inception_5a-5x5_reduce'	Convolution	32 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
119	'inception_5a-relu_5x5_reduce'	ReLU	ReLU
120	'inception_5a-5x5'	Convolution	128 5x5x32 convolutions with stride [1 1] and padding [2 2 2 2]
121	'inception_5a-relu_5x5'	ReLU	ReLU
122	'inception_5a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
123	'inception_5a-pool_proj'	Convolution	128 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
124	'inception_5a-relu_pool_proj'	ReLU	ReLU
125	'inception_5a-output'	Depth concatenation	Depth concatenation of 4 inputs
126	'inception_5b-1x1'	Convolution	384 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
127	'inception_5b-relu_1x1'	ReLU	ReLU
128	'inception_5b-3x3_reduce'	Convolution	192 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
129	'inception_5b-relu_3x3_reduce'	ReLU	ReLU
130	'inception_5b-3x3'	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
131	'inception_5b-relu_3x3'	ReLU	ReLU
132	'inception_5b-5x5_reduce'	Convolution	48 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
133	'inception_5b-relu_5x5_reduce'	ReLU	ReLU
134	'inception_5b-5x5'	Convolution	128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
135	'inception_5b-relu_5x5'	ReLU	ReLU
136	'inception_5b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1 1 1]
137	'inception_5b-pool_proj'	Convolution	128 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
138	'inception_5b-relu_pool_proj'	ReLU	ReLU
139	'inception_5b-output'	Depth concatenation	Depth concatenation of 4 inputs
140	'pool5-7x7_s1'	Average Pooling	7x7 average pooling with stride [1 1] and padding [0 0 0 0]
141	'newDropout'	Dropout	60% dropout
142	'fc'	Fully Connected	8 fully connected layer
143	'softmax'	Softmax	softmax
144	'classoutput'	Classification Output	crossentropyex

CRDM GoogLeNet layers breakdown

Performance Metrics



$$\frac{True\ Positive + True\ Negative}{TP + TN + FP + FN}$$

$$\frac{TP}{TP + FP}$$

ROI = (Return – Investment) / Investment

$Savings_{PHM} = Cost_{without\ PHM} - Cost_{with\ PHM}$

true negative: A correct decision that a condition does not exist.

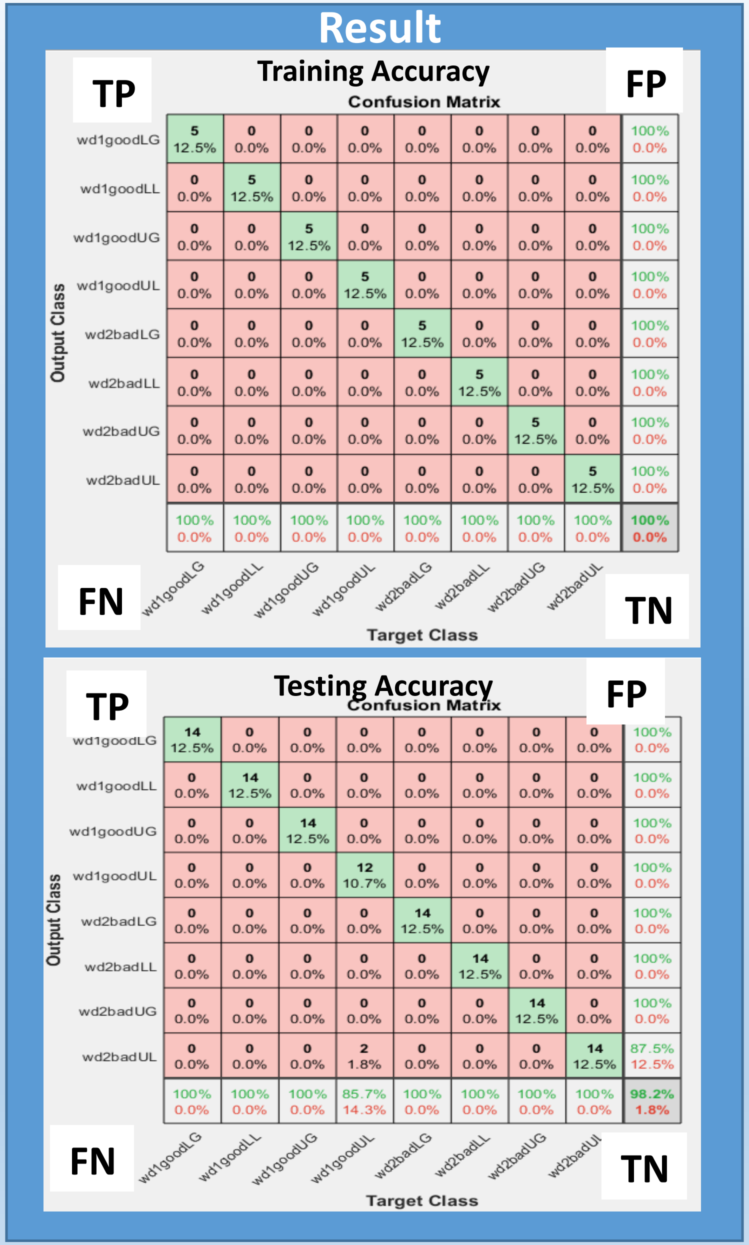
true positive: A correct decision that a condition does exist.

