PEA

Power Quality Thailand LTD.

A ML-Based Approach for Power Quality Disturbance Analysis in EV-Penetrated Distribution Network

Wanchalerm Patanacharoenwong
Panaya Sudta, PhD
Provincial Electricity Authority: PEA N.3

Sep 8th-9th 2025

OUTLINES

(1) Background

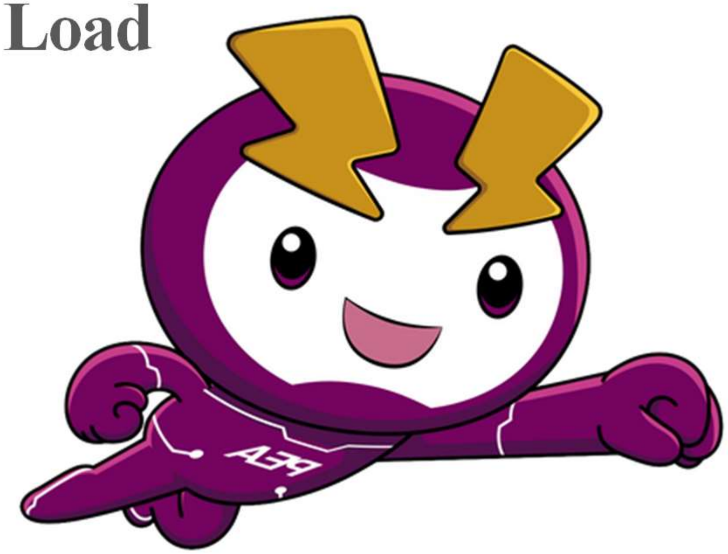
(2) Technical Implication of EV Load

(3) Methodology

(4) KNN & SVM

(5) Results & Conclusions

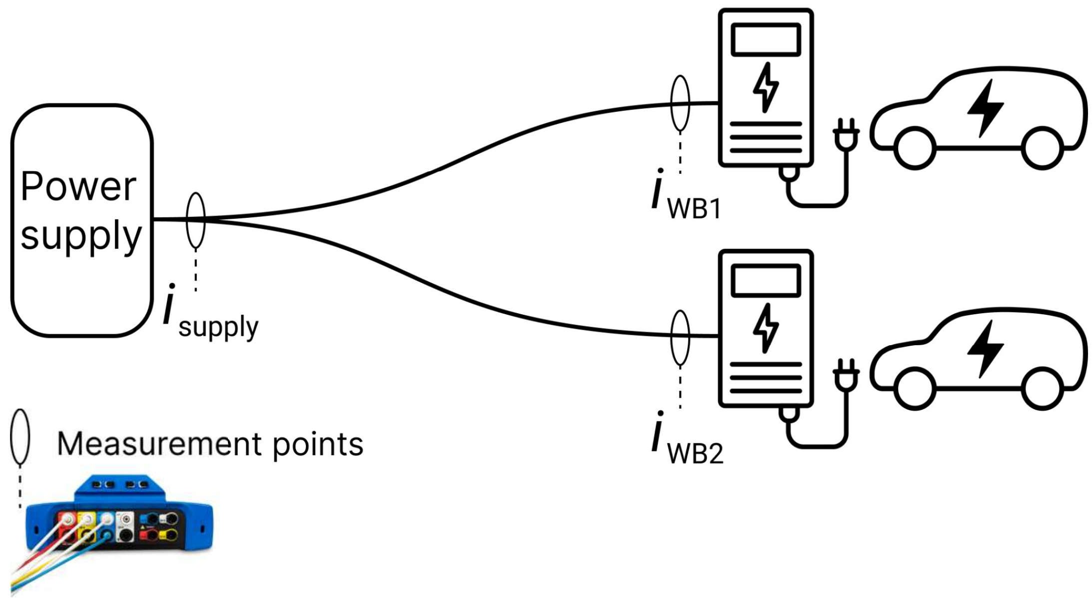
(6) Future work



Background & Expected Results

- Rise of EV adoption → higher stress on distribution grids
- EV charging = nonlinear, fluctuating loads
- Direct impact on Power Quality (PQ)

The growing integration of EV loads into distribution networks has introduced a range of power quality disturbances, including voltage sags, swells, harmonics, and flicker. These disruptions pose significant risks to grid stability and operational efficiency. This study investigates the application ML techniques for classifying PQ disturbances. Two algorithms, SVM and k-NN are evaluated for their accuracy, robustness, and reliability. Results demonstrate that while both models are effective in detecting PQ anomalies, SVM consistently delivers superior performance in terms of precision and stability, making it the more suitable choice for addressing PQ challenges in the IEEE 5-Bus system.

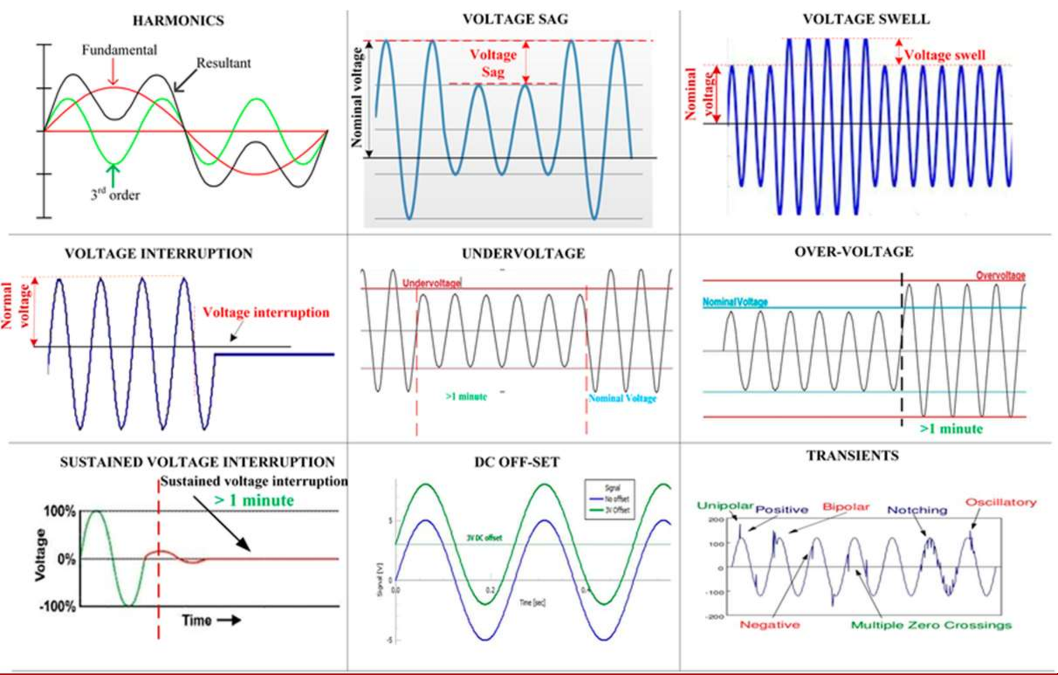


Representation of the experimental setup with measurement points.

