Detection of Load Redistribution Attacks: A Data-Driven Approach

Andrea Pinceti

Committee Members:
Dr. Lalitha Sankar (Chair)
Dr. Oliver Kosut
Dr. Anamitra Pal
Dr. Yang Weng

April 11th, 2019

Overview

- Introduction
- Detection of load redistribution attacks
    - Algorithms
    - Small scale systems
    - Large scale systems
- Conclusion

Introduction

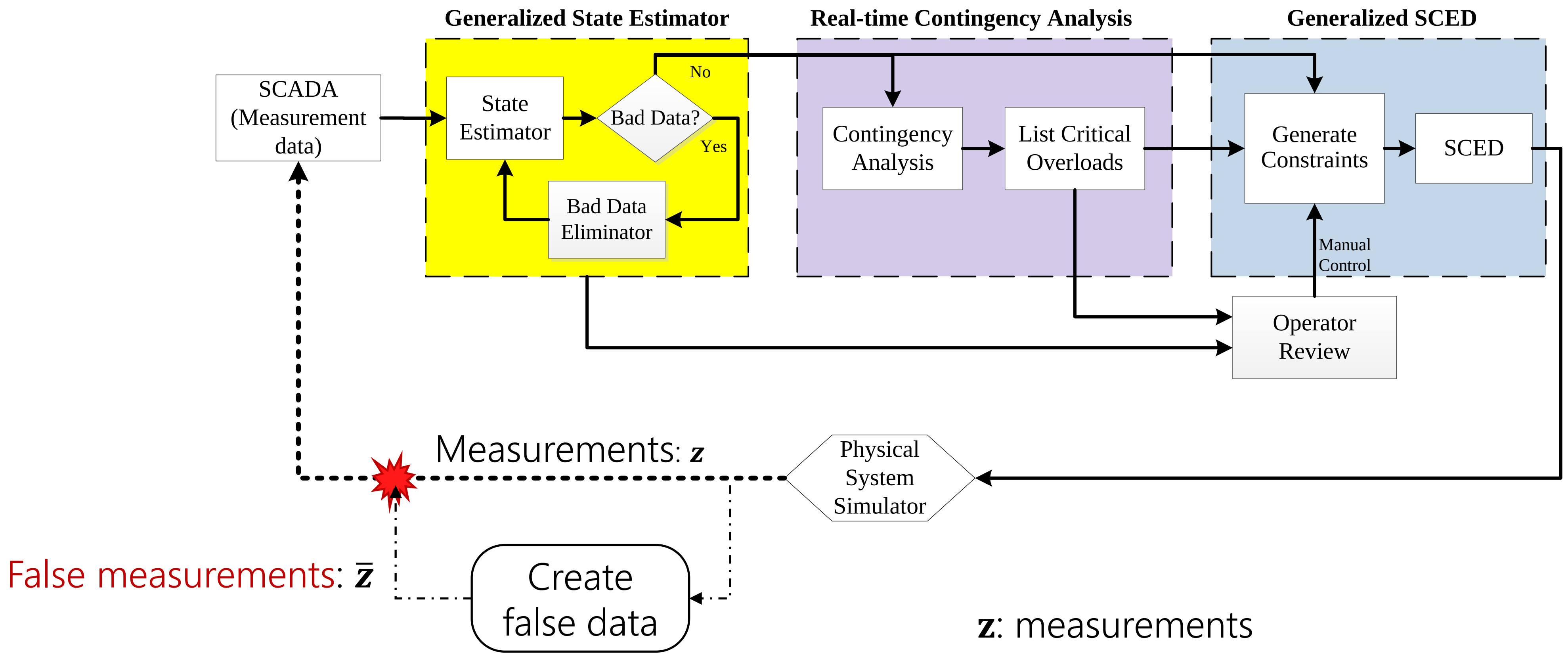
The electrical grid is a cyber-physical system (CPS) → Cyber vulnerabilities can have serious physical consequences

- 2015 Ukraine cyber-attack [1]
  - First sophisticated attack on a power grid
  - Months of preparation and study of their target
  - 30 substations were "switched off"
- 2018 *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors* [2]
  - Joint DHS and FBI investigation
  - Russian cyber-attacks have infiltrated the US electrical grid
  - Data from SCADA and control systems accessed by the attackers

What kind of sophisticated attacks are possible?