True Positive (TP)	False Negative (FN)
False Positive (FP)	True Negative (TN)

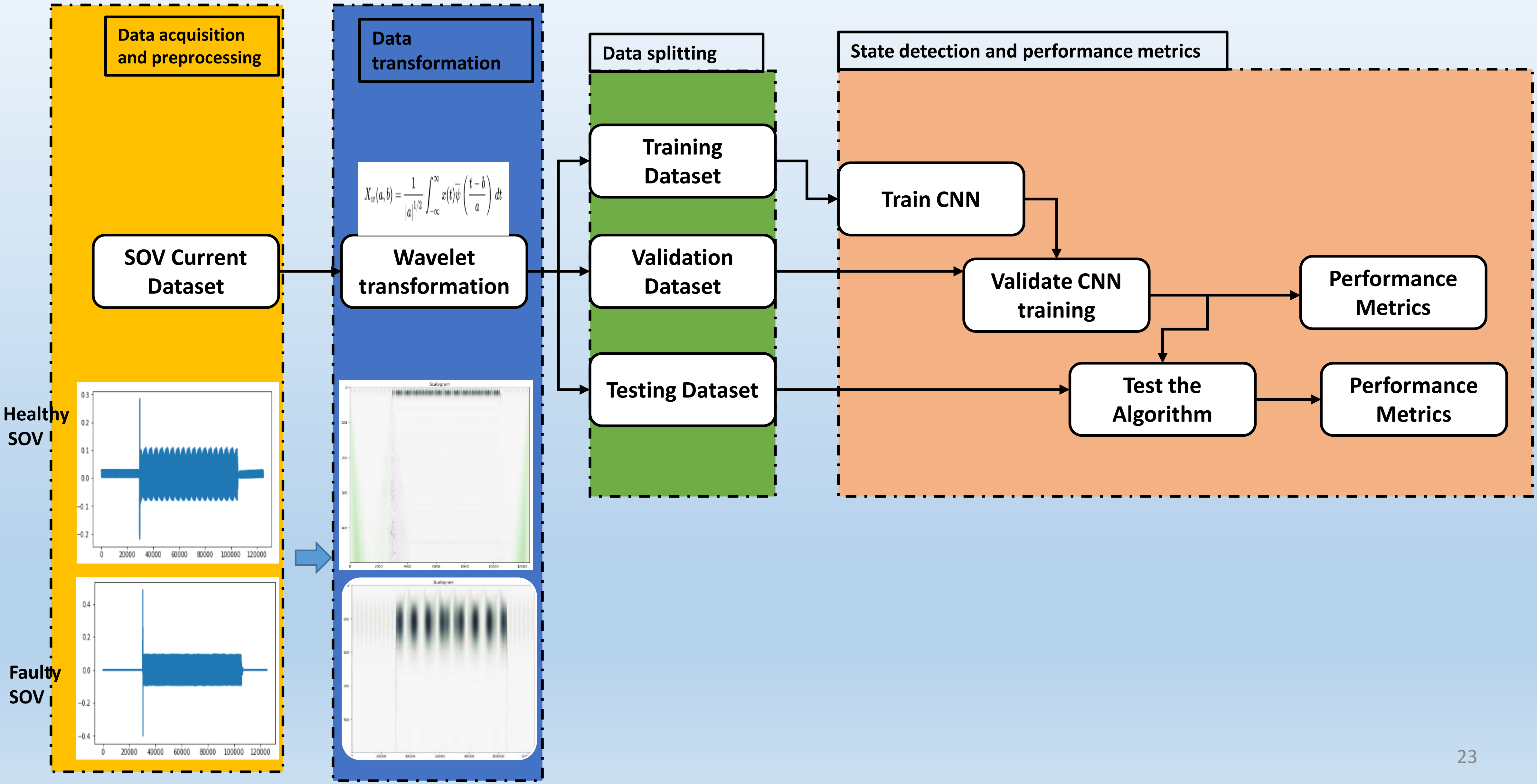
false negative: An incorrect decision that a condition does not exist when it actually does exist.

