Data Dimensionality Reduction Strategy for Fault Diagnosis in PV Panels Using Machine Learning Algorithms

CHOKR Bassel – 1st year PhD Student
Supervised By : N.Chatti – A.Charki – T.Lemenand
LARIS Laboratory – Université d'Angers
France

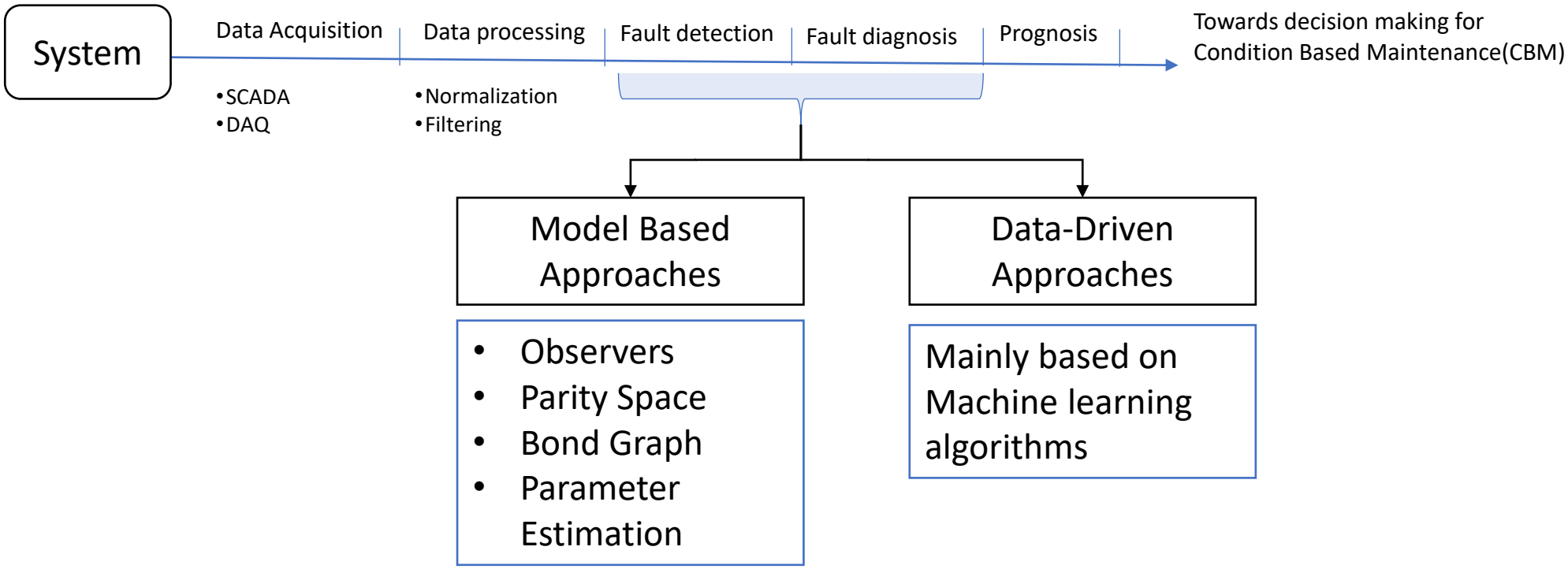
Société d'Automatique, de Génie Industriel & de Productique

Outline

1. State of the Art
2. Dataset
3. DDRS
4. Evaluation
5. Results
6. Conclusion

State of the Art

PHM : Prognostic and Health Monitoring



Both approaches are aiming on improving the availability of the studied systems (in this work it will be on PV systems)

Data Driven : Machine Learning

Supervised

Labeled Data

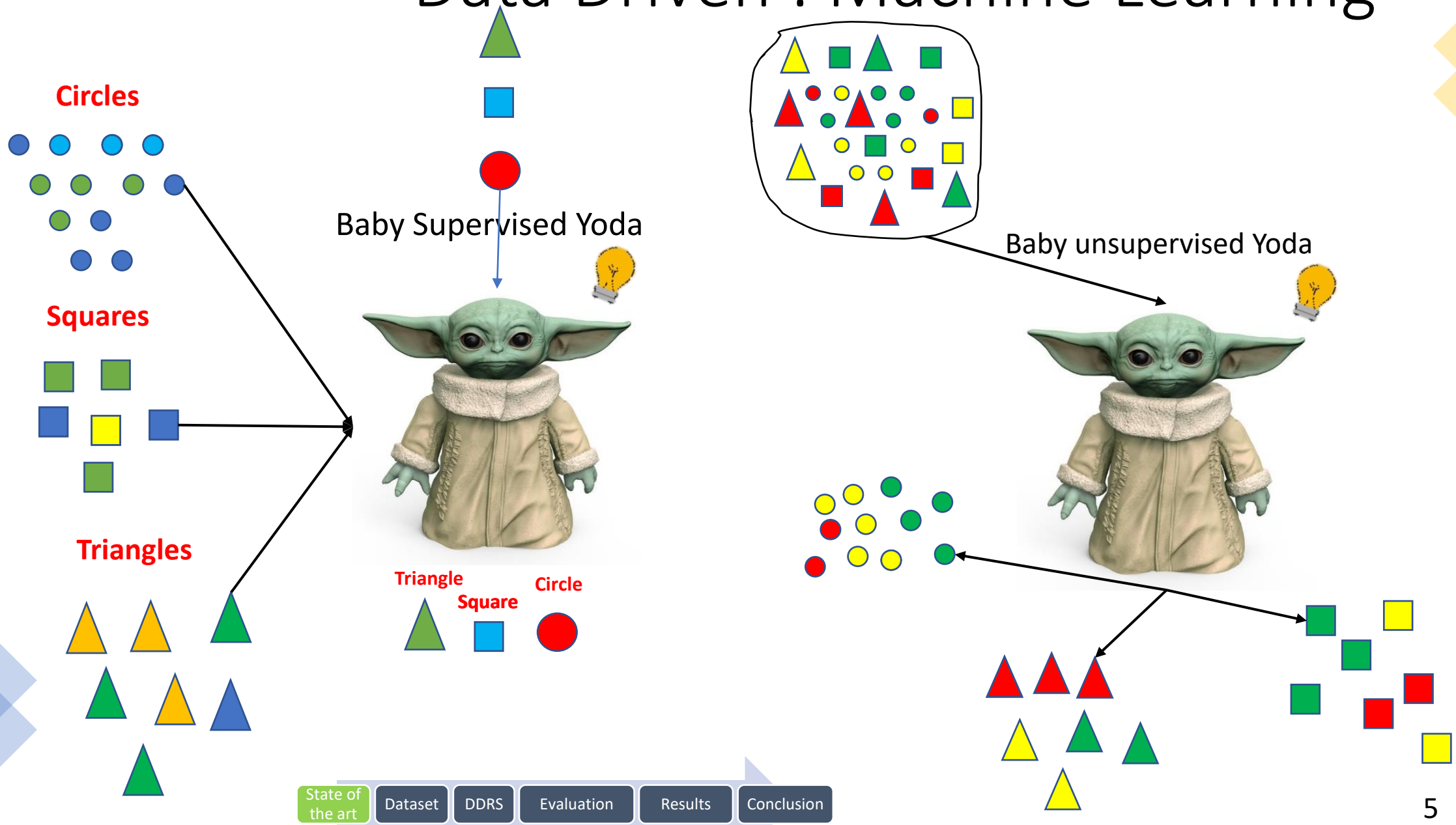
- Classification and Regression Applications
- SVM
- Decision Tree
- Logistic Regression
- Bayesian Network