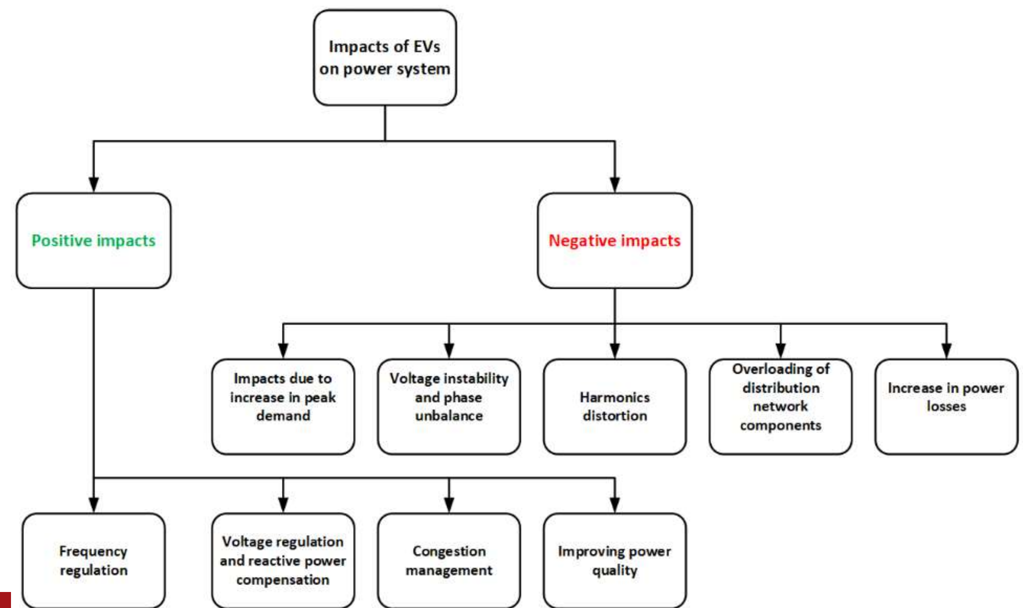
Credit: Interaction among Multiple Electric Vehicle Chargers: Measurements on Harmonics and Power Quality Issues

Power Quality Challenges

- Voltage sag & swell
- Harmonics
- Flicker
- Transients
- Short-circuits & switching events

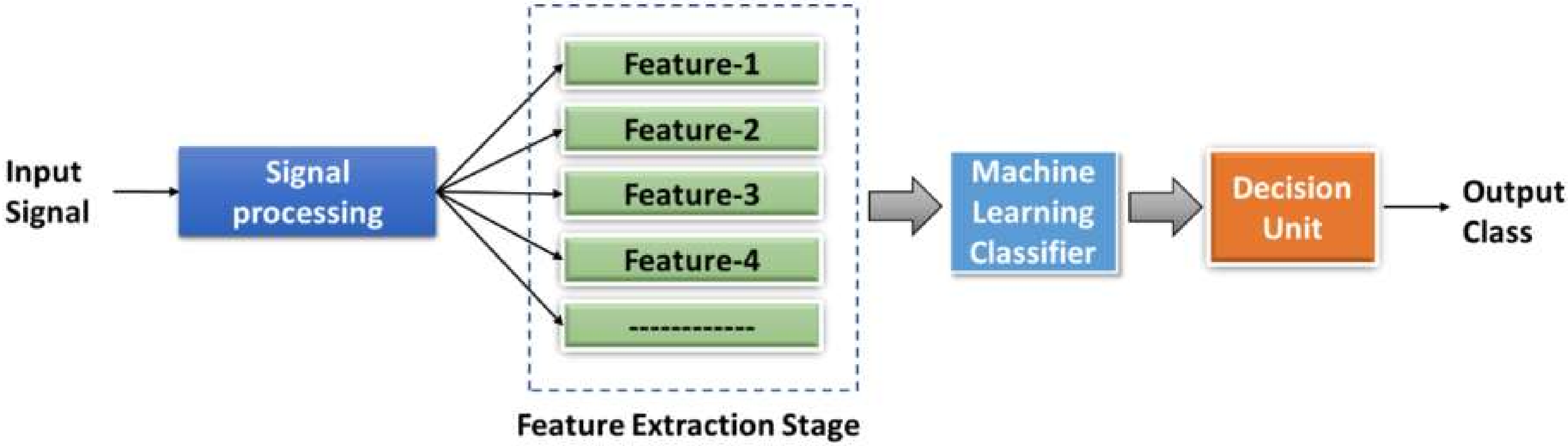


Beyond academic exploration, this research forms part of a SaaS framework designed to enhance smart grid resilience. The proposed system leverages AI models built on ML algorithms to automatically detect and issue real-time alerts when PQ disturbances arise from EV loads. This approach supports proactive grid management, enabling utilities and stakeholders to mitigate risks, maintain service quality, and improve the overall reliability of future EV-integrated power systems.



Why Machine Learning?

- Hardware-based fixes = reactive
- ML allows proactive PQ event detection
- Enables real-time classification and early mitigation.