false positive: An incorrect decision that a condition exists when it actually does not exist. Synonymous with the term false alarm.

Result



SOV fault detection with CNN

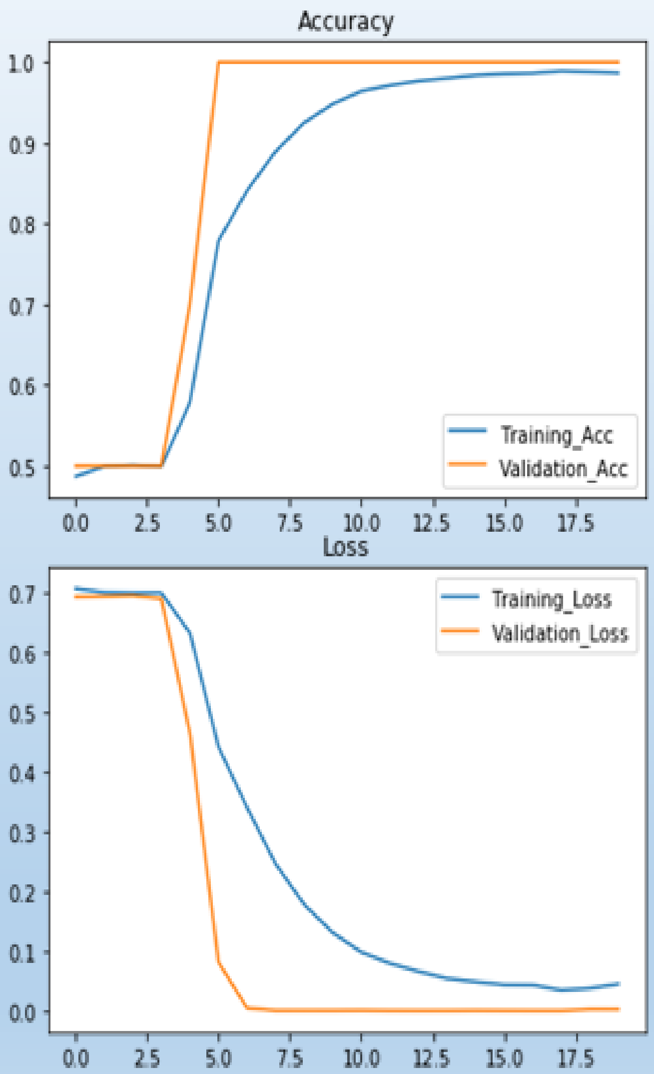


SOV fault detection with CNN

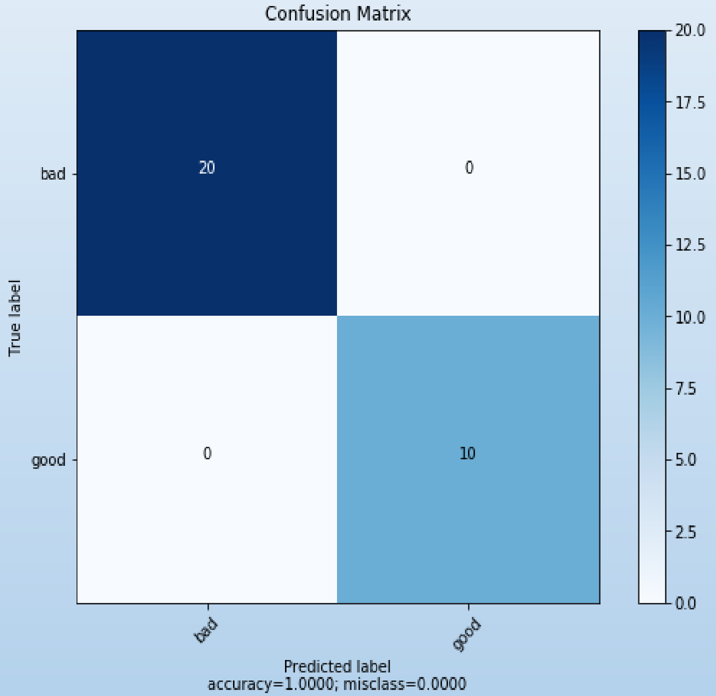
Network Parameters

Hyper parameter	Size	Number
Filter size	3 x 3	4
Max pooling	2 x 2	4
Dropouts		0.25, 0.24, 0.4, 0.5, 0.5
Activation function		LeakyReLU, Softmax (output)
Training epoch	30	
Batch size	32	
Metrics	Accuracy	
Learning rate	0.0001	
Optimizer	RMSprop	
Dense layer	2	
Output layer	1	

CNN training output



Verification with test dataset



SOV RUL Prediction

- Regression techniques is a form of predictive modelling techniques which investigates the relationship between the input parameters and the output parameters.
- Regression analysis is one of two commonly used techniques for RUL prediction, the other being particle filtering.

SOV FMEA

Component	Failure Cause	Failure Mechanism	Effect on device
Plunger	Binding of plunger in guide tube	Contaminants between guide tube and plunger	Sluggish operation or failure to operate
Valve body	Blockage of exhaust outlet	Contaminants or debris build up	Blocked or poor flow of fluid.

