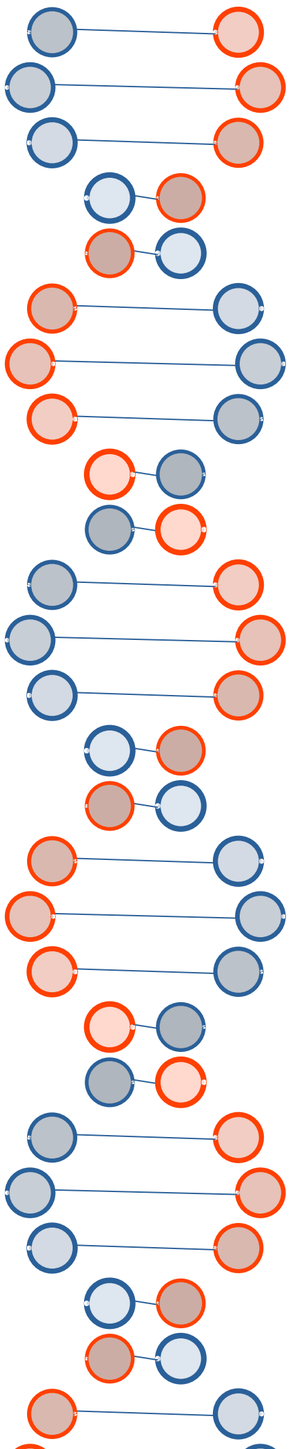


Electrical Substation Failure Prediction Using 1D-CNNs

Anand Rinchinjugder

Anderson College of Business and Computing
Regis University

Motivation

Electrical substation devices are all not reliable
Dispatch teams are not always ready to fix sites
Latest research focuses on classical ML models

Can we predict electrical substation failures in advance using multi-variable, multi-device, real world data?

Research Problem

Binary classification: Predict substation failure events in advance, given earlier days of data.

Let,

1. $i \in \{1,\ldots,N\}$ = index of measurement devices.
2. $t \in \mathbb{Z}$ = integer time steps (hourly).
3. $x_{i,t} \in \mathbb{R}^f$ = measurement device logs (the feature).
4. $T_{k,i}$ = timestamp of the kth failure for device i.

Define,

1. sliding window of length $W = \Delta T = [x_{i,t-W+1},\ldots,x_{i,t}]$,
2. target $y_{i,t} = 1$, if there exists a failure k. 0, otherwise.

Data Sources

Power Communications Monitor device (PCM). The device captures real-time and event-driven measurements at electrical substations

Data Sources

- The device records signal strength, margin to alarm, and standing wave ratio of each channel which are critical metrics at a electrical substation
- PCM failure means no substation data
- Electrical engineers analyze data from PCMs and deploy field engineers to substations to fix issues