Unsupervised

Unlabeled Data

- Clustering and Association
- K-mean clustering
- KNN

Data Driven : Machine Learning



Data Driven : Machine Learning

Supervised

Labeled Data

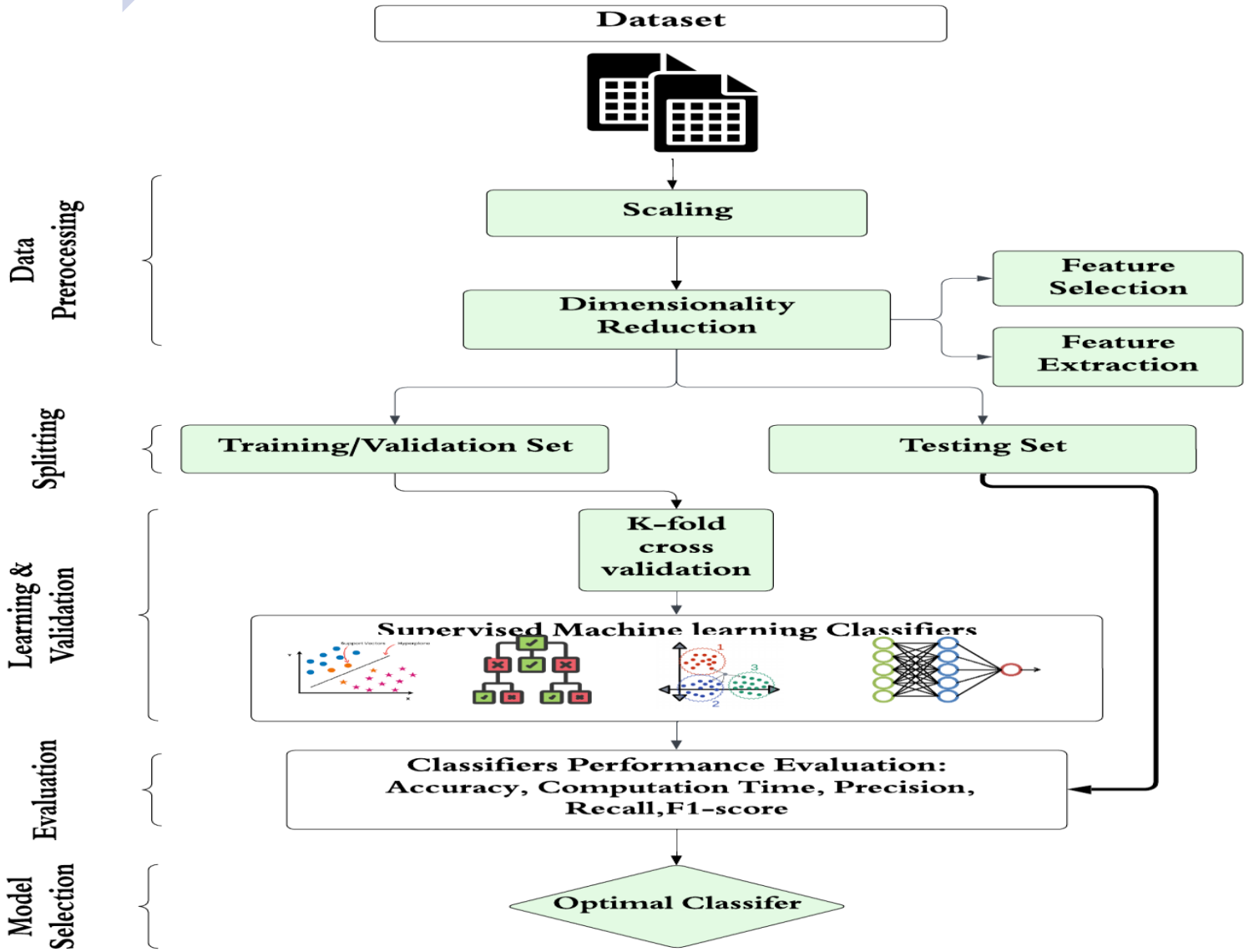
- Classification and Regression Applications
- SVM
- Decision Tree
- Logistic Regression
- Bayesian Network

Unsupervised

Unlabeled Data

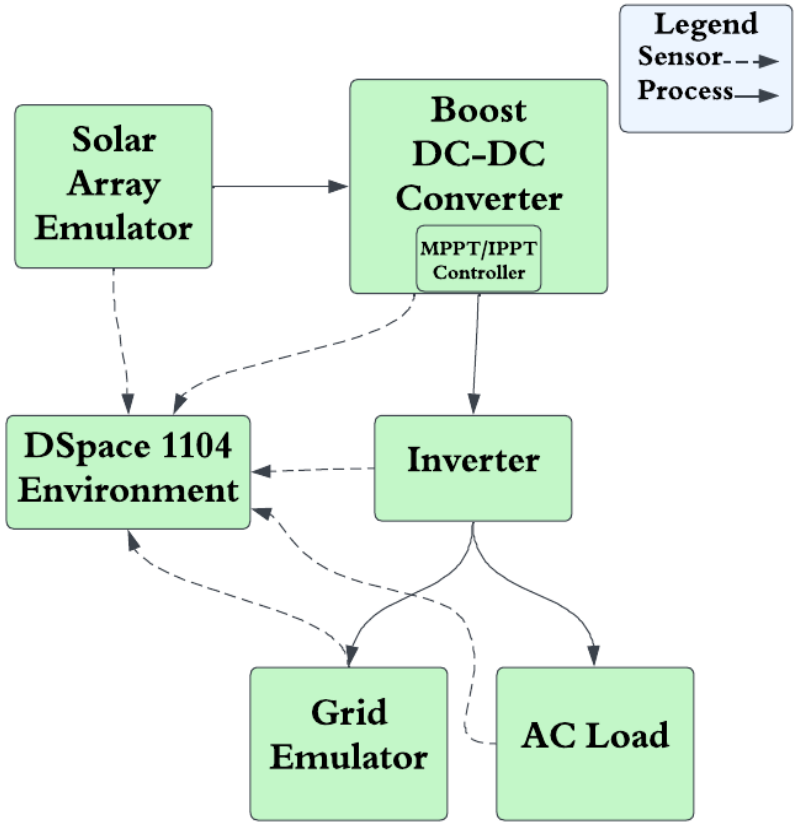
- Clustering and Association
- K-mean clustering
- KNN

Supervised ML Pipeline



GPVS Open-source

- Dataset was created experimentally in a lab and referenced by Azzdine et al [1]
- Data is divided to 8 modes , 1 for normal operating mode and 7 others each representing a fault
- Each mode is monitored for an average of 15 seconds and the data is recorded, with a sampling rate of 1000 samples/second