Experimental Cases for SOV Degradation

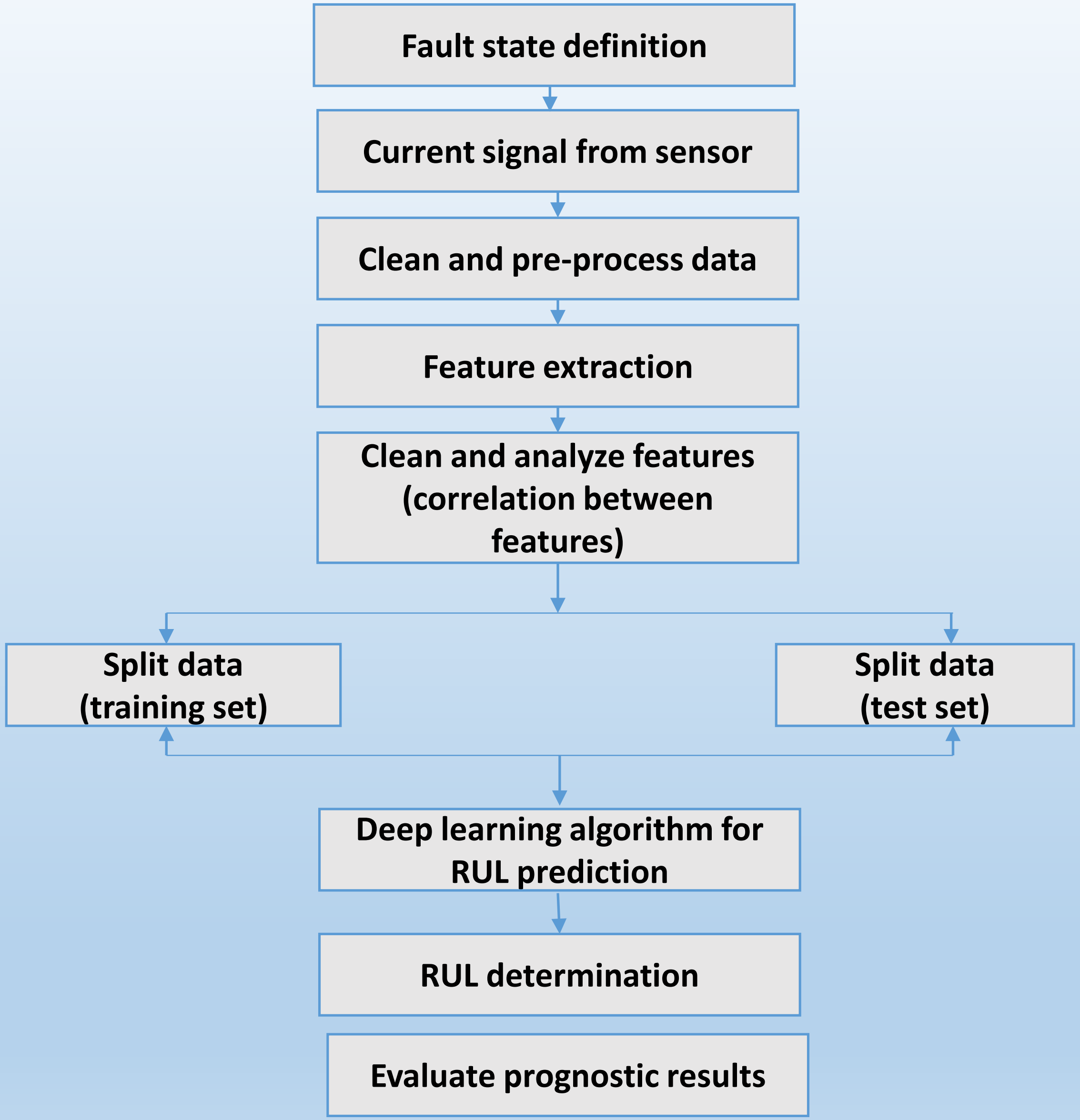
Cases	Name	Description
1	Normal	Normal, for the sake of the experiment was to allow process to flow out without any blockage while being pressurized.
2	Plunger Degradation 1,2,3 and 4	The tests performed for this case involved introducing debris between the plunger and the guide tube. The introduction of debris involved using a very thin material of thickness of 0.1mm. This was done to simulate a faulty condition.
3	Fail Open	This is a faulty state of the SOV in which the SOV fails to operate when a control signal is sent. The plunger was stuck closed.