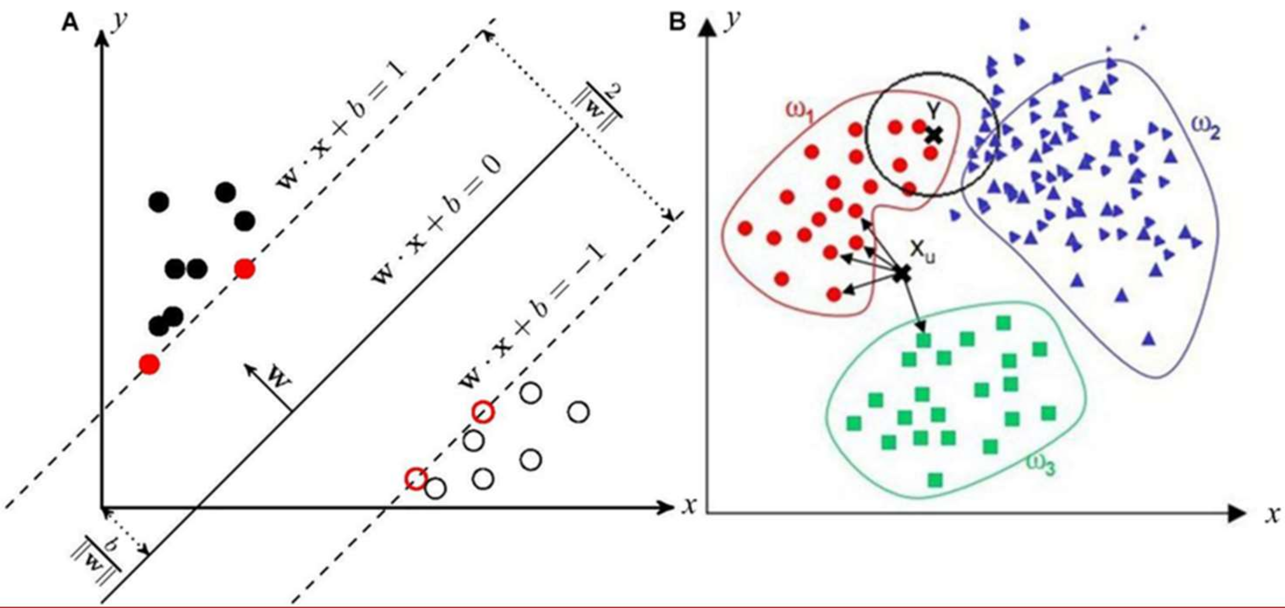


With the rising penetration of EVs, electrical distribution systems are becoming increasingly complex, rendering traditional PQ management methods inadequate. Conventional approaches, such as passive filters or voltage regulators, are largely reactive in nature and lack the flexibility to cope with rapidly changing load dynamics. This has created a demand for intelligent, real-time solutions capable of monitoring, detecting, and mitigating PQ disturbances more effectively. ML offers a promising pathway, as its algorithms can process large volumes of data, identify patterns within intricate electrical signals, and accurately classify PQ disturbances according to their type and severity.

Research Objectives

- Simulate PQ disturbances in EV-integrated networks
- Build labeled dataset of PQ events
- Compare ML models: SVM vs k-NN
- Evaluate performance across metrics

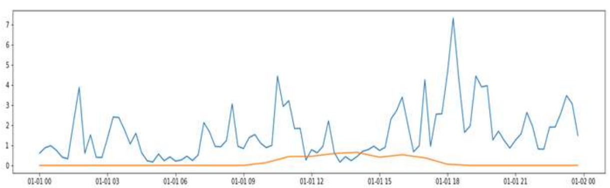
SVM

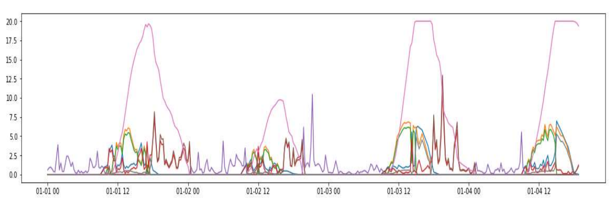


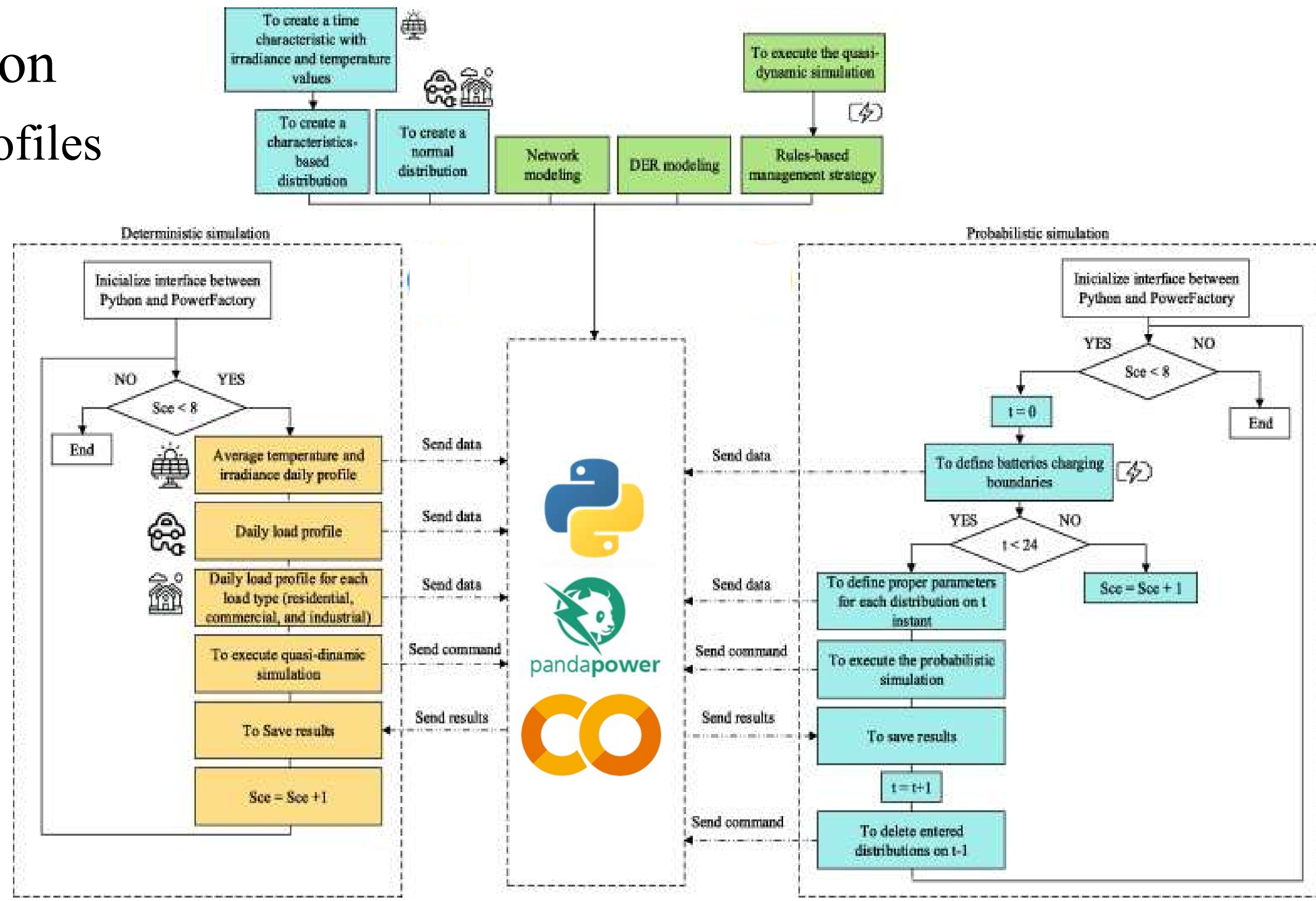
k-NN

Simulation Environment

- Program Language: Python
- Injected dynamic EV load profiles
- Captured PQ under
  normal,
  transient,
  distorted states







Problems formulations: Data Preparation

- Features: Voltage, Current, PF, Frequency, THD, Event Duration
- Samples: ~5,000 instances. Labeled by PQ event type (sag, swell, harmonics, etc.)