[1] Wired, "*Inside the cunning, unprecedented hack of Ukraine's power grid*", *March 2016*, https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
[2] Department of Homeland Security, Alert (TA18-074A), "*Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors*", *March 2018*

False data injection (FDI) attacks on state estimation (SE)



State estimation: $\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{e}$

Unobservable attack [1] : $\bar{\boldsymbol{z}} = \boldsymbol{z} + H\boldsymbol{c}$

$\mathbf{z}$: measurements
$\boldsymbol{x}$: system states
H: relationship between states and measurements
$\boldsymbol{e}$: random noise
$\boldsymbol{c}$: arbitrary attack vector

[1] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09, Chicago, Illinois, USA, 2009, pp. 21–32.

False data injection (FDI) attacks on state estimation (SE)

Attacks on state estimation

- Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS ’09, Chicago, Illinois, USA, 2009, pp. 21–32.
- O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on the smart grid,” IEEE Transactions on Smart Grid, vol. 2, no. 4, pp. 645–658, 2011

Line overflow attacks

- J. Zhang and L. Sankar, “Physical system consequences of unobservable state-and-topology cyber-physical attacks,” IEEE Transactions on Smart Grid, vol. 7, no. 4, pp. 2016–2025, July 2016.
- J. Liang, L. Sankar, and O. Kosut, “Vulnerability analysis and consequences of false data injection attack on power system state estimation,” IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3864–3872, Sept 2016.

Market attacks

- R. Moslemi, A. Mesbahi, and J. M. Velni, “Design of robust profitable false data injection attacks in multi-settlement electricity markets,” IET Generation, Transmission Distribution, vol. 12, no. 6, pp. 1263–1270, 2018.
- L. Jia, J. Kim, R. J. Thomas, and L. Tong, “Impact of data quality on real-time locational marginal price,” IEEE Trans. Power Systems, vol. 29, no. 2, pp. 627–636, 2014.

Need for Secure Data-Driven Bad Data Detectors

The FDI attacks described are part of a broad class of cyber threats:

LOAD REDISTRIBUTION ATTACKS

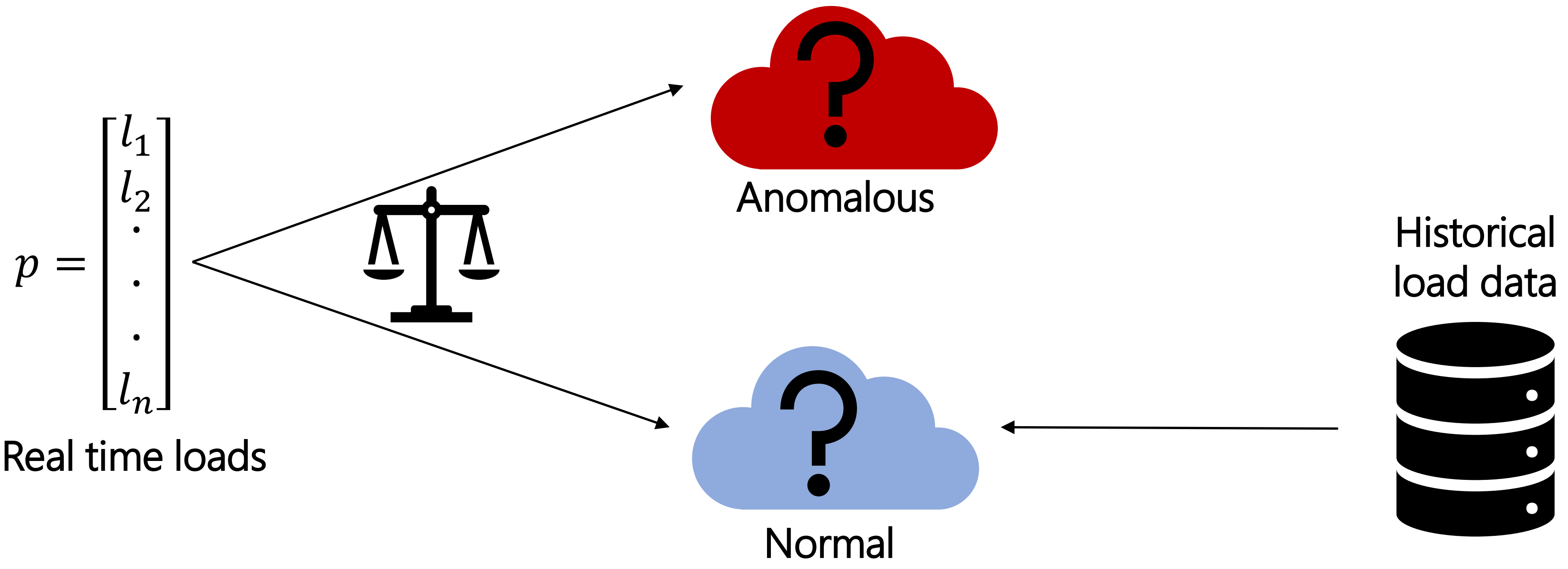
Traditional bad data detection (BDD) schemes can be bypassed