SOV RUL Prediction with Deep Neural Network

- **Deep neural networks** also be referred to as deep feedforward neural network, feed forward neural networks and multilayer perceptrons (MLP) are the most typical representation of deep learning models.

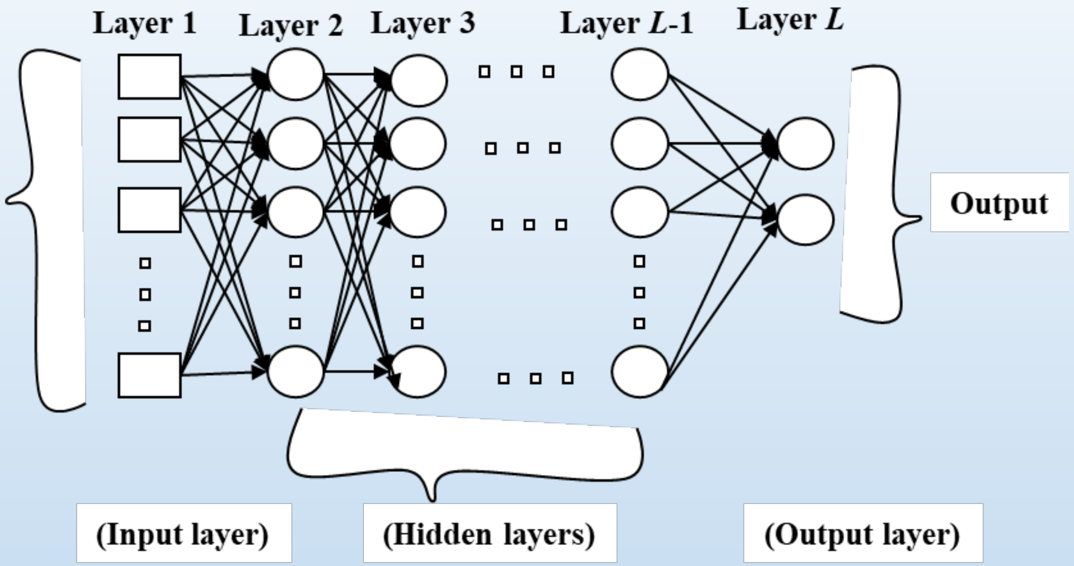
- **Deep learning** is a class of machine learning techniques where many layers of information processing stages in hierarchical architectures are exploited for pattern classification and for feature or representation learning.