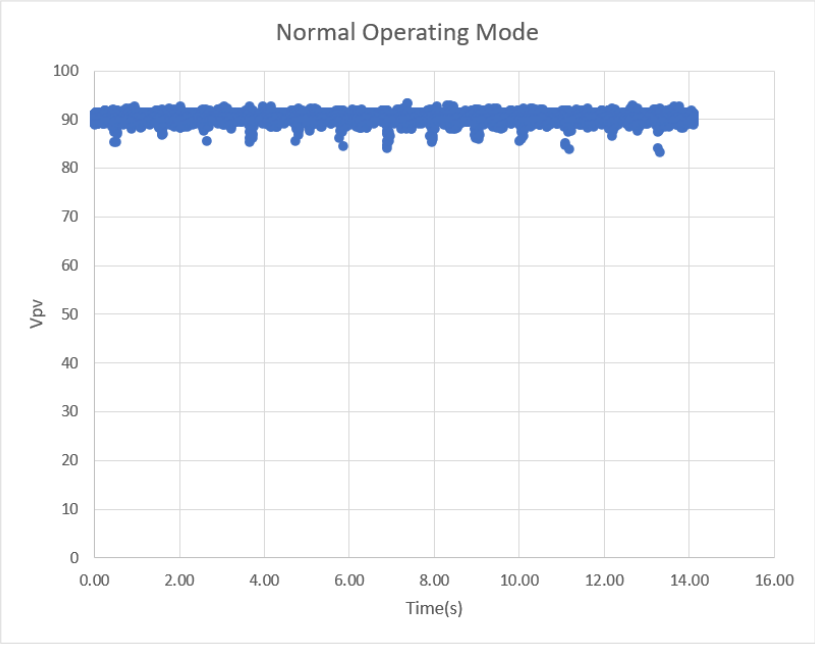
[1] B. Azzeddine, B. Wahiba, G. Amar, and M. Saad, “Real-time fault detection in PV systems under MPPT using PMU and high-frequency multi-sensor data through online PCA-KDE-based multivariate KL divergence,” *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106457, Feb. 2021

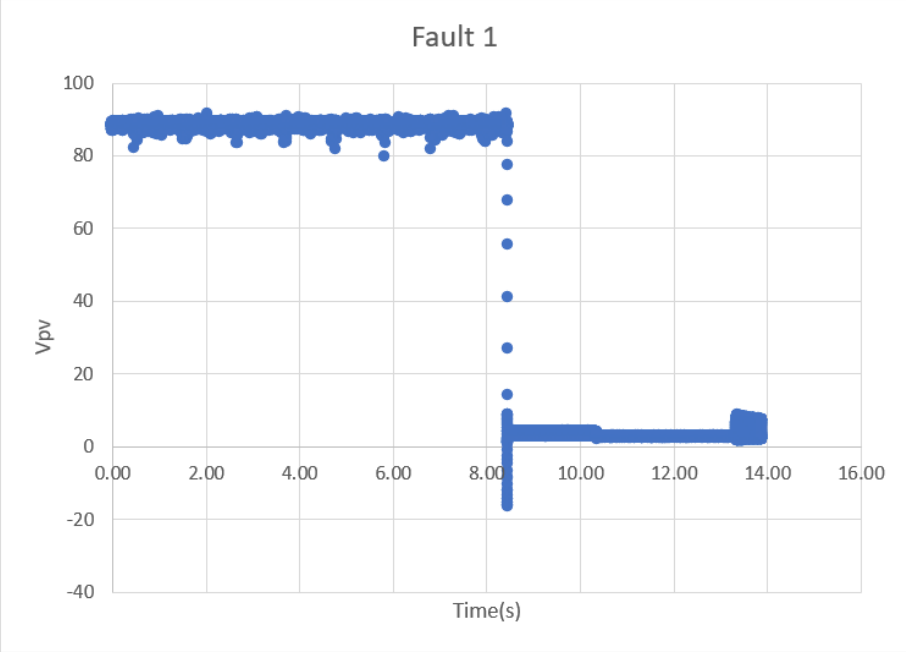
GPVS Dataset

Label	Case	Number of acquired samples	Description
F0	Normal Operating Mode	61006	No Fault injected
F1	Inverter fault	59006	Complete failure in one of the six IGBTs
F2	Feedback current sensor fault	64007	One phase sensor fault with 20% error.
F3	Grid anomaly	2961	Intermittent voltage sags
F4	PV array mismatch	79008	10 to 20% nonhomogeneous partial shading
F5	PV array mismatch	79007	15% open circuit in PV array
F6	MPPT/IPPT controller fault	39004	-20% gain parameter of PI controller in MPPT/ IPPT controller of the boost converter
F7	Boost converter controller fault	54006	+20% in time constant parameter of PI controller in MPPT/IPPT controller of the boost converter

Labeling the data

- Faults are induced during an unspecified time within operation
- Time of fault was approximated based on Azzedine et al work
- All data after fault induction are labeled as faulty by the name of the file





Grid Connected PV System (GPVS)

Ipv	Vpv	Vdc	ia	ib	ic	va	vb	vc	Iabc	If	Vabc	Vf	Label
2.214691	90.59448	147.0703	0.65039	-0.57739	-0.13343	-146.51	119.0256	26.43941	0.714643	49.99293	155.0793	50.00809	F0
2.176849	90.21606	147.0703	0.636962	-0.47668	-0.20728	-147.028	115.9155	29.6258	0.714386	50.0024	155.0849	50.00797	F0
2.266724	90.49072	147.0703	0.657104	-0.55725	-0.16028	-149.764	112.4077	33.98148	0.714386	50.0024	155.0849	50.00797	F0
2.348083	90.28931	147.6563	0.643676	-0.44983	-0.25428	-150.825	110.1656	39.10461	0.714386	50.0024	155.0849	50.00797	F0
2.315918	89.9231	147.0703	0.670531	-0.52368	-0.19385	-151.958	105.5728	43.45225	0.714232	50.01188	155.0898	50.00786	F0
2.212799	90.22217	147.3633	0.65039	-0.42297	-0.30127	-153.043	103.0173	48.50306	0.714232	50.01188	155.0898	50.00786	F0

Data Scaling

Data was normalized using z – score :