Data Exploration

- Visualize
- Find missing
- Look for correlations

Data Cleaning

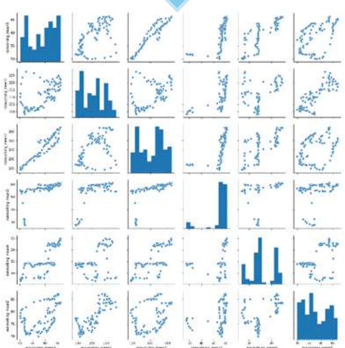
- Check: did I fix potential issues?

Model Building

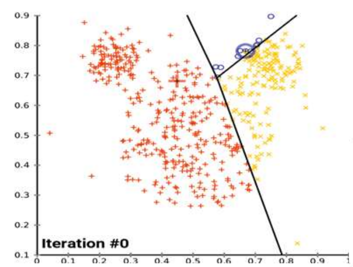
- Visualize Results
- Model Diagnostics
- Residual diagnositcs
- ROC curves
- etc

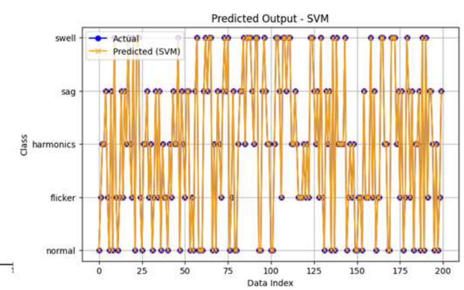
Present Results

- Charts
- Graphs
- Tables
- Visualizations to explain model, explain results



9	5000.0	5000.0	5000.0	-0.001698	0.449438	0.451136
10	5500.0	5500.0	5500.0	-0.001708	0.337079	0.338787
11	6000.0	6000.0	6000.0	-0.001701	0.404494	0.406195
12	6500.0	6500.0	6500.0	-0.001700	0.404494	0.406195
~~13~~	~~NaN~~	~~NaN~~	~~NaN~~	~~-0.001696~~	~~0.426966~~	~~0.428663~~
~~14~~	~~NaN~~	~~NaN~~	~~NaN~~	~~-0.001689~~	~~0.370787~~	~~0.372475~~

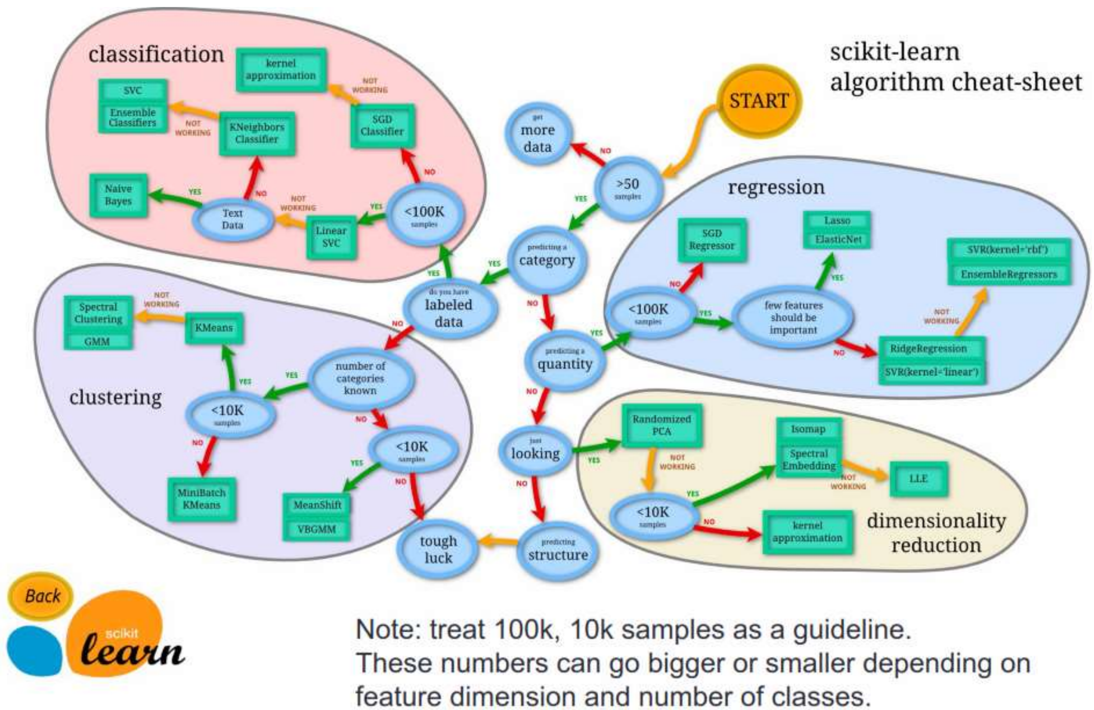




Machine Learning Algorithms

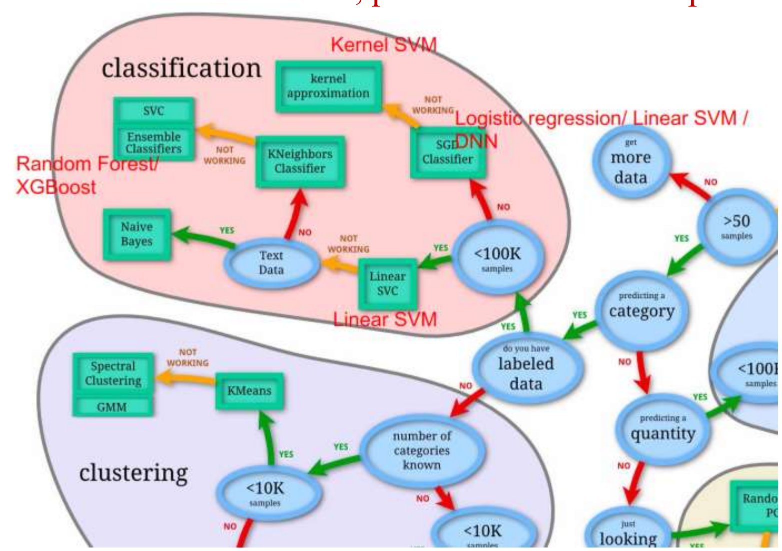
Types of machine learning

1. Supervised learning: Learn a model F from pairs of **(x,y)**
2. Unsupervised learning: Discover the hidden structure in unlabeled data **x (no y)**
3. Reinforcement learning: Train an agent to take appropriate actions in an environment by maximizing rewards.