- The **Deep Neural Network (DNN)** is generally a stack of multiple hidden layers instead of only one hidden layer in the standard ANN architecture.



Prognostic framework for RUL determination

SOV RUL Prediction with Deep Neural Network



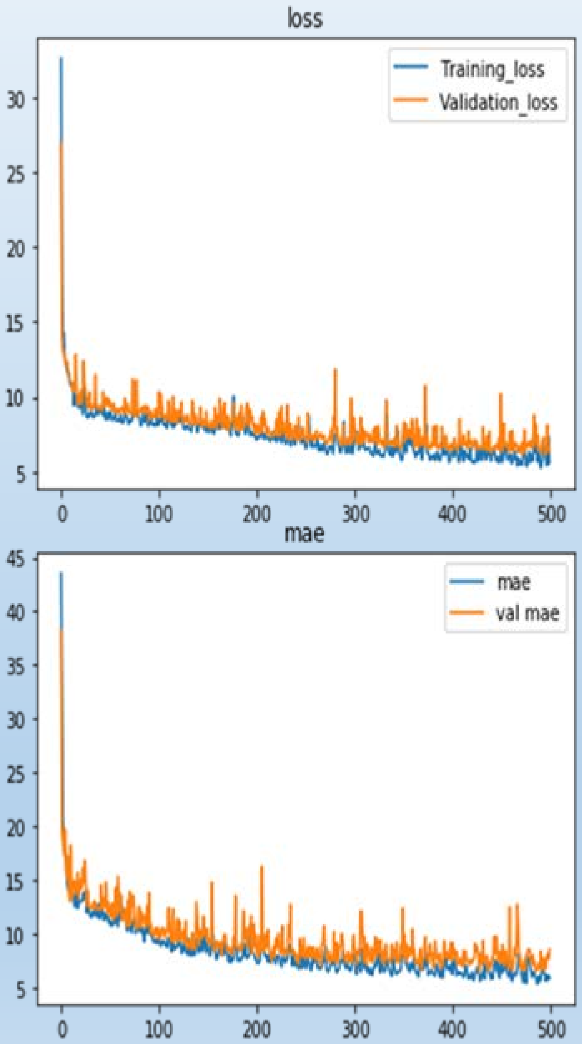
Typical DNN architecture

DNN network parameters

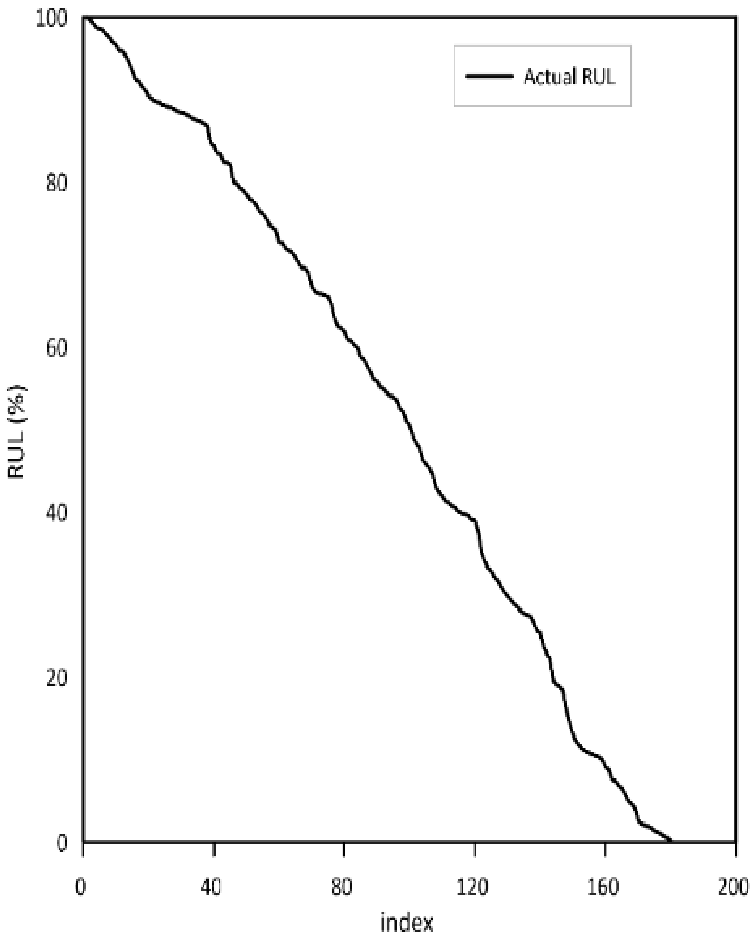
Parameter	Size	Number
Input dense layer	128 neurons	1
Hidden layers	256 neurons	6
Kernel initializer	normal	
Activation function	Input, hidden layer: ReLU Output: linear	
Output dense	1	1
metrics	Mean absolute error, mean square error	