- SE-BDD tuned to identify and correct noisy measurements, not intelligently designed false data
- Load estimators may flag unusually large loads, not slightly modified groups of loads

Smarter detection algorithms that can identify such attacks and other anomalies are needed

Attack detection approach

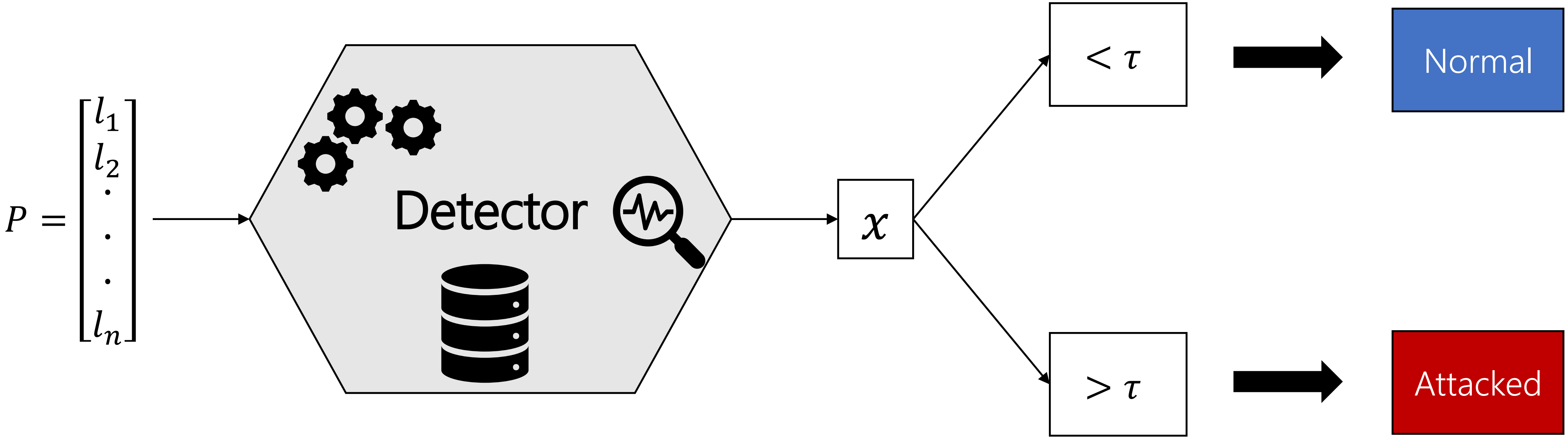
- Three detectors to analyze the measured loads of a power system
- Leverage the large amount of historical data available to operators for real time decision making
- Use machine learning algorithms to analyze the observed loads and determine if they are normal or if they have been maliciously modified