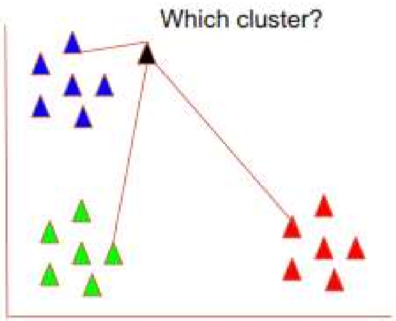
SVM: RBF kernel, tuned C & γ

k-NN: Euclidean distance, parameter k chosen experimentally

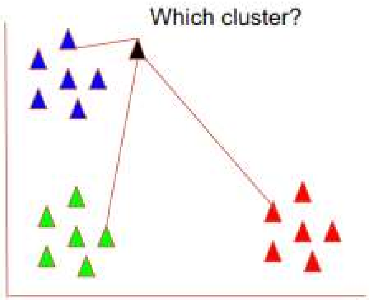


Nearest Neighbour classification

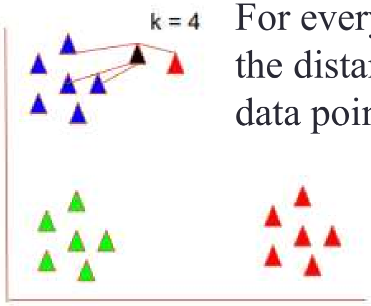
Find the closest training data, assign the same label as the training data



Given query data. For every point in the training data
Compute the distance with the query. Assign label of the smallest distance



Nearest Neighbour is susceptible to noise in the training data
Use a voting scheme instead



For every point in the training data. Compute the distance with the query. Find the K closest data points. Assign label by voting.

The votes can be weighted by the inverse distance (weighted k-NN).

Distance (Similarity) Measure

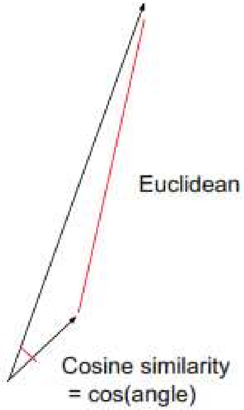
F(X_1 ,X_2) = d

$$X_1 = [x_{1,1}, x_{1,2}, \ldots, x_{1,n}]$$

$$X_2 = [x_{2,1}, x_{2,2}, \ldots, x_{2,n}]$$

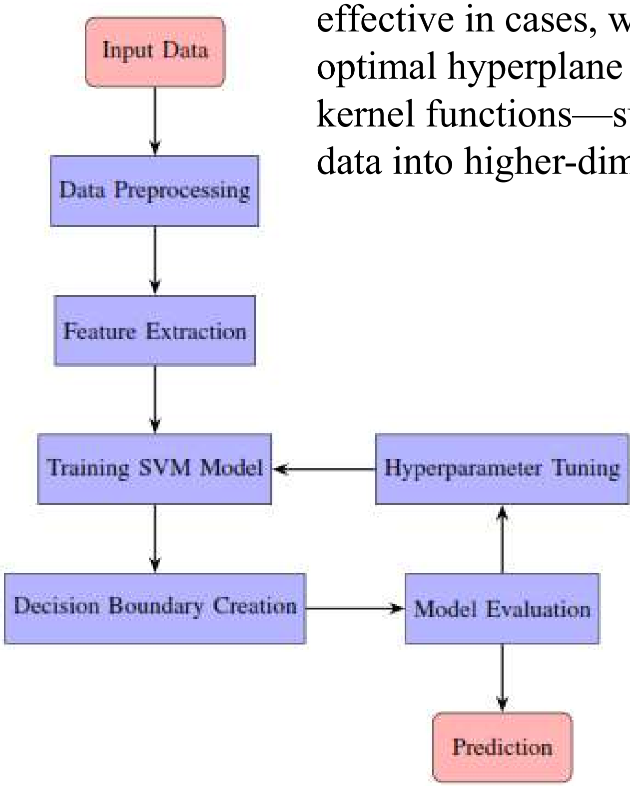
Euclidean distance $\sqrt{\Sigma_i (x_{1,i} - x_{2,i})^2}$

Cosine similarity $\frac{X_1 \cdot X_2}{|X_1||X_2|} = \frac{\Sigma_i x_{1,i} x_{2,i}}{\sqrt{\Sigma_i x_{1,i}^2}\sqrt{\Sigma_i x_{2,i}^2}}$



SVM Model Workflow

Common question: how do we ensure an ‘optimal’ hyperplane?

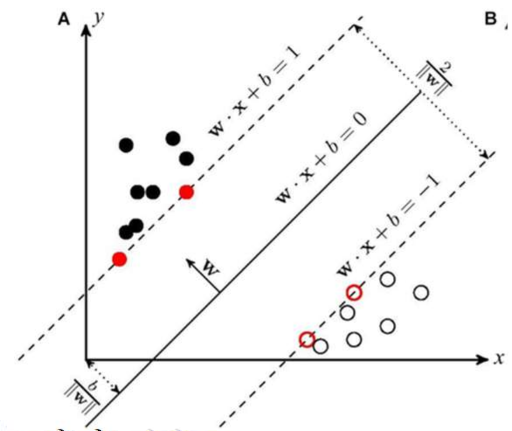


SVM are supervised machine learning algorithms widely applied in classification problems, particularly effective in cases, where data exhibit nonlinear separability. The core principle of SVM is to identify an optimal hyperplane that maximizes the margin between distinct classes within a feature space. By utilizing kernel functions—such as linear, polynomial, or radial basis function (RBF)—SVM can transform input data into higher-dimensional spaces, enabling the separation of complex, nonlinearly distributed patterns.