$$Z = \frac{x_i - \mu}{\sigma}$$

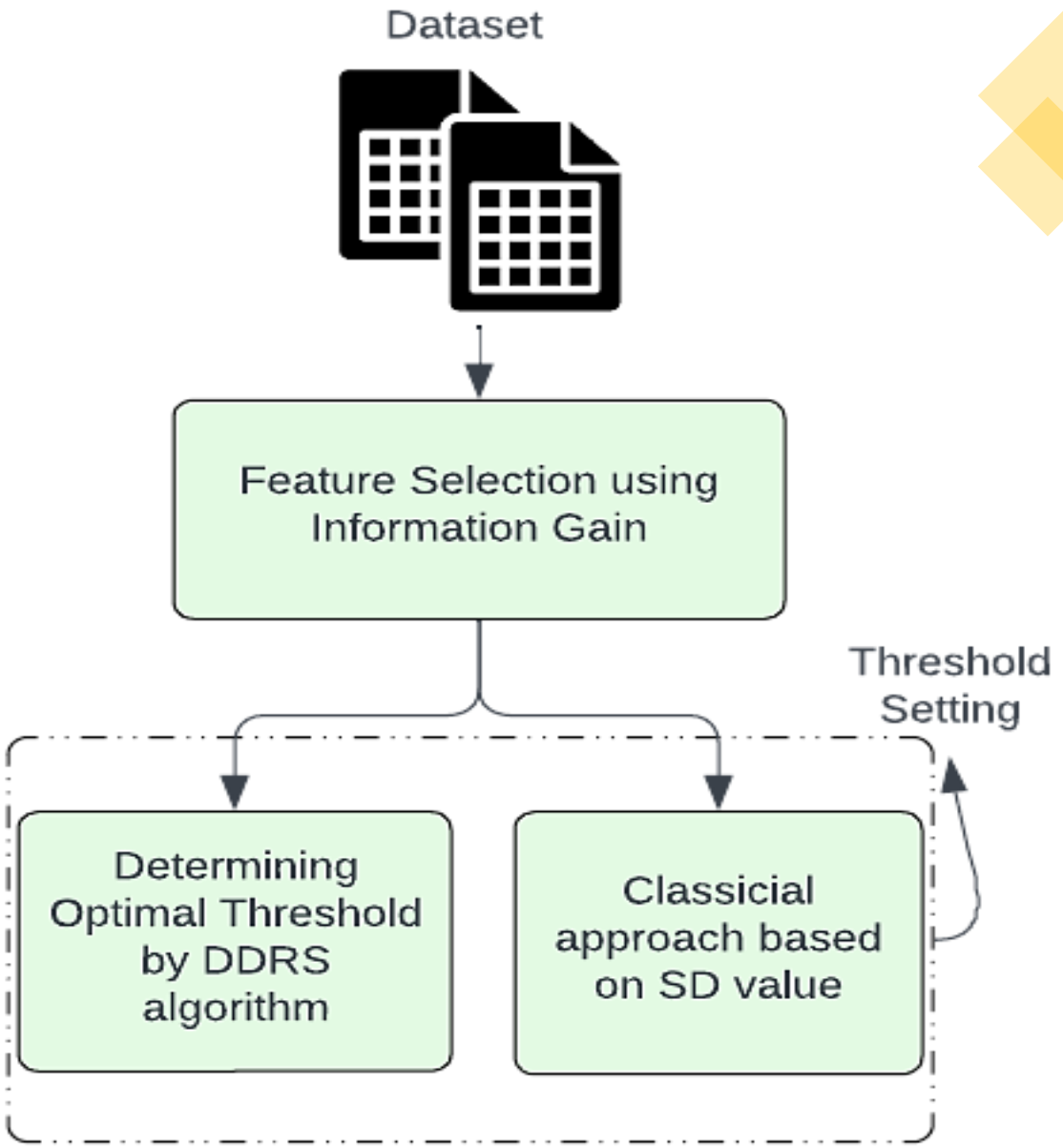
The new values will scale in a range of $\mu = 0$ and $\sigma = 1$

Data Splitting

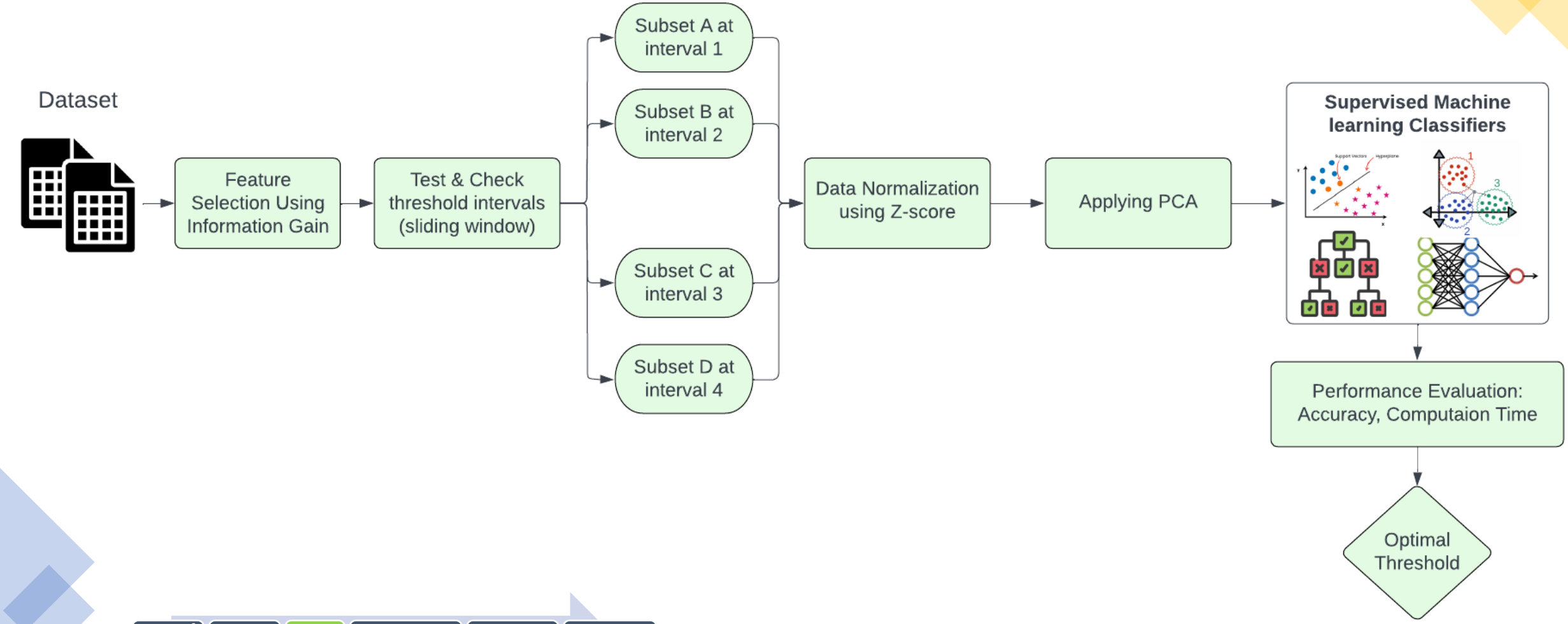
- Data is split based on 70/30 training/testing ratio
- For training/validation k-fold cross validation method with k = 5 is used

Problem Statement

- Based on interest on working on feature engineering
- Information gain and PCA are used
- Information gain is calculated using standard deviation as a threshold to select features subset
- A strategy (DDRS) is proposed to find the optimal threshold for feature selection using information gain
- Classifiers are evaluated based on DDRS threshold



Data Dimensionality Reduction Strategy (DDRS)



Information Gain

- A filtering feature selection method aiming to determine the significance of a certain feature used for classification
- Based on Shannon entropy equation :

$$E(y) = -\sum_i^C p_i \log_2 p_i$$

- Information gain =

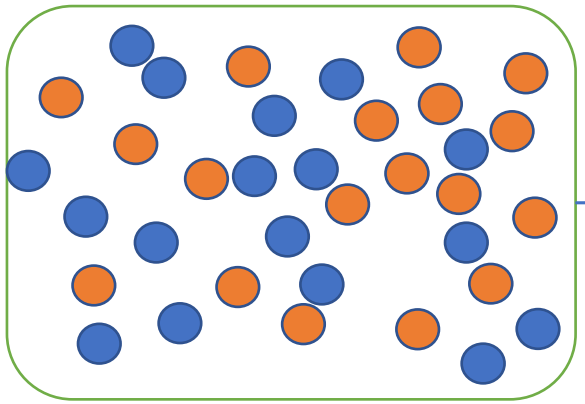
  Entropy of parent node – weighted entropy of child nodes