Attack detection approach

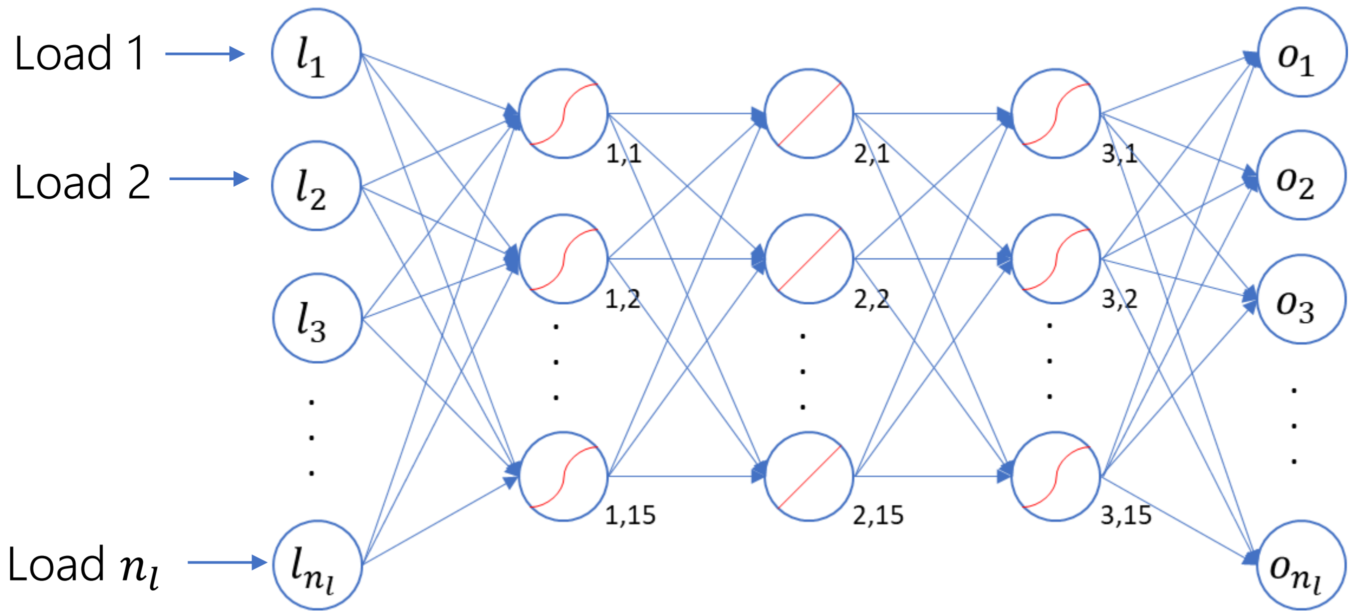
The approach consists of:

- Feeding the estimated load vector P as input to the detector
- The detector generates a **scalar** value x based on a metric specific to the machine learning technique used
- This value is compared against a predetermined threshold τ to label the loads as normal or attacked



Replicator Neural Network

- Compress and reconstruct the input data
- Same number of output neurons as the number of inputs
- Three hidden layers with 15 neurons each
- Layers 1 and 3: sigmoid activation Layer 2: linear activation function

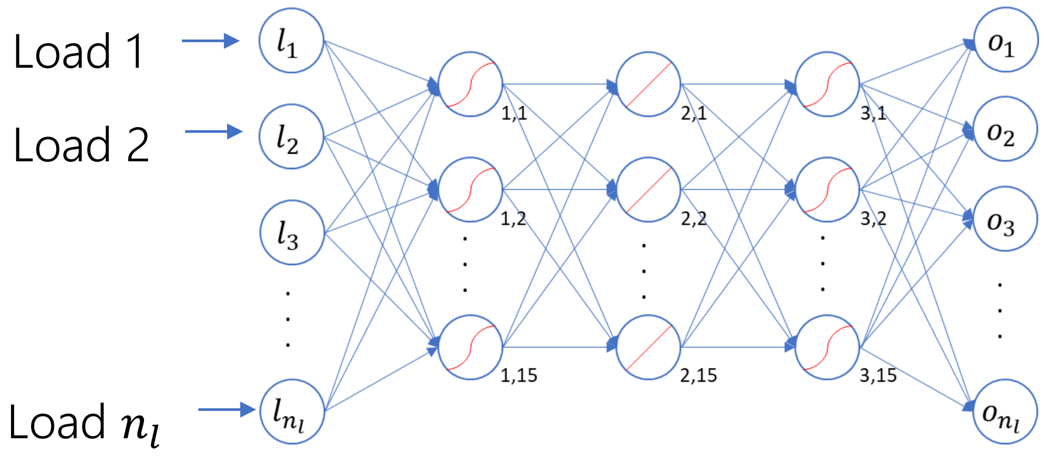


Replicator Neural Network

- Intuitively, the neural network learns a model of the correlation between the system loads
- Real load vectors will follow the learnt model and have small replication error
- A load vector resulting from a load redistribution attack will yield a big replication error
- The detection is performed by
    - measuring replication error δ_i for the measured load vector $\boldsymbol{l_i}$