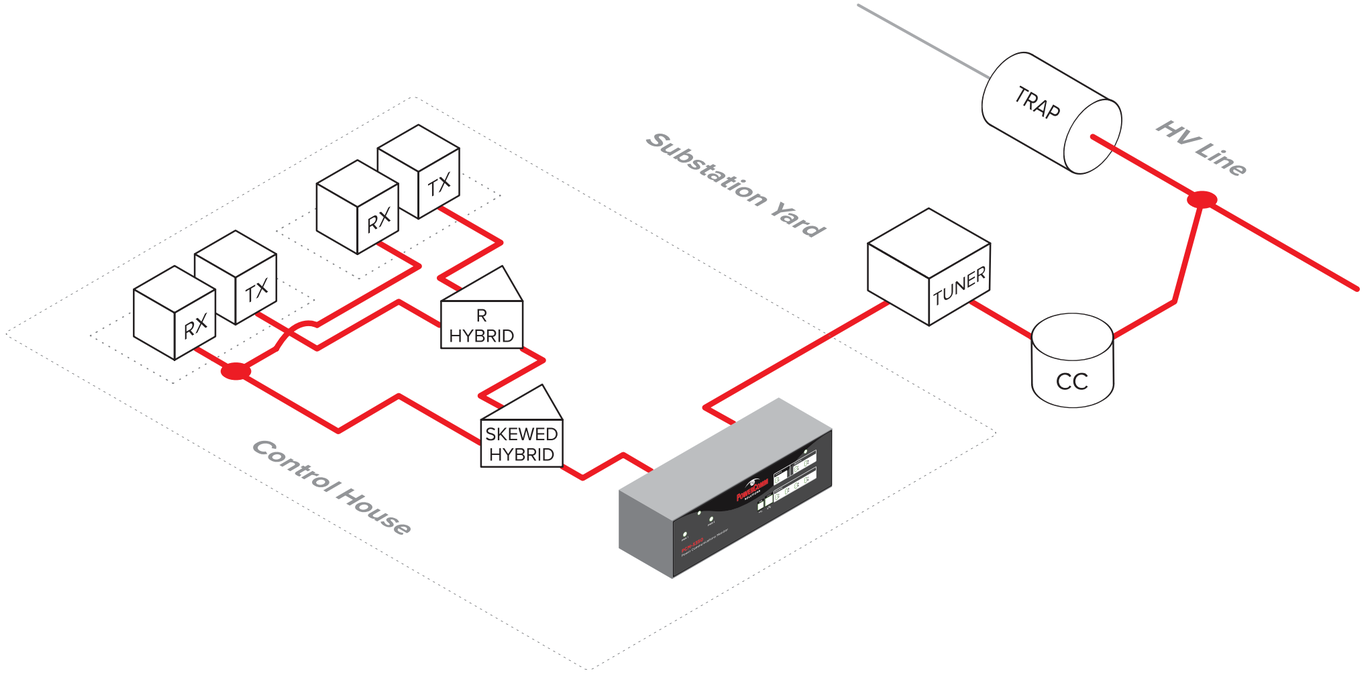


fig source: https://powercommsolutions.com/products/5350

Feature Selection/Engineering

- 12 Features: signal level, margin level, SWR for multiple channels

- Sliding Window Strategy
  - A 1D CNN needs a fixed length input window size
  - 24, 168, 336, 720, 2160, 4380 hour window sizes × 12 features

- 24, 168, 720, 2160, 4380, 8760 hour lead time windows.
  - Label is attached to the end of each window
  - Earlier time gives operators time to dispatch a maintenance team

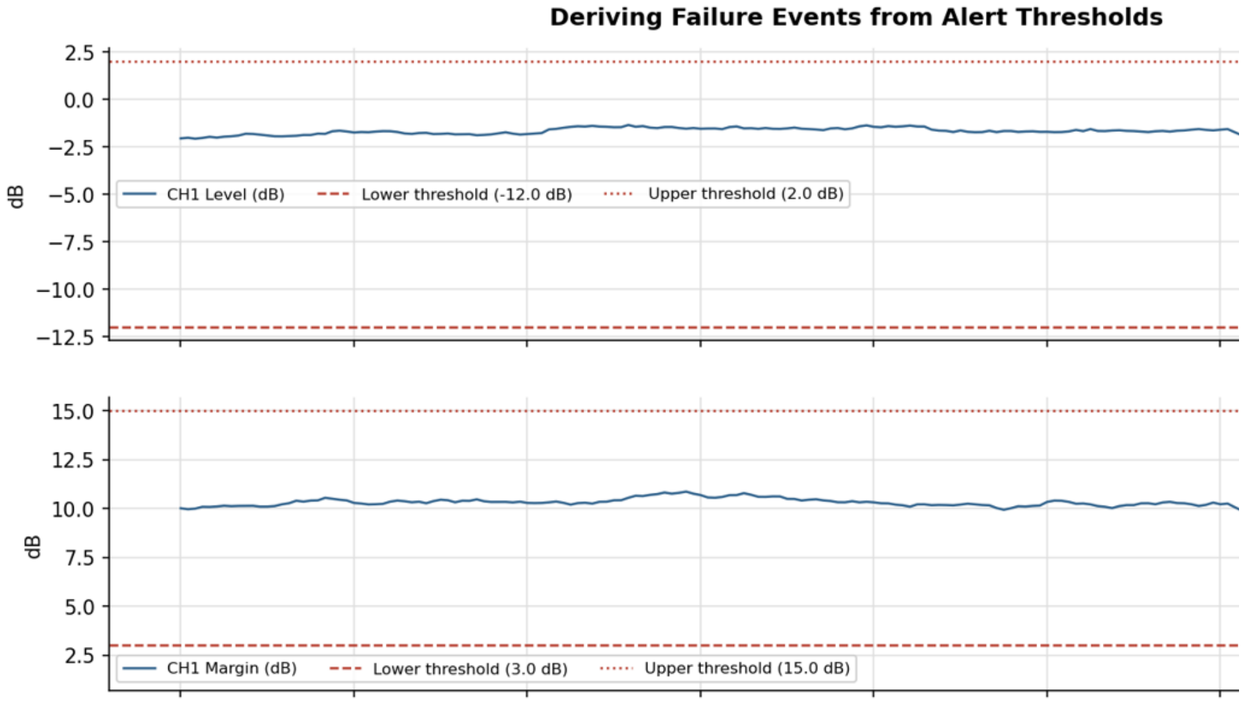
- Per PCM normalization
  - Per PCM RobustScaler fitted on training data, then applied to val/test