SOV RUL Prediction with Deep Neural Network

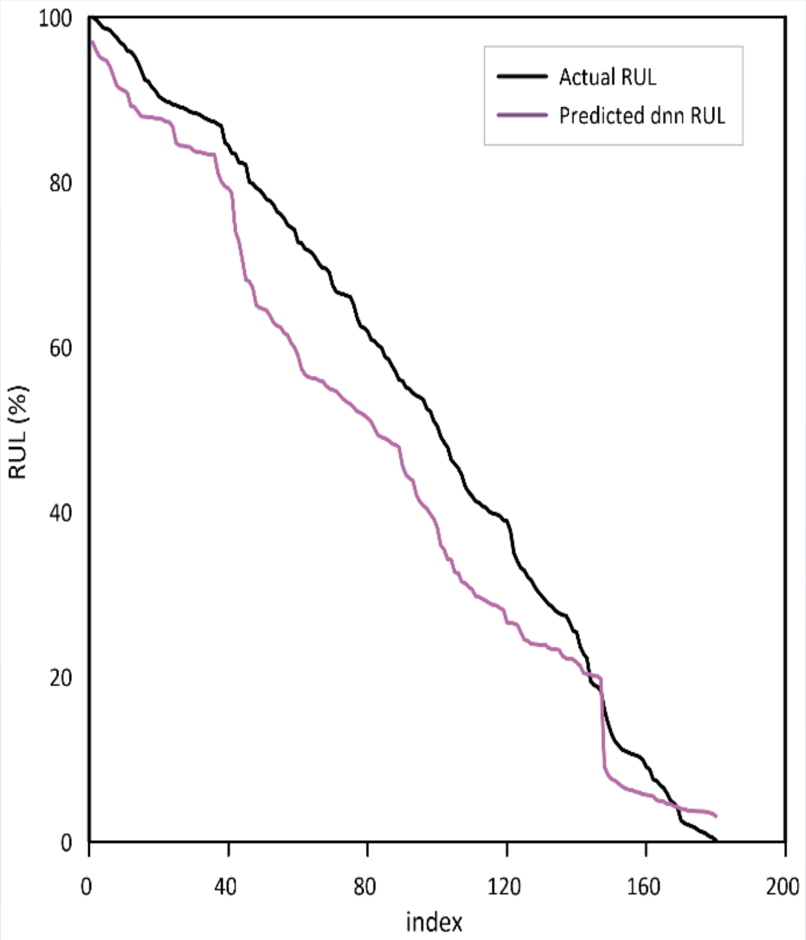
DNN training results



Actual RUL prediction

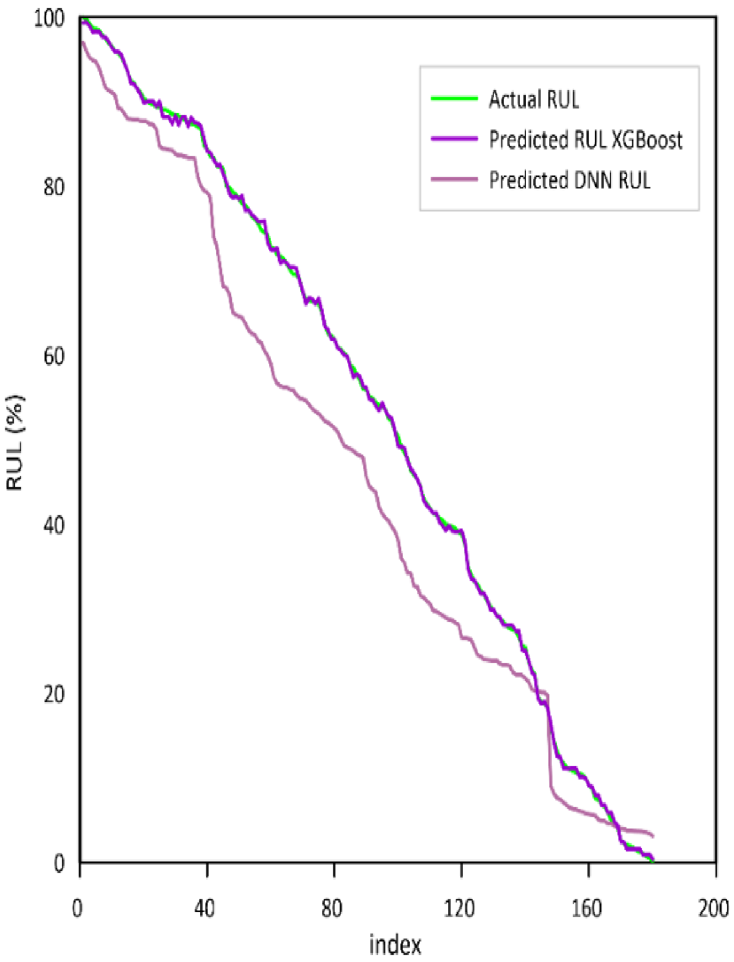


Actual RUL prediction vs Predicted RUL (DNN)



SOV RUL Prediction with Deep Neural Network

Comparison with other Regression algorithm



XGBoost regressor vs DNN vs Actual RUL output

Evaluation metrics of RUL algorithms

- **Mean absolute error (MAE):** This measure measures how close the predictions are to the actual output. The smaller the MAE, the better the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |(y_i - \bar{y}_i)|$$

- **Mean squared error (MSE):** This metric is similar to the MAE. MSE measures the average of the squares of the errors in the deviations from the actual RUL. The closer this value is, the better.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y}_i)^2$$

- **Root mean squared error (RMSE):** This is the square root of the MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$$

Performance comparison

	DNN Regressor	XGBoost
MAE	7.499388445197211	0.33273581431839216
MSE	102.47562603057746	0.18303516281395382
RMSE	10.123024549539405	0.42782608944985323

Summary

- Prognostics and health monitoring is being widely used in various engineering systems and its application in the NPP would also be beneficial just as in other fields.

- The overview of PHM as related to data driven approach has been discussed.

- Some PHM examples and current project being worked on were discussed.

Thank You


KINGS
Uniqueness Is
Better Than The Best
KINGS
KEPCO INTERNATIONAL
NUCLEAR GRADUATE SCHOOL
Scroll Down
Top