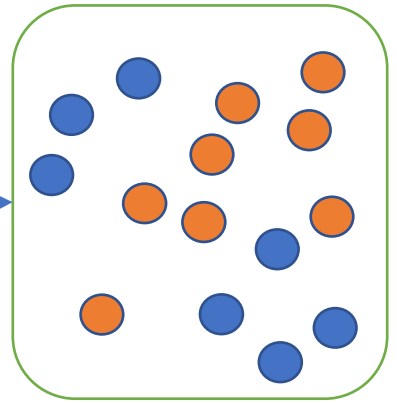
Information Gain

O: 18

B : 17

$$E(a) = -\left(\frac{18}{35}\cdot \log_2 \frac{18}{35}\right) - \left(\frac{17}{35}\cdot \log_2 \frac{17}{35}\right) = 0.999$$

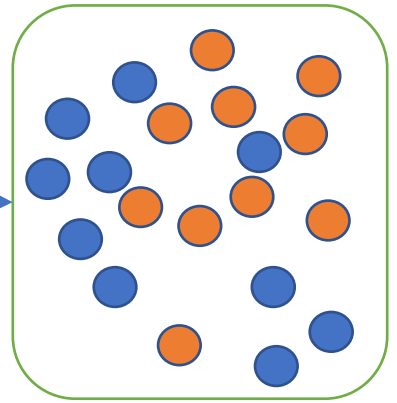




O: 8

B : 7

$$E(b) = -\left(\frac{8}{15}\cdot \log_2 \frac{8}{15}\right) - \left(\frac{7}{15}\cdot \log_2 \frac{7}{15}\right) = 0.997$$



O: 10

B : 10

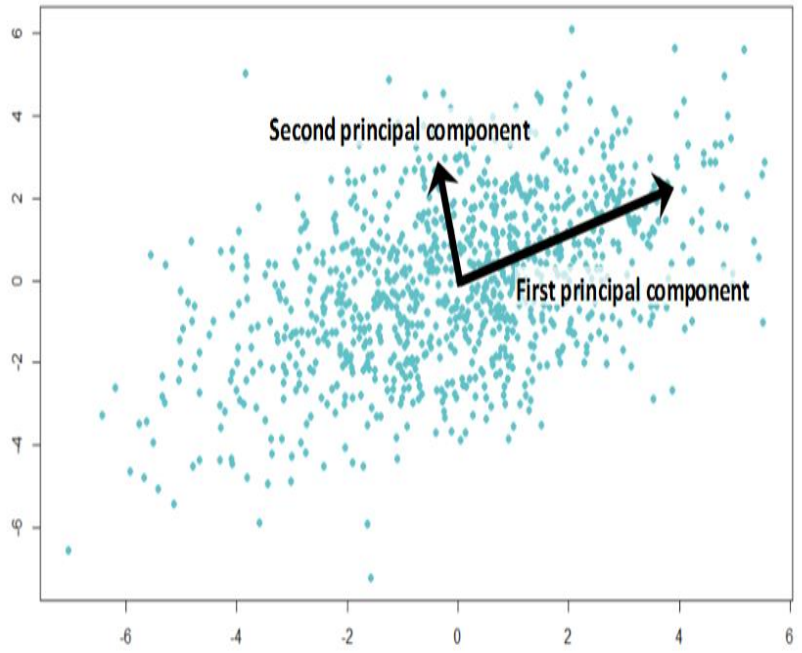
$$E(c) = -\left(\frac{10}{20}\cdot \log_2 \frac{10}{20}\right) - \left(\frac{10}{20}\cdot \log_2 \frac{10}{20}\right) = 1.0$$

$$IG = E(a) - weighted\ entropy = 0.102$$

$$Weighted\ Entropy = \frac{15}{35}\cdot E(b) + \frac{20}{35}\cdot E(c) = 0.897$$

Principal component analysis (PCA)

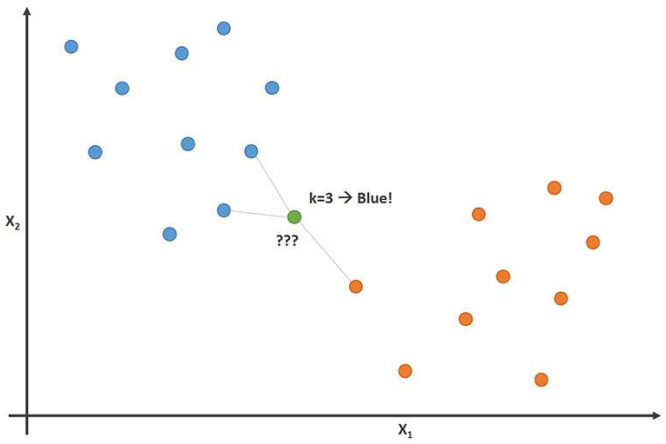
- A feature extraction method aiming to transform the set of features into a new one based on principal components(PCs).
- Principal components are eigen vectors representing the original data
- PCs represent the original dataset in an increasing accumulative variance ; thus, the more principal components selected the more variance (information) is represented
- The first component represents the highest variance, the second PC the second highest, and so on
- The possible number of components to be formed is equal to the total number of original features
- The explained variance value of each PC is calculated as following :



$$EV = \frac{eigen\ value\ of\ PC\ (eigen\ vector)}{total\ of\ eigen\ values}$$

Classifiers

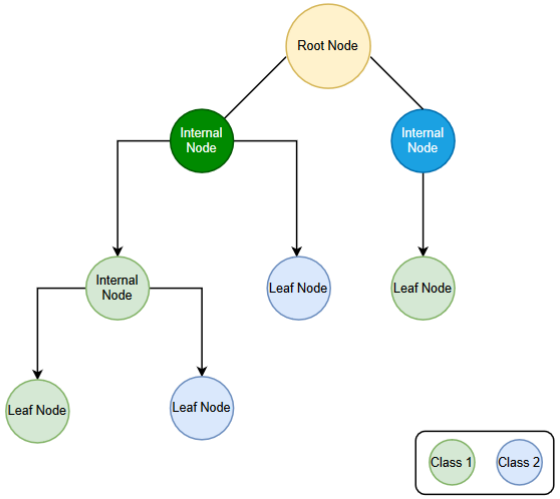
KNN



(Mierswa, 2017)

- Datapoint is labeled based on the majority vote of its nearest neighbors
- Euclidean distance is one of the ways to calculate the distance between the questioned datapoint and its neighbors
- Hyperparameters : number of neighbors K – distance (Euclidean – cosine distance)