- Why not ML based?
  - Used domain knowledge. Including any other measurements would add noise, not signal

Target Selection

1. The database has no "this was a failure" column, BUT
   - The database has failure thresholds for each device
   - thrsh_alert_lower = minimum acceptable value
   - thrsh_alert_upper = maximum acceptable value

1. is_failure = (value < thrsh_alert_lower) OR (value > thrsh_alert_upper)



EDA

PCM Raw Measurement Logs Table

SQL database

25.5 million rows of data across 268 PCMs extracted from 2018 to 2025

Timestamps are stored as strings (20230809T122733), so datetime parsing needed

PCMs measurement logs are recorded at irregular times

Results | Messages

	id	pcm_id	timestamp	value	meta	pk_index
1	4	6	20230809T132738	35		1
2	2	5	20230809T142744	0		2
3	0	3	20230809T152747	41		3
4	29	342	20230809T162837	46		4
5	4	1	20230809T182800	30		5
6	23	4	20230809T192806	0		6
7	13	344	20230809T202812	0		7
8	10	9	20230809T212816	1		8
9	8	344	20230809T222819	19		9
10	5	341	20230809T232912	-39		10
11	3	3	20230810T002920	40		11
12	5	1	20230810T030217	19		12
13	27	5	20230810T040306	0		13
14	8	343	20230810T060235	21		14
15	8	344	20230810T080242	19		15
16	31	248	20230810T090335	11		16
17	22	4	20230810T110304	220597		17
18	20	3	20230810T120310	102248		18
19	8	8	20230810T130317	22		19

Query executed successfully.

EDA

	id	name	detail	pcm_id	thrsh_alarm_lower	thrsh_alarm_upper	thrsh_alert_lower	thrsh_alert_upper	thrsh_db_alarm_lower	thrsh_db_alarm_upper	thrsh_db_alert_lower	thrsh_db_alert_upper	log_interval
1	0	Wideband Level dBm	Wideband Level in dBm	6	15	44	20	43	60	-1	65	-2	3600
2	1	Wideband Level Vrms	Wideband Level in Vrms	6	-120	120	10	110	-115	115	15	105	3600
3	2	Wideband Current	Wideband Current in Arms	6	0	0	0	0	0	0	0	0	3600
4	3	Level dBm CH 1	Sec Phase 3 DCUB Tx 236 KHz - Level in dBm	6	34.25	44.25	36.25	42.25	34.25	44.25	36.25	42.25	3600
5	4	Level dBm CH 2	Sec Phase 3 DCUB Rx 231 KHz - Level in dBm	6	15.92	40.92	25.92	38.92	15.92	40.92	25.92	38.92	3600
6	5	Level dBm CH 3	Other Phase 1 Tx - Level in dBm	6	17.34	27.34	19.34	25.34	17.34	27.34	19.34	25.34	3600
7	6	Level dBm CH 4	Channel 4 - Level in dBm	6	30	35	30	33	30	35	30	33	3600
8	7	Level dBm CH 5	Other Phase 1 Rx - Level in dBm	6	7.08	32.08	17.08	30.08	7.08	32.08	17.08	30.08	3600
9	8	Level Vrms CH 1	Channel 1 Level in Vrms	6	100	120	-100	-120	95	115	-95	-115	3600
10	9	Level Vrms CH 2	Channel 2 Level in Vrms	6	100	120	-100	-120	95	115	-95	-115	3600
11	10	Level Vrms CH 3	Channel 3 Level in Vrms	6	100	120	-100	-120	95	115	-95	-115	3600
12	11	Level Vrms CH 4	Channel 4 Level in Vrms	6	100	120	-100	-120	95	115	-95	-115	3600
13	12	Level Vrms CH 5	Channel 5 Level in Vrms	6	100	120	-100	-120	95	115	-95	-115	3600
14	13	Margin Level CH 1	Sec Phase 3 DCUB Tx 236 KHz - Level Diff from PC...	6	-5	5	-3	3	-5	5	-3	3	3600
15	14	Margin Level CH 2	Sec Phase 3 DCUB Rx 231 KHz - Level Diff from PC...	6	-20	5	-10	3	-20	5	-10	3	3600
16	15	Margin Level CH 3	Other Phase 1 Tx - Level Diff from PCM Nominal	6	-5	5	-3	3	-5	5	-3	3	3600
17	16	Margin Level CH 4	Channel 4 - Level Diff from PCM Nominal	6	-5	5	-3	3	-5	5	-3	3	3600
18	17	Margin Level CH 5	Other Phase 1 Rx - Level Diff from PCM Nominal	6	-20	5	-10	3	-20	5	-10	3	3600