Goal: choose w such that r with

$$r = \frac{f(x, w)}{\|w\|}$$

gets maximal



But also ensure points are on correct side of decision boundary, i.e.

$$y_i\, f(x_i, w) - 1 \geq 0 \qquad \forall i$$

SVM in PQ

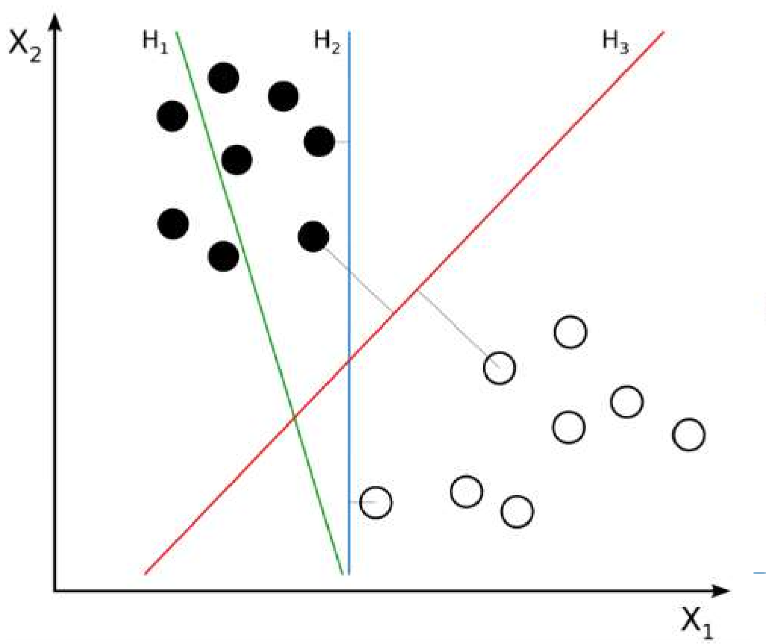
- Margin maximization principle
- Effective in nonlinear classification
- RBF kernel maps data into higher dimensions

Common question: how do we ensure an 'optimal' hyperplane?

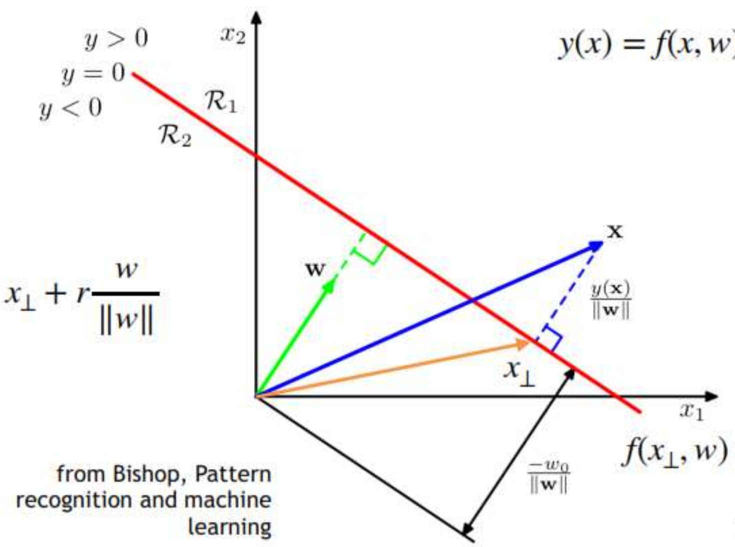
Primal problem: $\hat{w} = \arg\min_{w} \left\{ \sum_{i=1}^{s} \max\left(0, 1_i - (YXw)_i\right) + \frac{\alpha}{2}\|w\|^2 \right\}$

Dual problem: $\hat{\lambda} = \arg\max_{\lambda \in [0,1]^s} \left\{ \langle \lambda, 1 \rangle - \frac{1}{2\alpha}\|X^T Y \lambda\|^2 \right\}$

Connection between the two: $\hat{w} = \frac{1}{\alpha} X^T Y \hat{\lambda}$



$$x = x_\perp + r\frac{w}{\|w\|}$$



from Bishop, Pattern recognition and machine learning

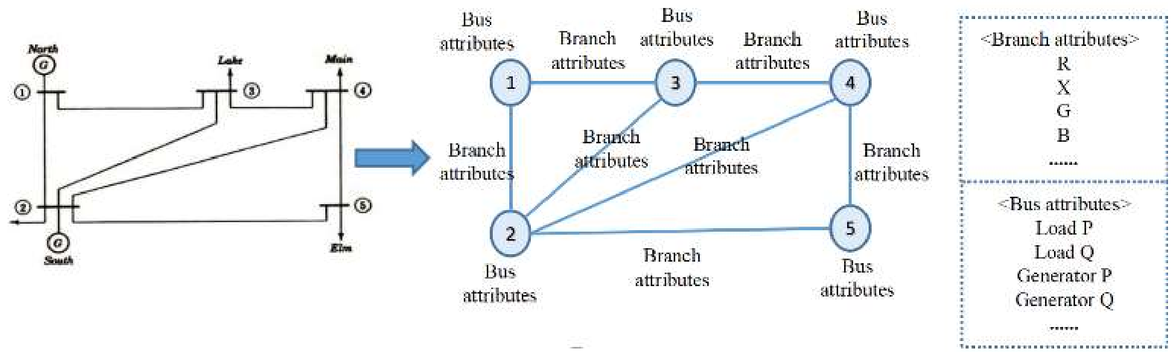
$$\begin{aligned} y(x) = f(x, w) &= w_0 + \langle w, x \rangle \\ &= w_0 + \langle x_\perp, w \rangle + r\frac{\langle w, w \rangle}{\|w\|} \\ &= w_0 + \langle x_\perp, w \rangle + r\|w\| \\ &= r\|w\| \end{aligned}$$

$$\Rightarrow \qquad r = \frac{f(x, w)}{\|w\|}$$

$$\begin{aligned} f(x_\perp, w) = y(x_\perp) &= 0 \\ &= w_0 + \langle w, x_\perp \rangle \end{aligned}$$