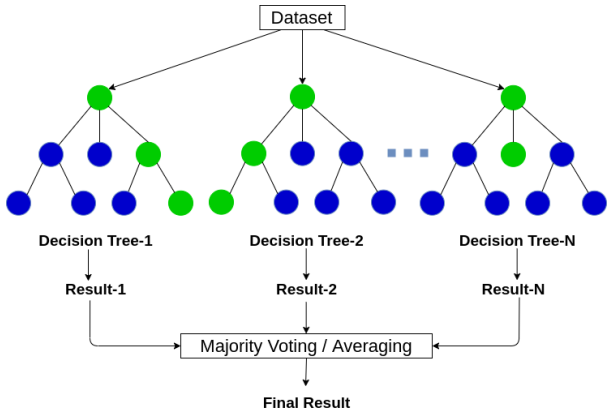
CART



(Deepankar, 2021)

- A decision tree is created from root notes, internal and leaf nodes (decision)
- A datapoint is classified based on the series of conditions it is evaluated on through the decision tree
- A decision then is specified to classify the questioned datapoint
- Hyperparameters: purity measuring criterion

Random Forest

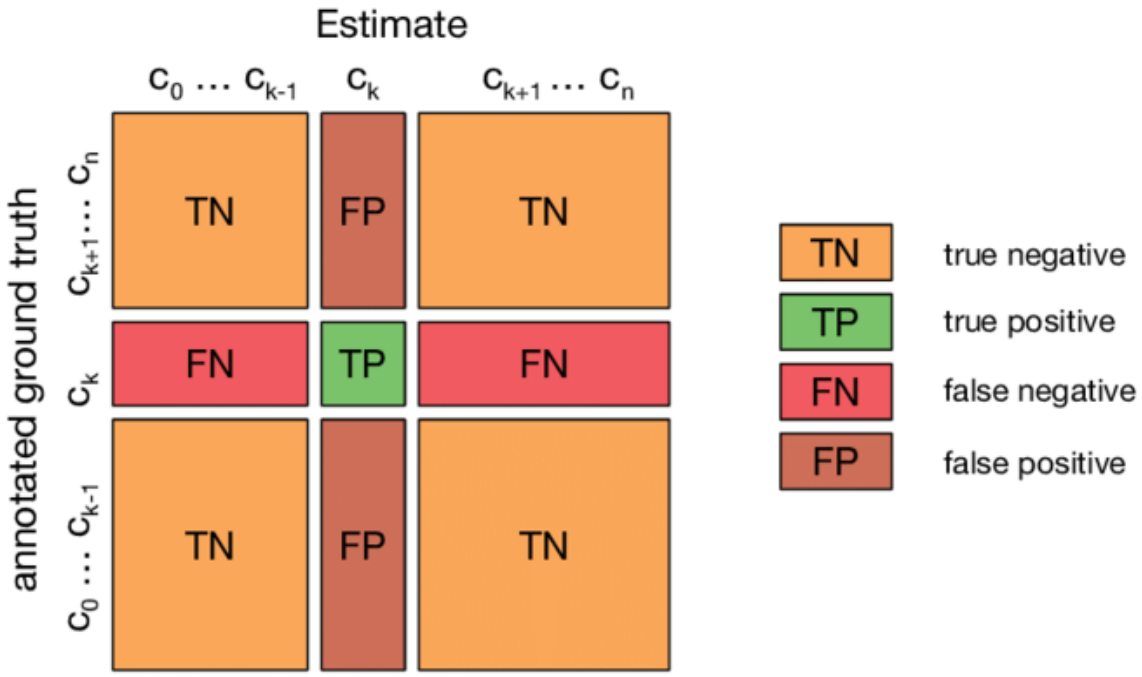


(Elbeltagi, 2021)

- Random forest is multiple decision tree, created through different nodes
- Decision of Classification is determined based on the majority vote of the decision trees
- Hyperparameters: purity measuring Criterion, number of trees