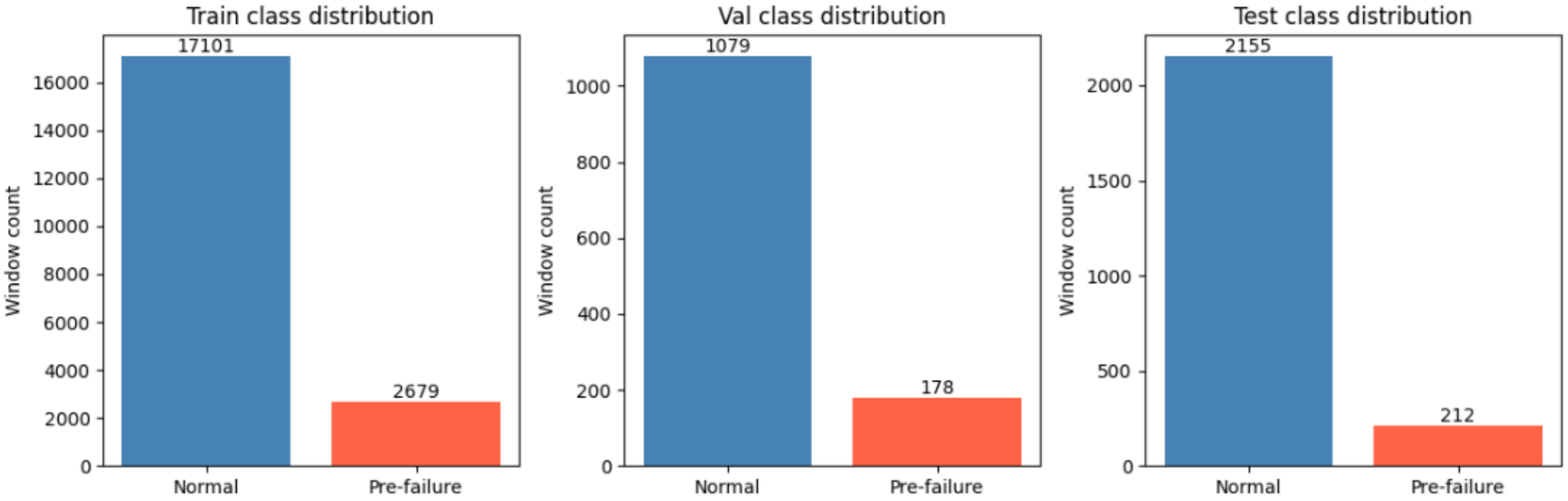
PCM Metadata Table

- PCM metadata table and the raw log table are separate. Need to join the tables
- Each PCM has multiple ids indicating different measurements

EDA

- Summary statistics
- Checking NaN values
- Checking for outliers

Positive Class Imbalance



Data Cleaning and Preparation

Timestamp parsing, id mapping
Handling NaNs
- CH4, CH5 NaN for most PCMs dropped (>95% threshold)