IEEE 5-Bus Test System

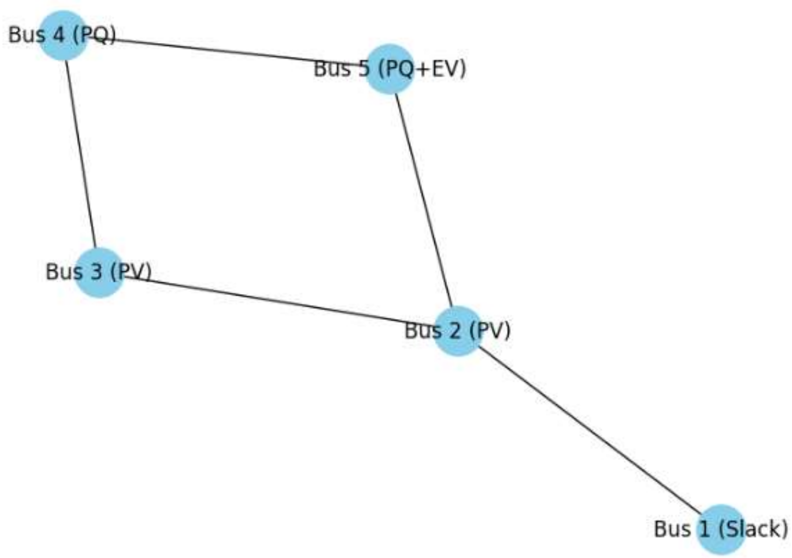
- 5-bus standard distribution model
- 1 slack bus, 2 PV buses, 2 PQ buses
- 100 MVA base system
- Widely used for stability & PQ studies



Bus No	Bus Voltage	Generation		Load	
		MW	MVar	MW	MVar
1	1.06+j0.0	0	0	0	0
2	1.0+j0.0	40	30	20	10
3	1.0+j0.0	0	0	45	15
4	1.0+j0.0	0	0	40	5
5	1.0+j0.0	0	0	60	10

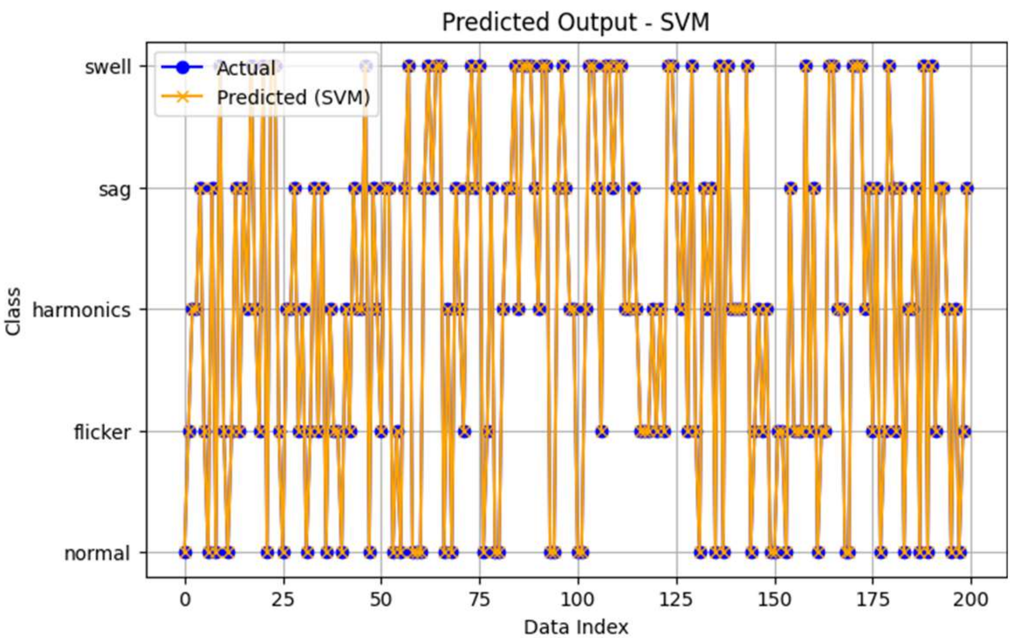
Line Data for IEEE 5-Bus System

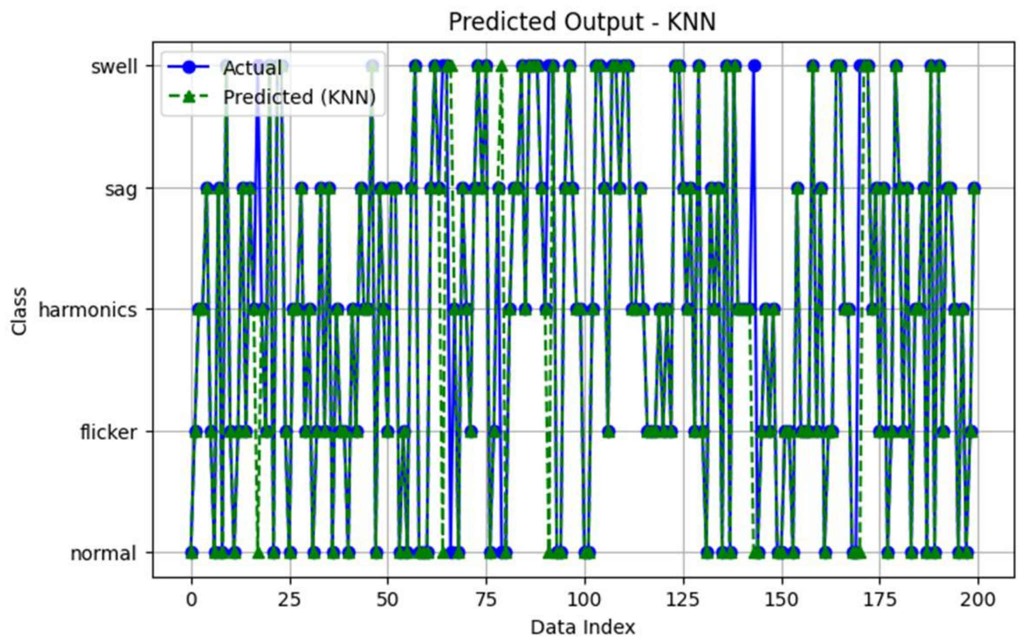
Line	Line Impedance		Line Charging
	R per unit	X per unit	
1-2	0.02	0.06	0.0+j0.03
1-3	0.08	0.24	0.0+j0.025
2-3	0.06	0.25	0.0+j0.02
2-4	0.06	0.18	0.0+j0.02
2-5	0.04	0.12	0.0+j0.015
3-4	0.01	0.03	0.0+j0.01
4-5	0.08	0.24	0.0+j0.025



Build an IEEE-style 5-Bus Distribution Network (pandapower)