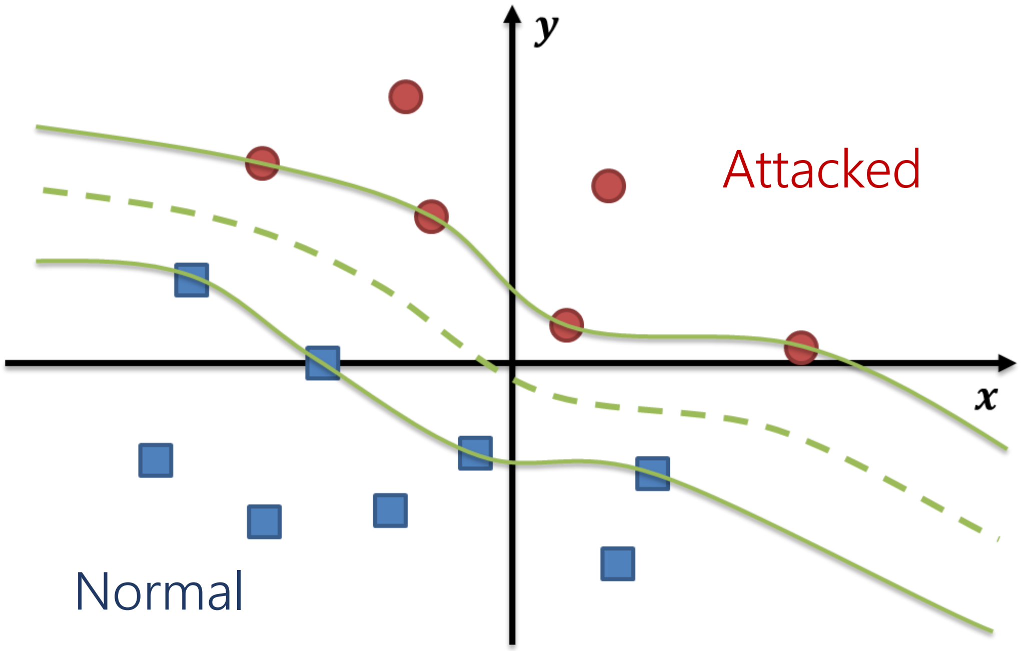
$$\delta_{\mathrm{i}} = ||\mathbf{l_i} - \mathbf{o_i}||_2$$

    - if δ_i is greater than a given threshold the loads are labelled as attacked



Support Vector Machine

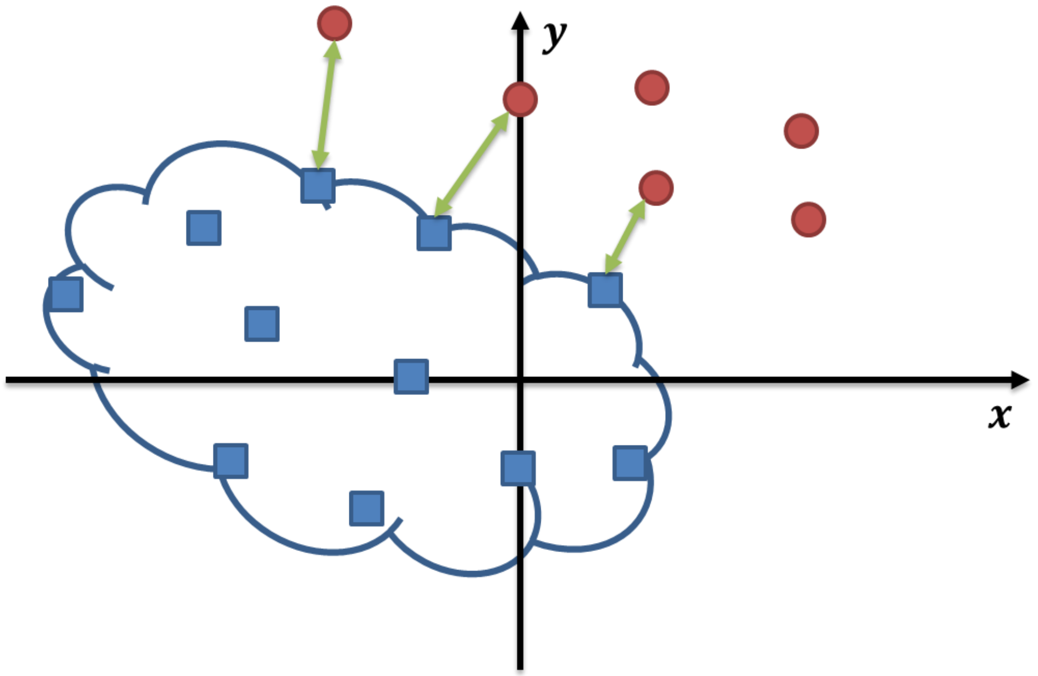
- Training determines a complex hyperplane containing all normal load profiles
- Testing of a sample returns a *confidence score* (normalized distance)
- Confidence score between -1 and +1 (attacked and normal respectively)
- The confidence score is compared to the threshold