Confusion Matrix

Multi-Class Classification



(Kruger, 2016)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

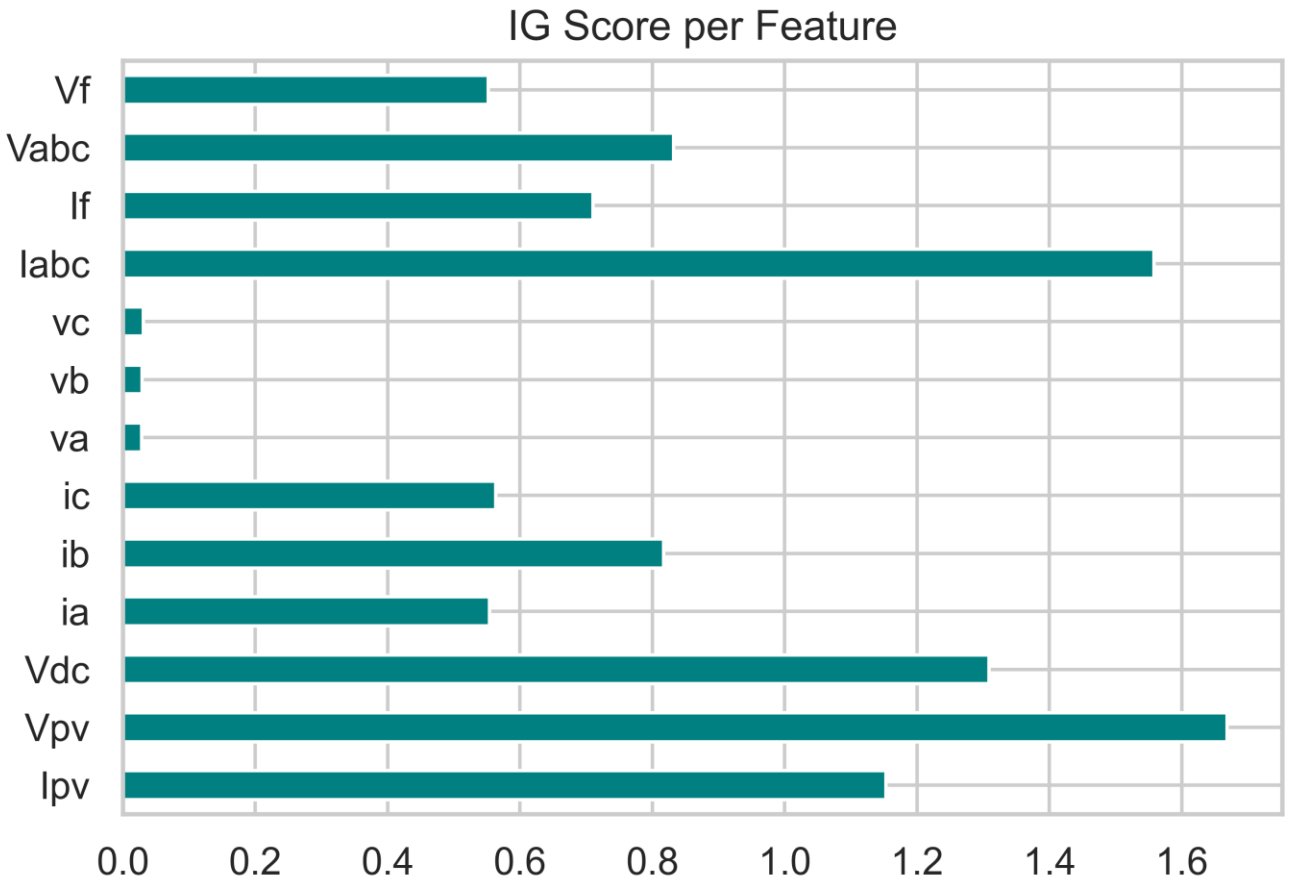
$$Recall = \frac{TP}{TP - FN}$$

$$F1 - score$$

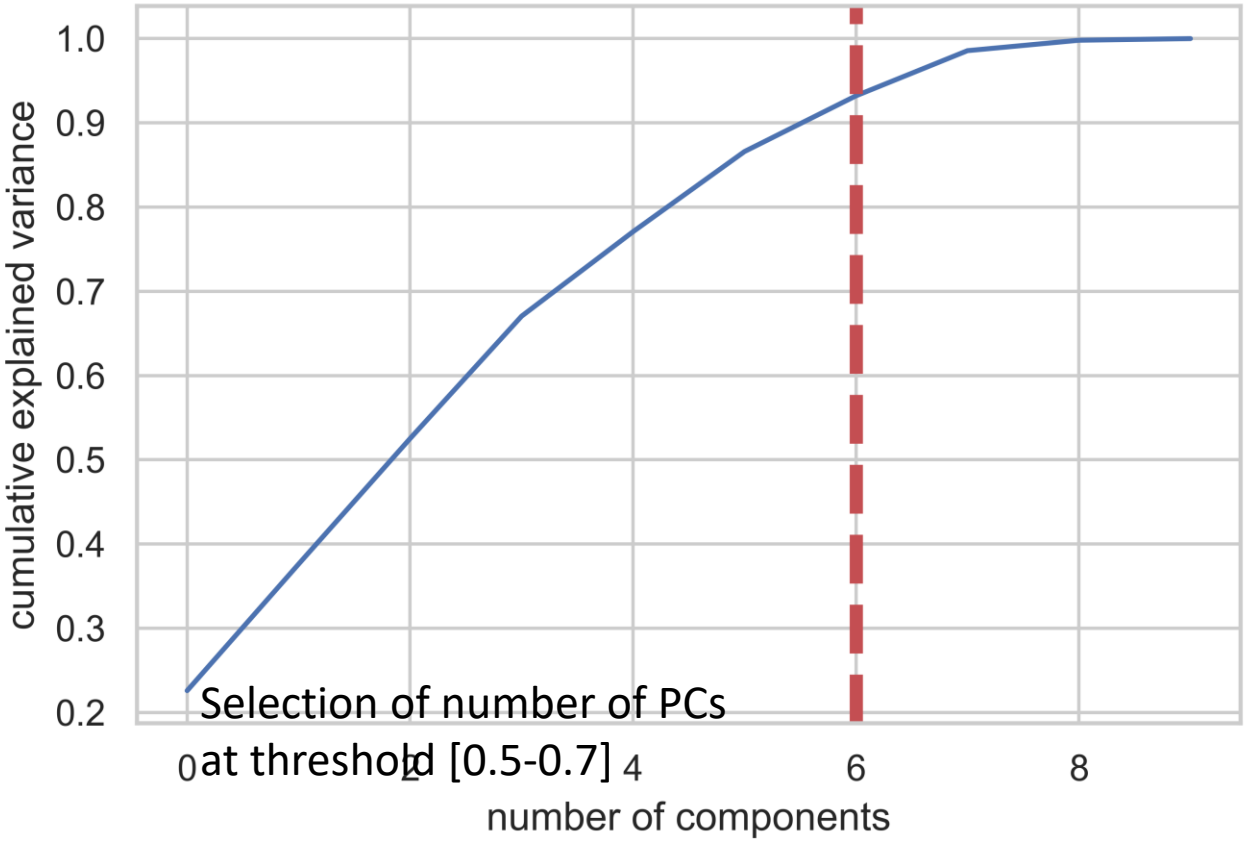
$$= 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Computation Time = Time needed for learning / prediction

Information Gain



PCA


cumulative explained variance
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
2
4
6
8
number of components
Selection of number of PCs
at threshold [0.5-0.7]