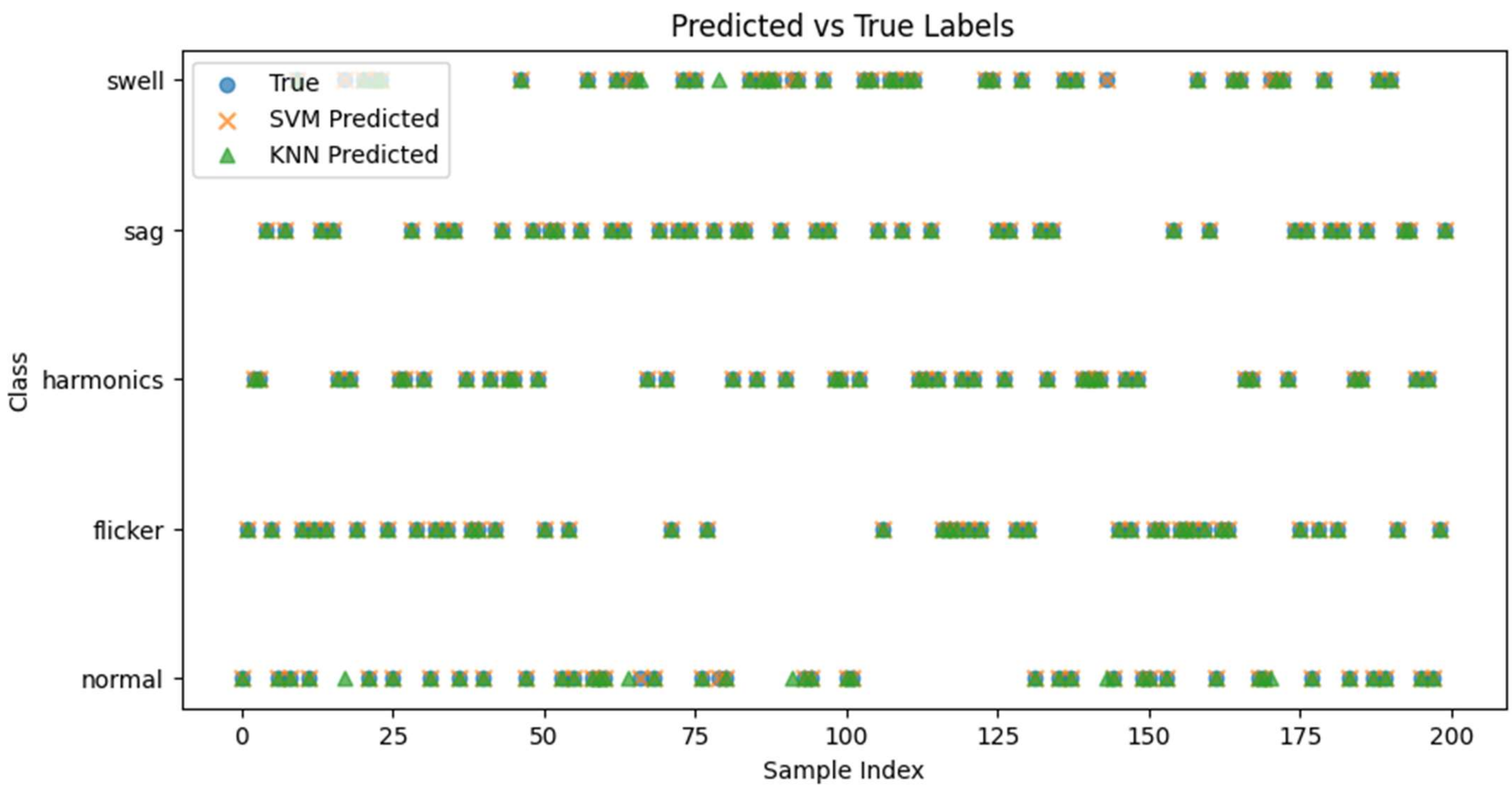
Classification Results





	vm_ev	vm_min	vm_max	vm_mean	line_loading_max_pct	p_total_mw	q_total_mvar	system_pf	freq_hz	df_hz	thd_pct	flicker_idx	event_duration_s	label
0	0.993962	0.991895	1.01	1.001171	62.085193	-0.987342	2.839183	0.328461	50.006094	0.006094	3.729483	0.000000	0.237523	normal
1	0.992104	0.990650	1.01	1.000551	47.437267	-0.731591	2.902105	0.244442	50.018811	0.018811	4.761080	0.000000	0.134891	sag
2	0.994271	0.992100	1.01	1.001274	63.620001	-1.017433	2.831943	0.338111	50.002557	0.002557	3.554429	0.000000	0.199160	swell
3	0.993834	0.991809	1.01	1.001129	61.928643	-0.985493	2.839427	0.327887	49.982939	-0.017061	12.214950	0.000000	0.238890	harmonics
4	0.993480	0.991572	1.01	1.001010	58.563867	-0.919306	2.855943	0.306409	49.996303	-0.003697	3.745688	0.001583	1.244508	flicker

Classification Results



Results & Discussion

- Train/test split with labeled dataset
- Metrics: Accuracy (±3),
- Precision, Recall, F1-score
- Confusion matrix (7 classes)

- SVM outperforms k-NN consistently
- Higher accuracy & precision
- More balanced performance across classes

METRIC	SVM	KNN
Accuracy (±3)	0.94	0.88
F1 Score	0.54	0.46
Precision	0.71	0.49
Recall	0.59	0.53
Confusion Matrix	[[2, 0, 0, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 2, 1], [0, 0, 0, 2, 0, 0, 0], [0, 0, 0, 1, 4, 0, 0], [0, 0, 0, 0, 2, 0, 0], [0, 0, 0, 0, 0, 0, 1]]	[[2, 0, 0, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 0], [0, 0, 2, 0, 1, 0, 0], [1, 0, 1, 0, 0, 0, 0], [1, 2, 1, 0, 0, 0, 0], [1, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1]]

The comparison of **SVM** and **KNN** for power quality classification. **SVM** delivers consistently high accuracy and balanced performance, while **KNN** achieves competitive results on imbalanced data. Overall, SVM is more reliable, but KNN offers adaptability depending on task needs.

Key Insights, Practical Applications

- SVM handles nonlinear PQ data better
- Robust to noise & high-dimensional features
- k-NN limited by sensitivity to scaling
- Integrate into PQ monitoring systems
- Real-time classification at utility substations
- Triggers automated mitigation (filters, DSM, protection settings)
- EV charging stresses PQ → urgent need for ML tools
- SVM = reliable, accurate classifier for PQ events
- k-NN = simple but less robust
- Simulation-driven ML offers scalable path to grid resilience

https://github.com/Nick-Panaya/PQSynergy.git

Limitations & Future Work

- Small-scale test system (IEEE 5-Bus)
- Limited detail on feature extraction & sampling
- Future: real-world datasets, hybrid models, online learning