Nearest Neighbor

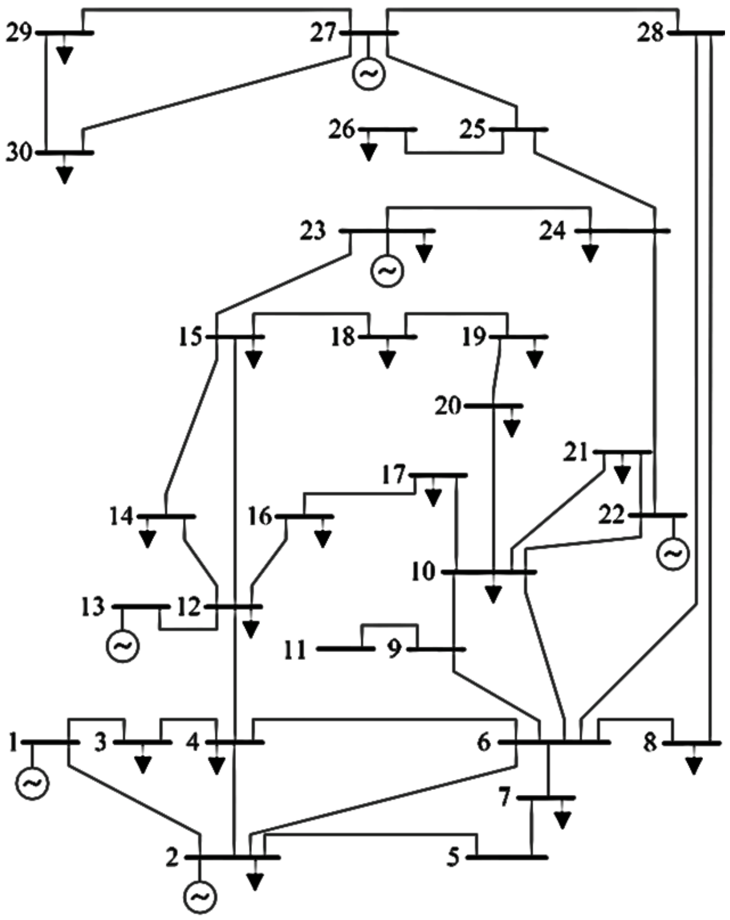
- Normal data lies in limited, dense regions of space
- Anomalies are located further from these neighborhoods
- Search the historical data for a load pattern close (or similar) to the observed loads
- The minimum distance d_i between observed loads $\boldsymbol{l_i}$ and the n_h historical loads $\boldsymbol{h_j}$ is used as detection metric

$$d_{\mathrm{i}} = \min_{j=[1:n_h]} ||\boldsymbol{l_i} - \boldsymbol{h_j}||_2$$



Load Data

- The three detectors are tested on the IEEE 30 bus system
- Historical load data for each of the 20 system loads is generated using real data:
  - PJM publishes hourly load data for each of the 20+ zones [a];
  - Data from 2012 to 2016 is mapped and scaled to each of the 20 loads in the IEEE 30-bus system
  - The first 4 years are used for training and the last one for testing



[a] PJM. (11 Nov 2017). PJM Metered Load Data [Online]. Available: https://www.pjm.com/markets-and-operations/ops-analysis/historical-load-data.aspx