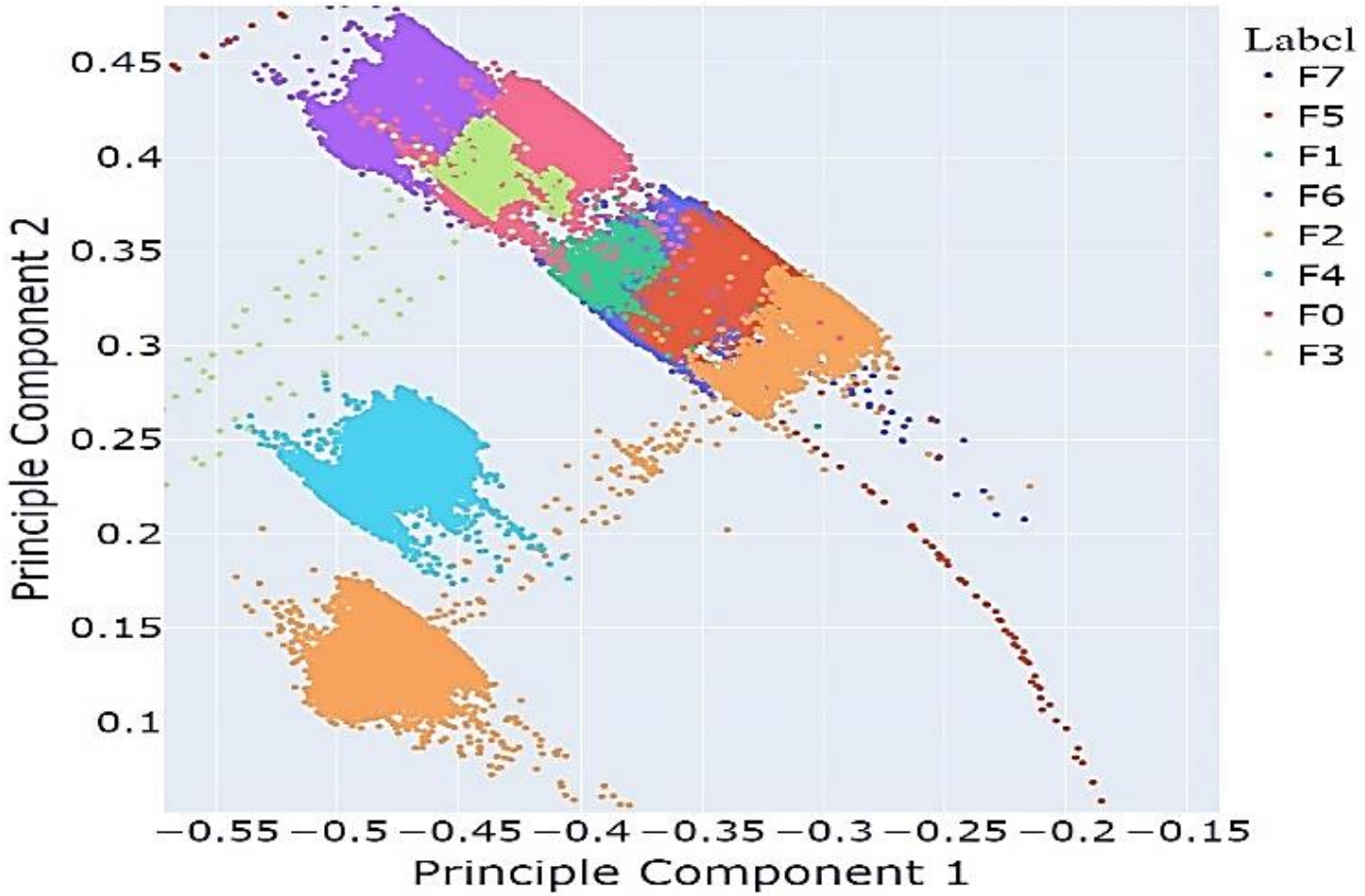
Optimal Threshold

Threshold		CART		KNN		RF	
		ACC	CT	ACC	CT	ACC	CT
Results at different thresholds	[0.1,0.3]	93.8	3.982	96.8	0.894	97.3	73.01
	[0.3,0.5]	93.8	3.903	96.8	0.723	97.3	66.41
	[0.5,0.7]	93.5	3.206	95.0	0.6	96.3	67.65
	[0.7,0.9]	95.4	2.909	95.7	1.078	97.5	67.124
	[0.9,1.1]	93.7	1.779	95.3	0.602	95.4	35.5
	[1.1,1.3]	93.5	0.968	94.9	0.408	95.0	33.330
	[1.3,1.5]	96.3	1.009	97.4	0.326	97.32	31.91
Results at SD	SD	93.8	4.851	96.8	1.515	97.27	76.137

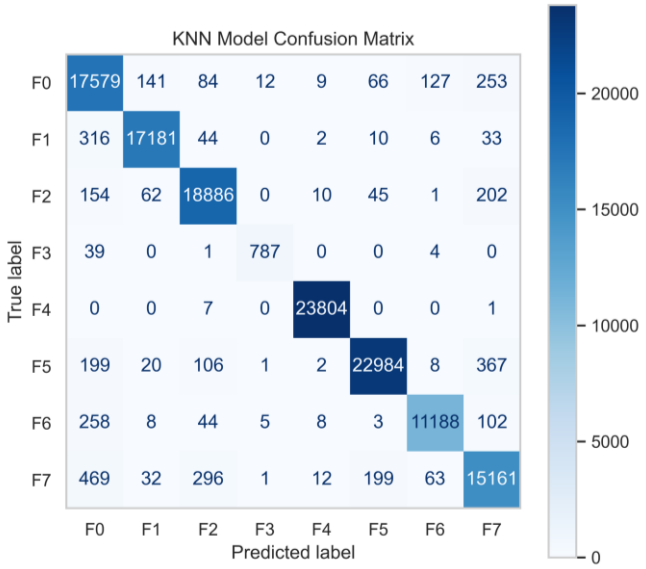
Optimal Threshold

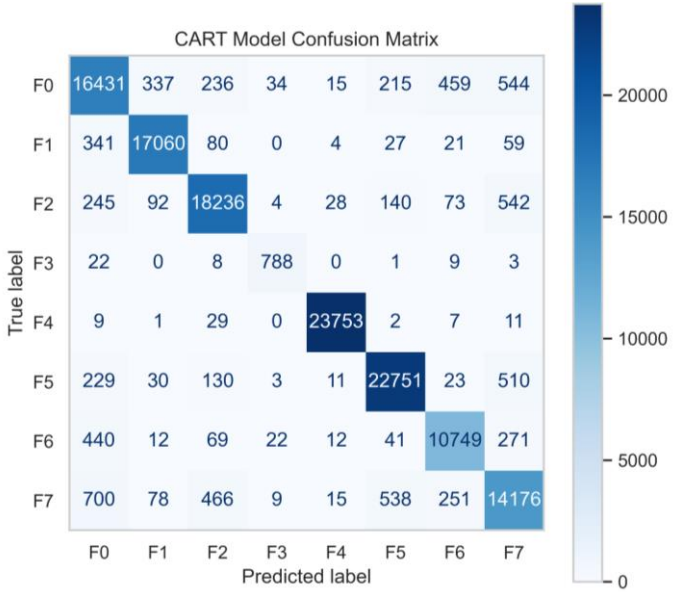
By selecting this threshold, the remaining features were two :
Vpv & Iabc

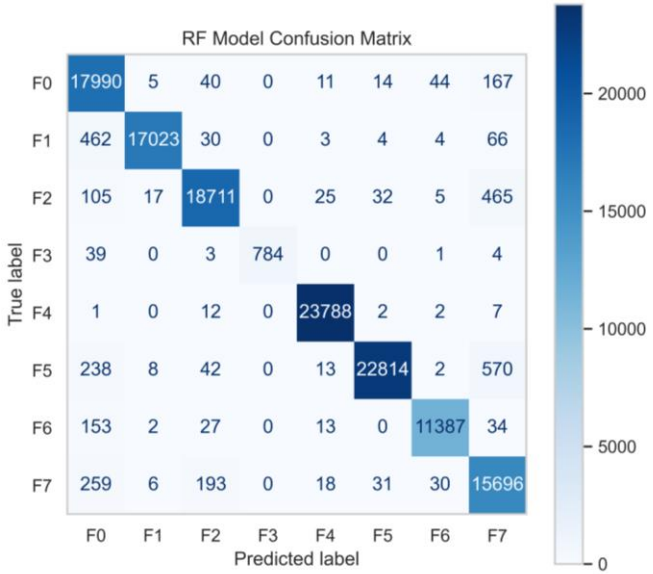
2D PCA at optimal Threshold



Confusion Matrix at Optimal Threshold







Most efficient Algorithm per Class

Label	CART			KNN			RF		
	Precision	Recall	F1score	Precision	Recall	F1score	Precision	Recall	F1score
F0	96	97	96	96	99	98	96	99	98
F1	98	99	99	99	99	99	99	99	99
F2	99	99	99	99	100	100	99	99	99
F3	90	90	90	96	89	92	97	89	92
F4	100	100	100	100	100	100	97	94	95
F5	93	94	94	97	94	95	97	94	95
F6	99	99	99	100	100	100	100	100	100
F7	90	89	89	91	95	93	92	95	93

Classifier	Cross Validation accuracy(%)	Training accuracy (%)	Testing accuracy(%)	Learning CT (s)	Prediction CT(s)
CART	96.97	99.99	97	1.072	0.015
KNN	97.36	98.33	97.88	0.464	2.17
RF	97.78	99.99	97.84	31.56	1.63

Conclusion

- The three algorithms performed well as evaluated through training testing and validation
- Based on testing accuracy KNN scores the highest value 97.88
- CART is the fastest to predict the faults, taking 0.015 seconds to classify and diagnose the faults
- DDRS proved that IG threshold is better to be tested in order to select optimal threshold
- DDRS also proved that if a threshold is optimal for a given algorithm, it can be generalized to other algorithms

Future Work

- Using DDRS on datasets with more features
- Currently working on a dataset with 80 features collected in 2017 from a wind turbine power system
- FDD study on wind turbines
- Developing an AI tool working on prognosis to predict the faults before occurring in addition to estimating the RUL of PV systems.

Contribution:

Paper under revision submitted to Solar Energy Journal - Elsevier

Q & A

Thank you