Outlier clipping
- Drop values higher/lower than 3 std

pivoting from long format to wide format data
Irregular timestamps
- resampling to 1 hour grid

Raw Data Format — Before and After Parsing

Raw (opaque ids, string timestamps)

pk_index	pcm_id	id	value	timestamp
1001	384	3	-2.1	20210304T083212
1002	384	13	8.4	20210304T083212
1003	384	27	1.31	20210304T083212
1004	341	3	-5.3	20210304T083456
1005	341	4	-4.8	20210304T083456

After parsing & ID mapping

pcm_id	datetime	measurement_id	value
384	2021-03-04 08:32	CH1_Level_dB	-2.1
384	2021-03-04 08:32	CH1_Margin_dB	8.4
384	2021-03-04 08:32	CH1_SWR_points	1.31
341	2021-03-04 08:34	CH1_Level_dB	-5.3
341	2021-03-04 08:34	CH2_Level_dB	-4.8

Models Architecture

Simple 1D-CNN for detecting local patterns

- 3 conv stages with BatchNorm and MaxPool
- 38k parameters

Resnet1D for deeper hierarchies

- 3 Residual blocks
- Skip Connections
- 221K parameters

TCN for exponential causal context

- 8 TemporalBlocks, dilations (1, 2, 4, 8, 16, 32, 64, 128)
- Only past context, no future leakage
- Weight normalisation (not batch norm)
- 287K parameters

Model Training

1. Loss = BCEWithLogitsLoss
   - pos_weight=1.0
   - WeightedRandomSampler
2. Optimizer = AdamW
3. Schedule = OneCycleLR
4. Early stopping
   - patience=10 on val AUC-PR

Result

Model	Best LR	Best Channels	Best Dropout	AUC-PR
Simple 1D-CNN	0.001	32	0.3	0.81
Resnet1D	0.0005	32	0.3	0.79
TCN	0.001	32	0.4	0.74

Baseline Comparison

- **Simple 1D-CNN performs best on AUC-PR than traditional ML models**
- **XGBoost performs the best on AUC-ROC**
- **All models performs similarly on AUC-ROC**

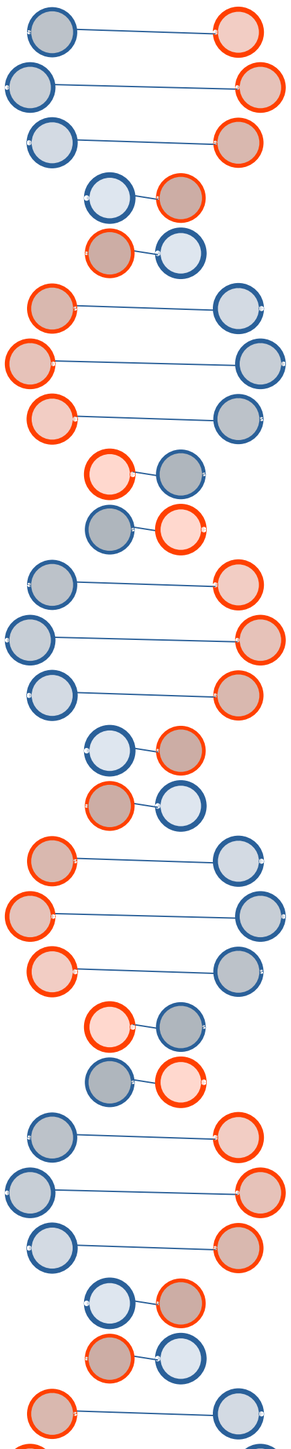
Model	AUC-PR	AUC-ROC
Simple 1D-CNN	0.81	0.80
XGBoost	0.67	0.83
Logistic Regression	0.66	0.80
Random Forest	0.63	0.81

Findings

1. **Simple 1-D CNN performs better than ResNet1D and Temporal CNN.**

1. **1D CNN performs similar to classical ML models on AUC evals**

1. **Local patterns outperform full window TCN. Maybe Substation failures are sudden bursts not, gradual trends**



Tools

pyodbc + SQL
pandas + numpy
sklearn
PyTorch
matplotlib + seaborn