Load Data

- The PJM zones and the loads of the 30 bus system are ranked and matched based on their relative size (Table I)
- The PJM data is scaled to the 30 bus system based on the following mapping ratio:

$$m_{\text{ratio}} = \frac{\text{30 bus net load}}{\text{max PJM net load}} \times k = \frac{189.2\ \text{MW}}{144644\ \text{MW}} \times 1.39$$

k is a constant chosen to obtain a congested system

TABLE I
RELATIVE SIZE OF PJM ZONES AND 30-BUS SYSTEM LOADS

PJM zone		30-bus system load	
Zone name	**Size [%]**	**Bus location**	**size [%]**
AEP	14.5	**8**	15.9
CE	13.7	**7**	12.1
DOM	12.4	**2**	11.5
ATSI	8.25	**21**	9.25
PS	6.34	**12**	5.92
AP	5.59	**30**	5.60
PE	5.41	**19**	5.02
PL	4.51	**17**	4.76
BC	4.27	**24**	4.60
PEP	4.08	**15**	4.33
JC	3.85	**4**	4.02
DEOK	3.35	**14**	3.28
DPL	2.60	**10**	3.07
DAY	2.14	**16**	1.85
ME	1.89	**26**	1.85
PN	1.88	**18**	1.69
DUQ	1.81	**23**	1.69
AE	1.73	**3**	1.27
EKPC	1.46	**29**	1.27
RECO	0.26	**20**	1.16

Attacked Data

- Attacks designed via bi-level attack optimization problem
- Attack consequences: line overflow
- First level models attacker's actions
- Second level models system's response to the attack
- Two values of load shift are considered: 10% and 15%

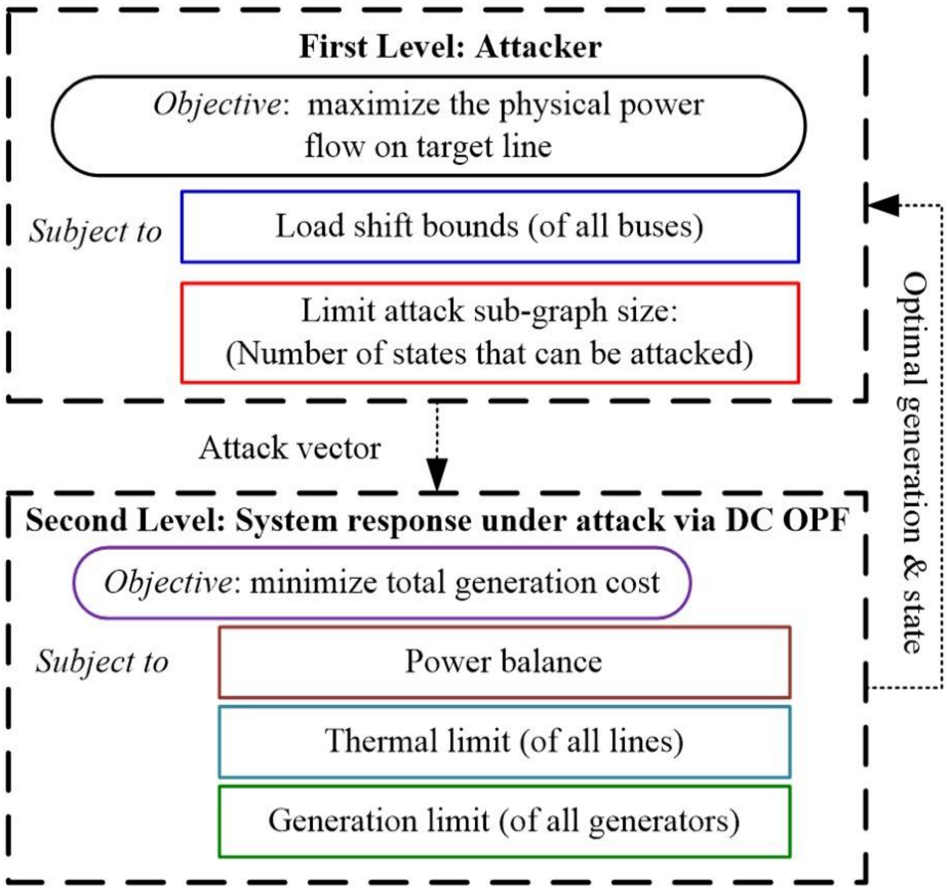


Figure courtesy of Mr. Zhigang Chu

Attacked Data

- DCOPF run on every hour of 2016
  - 1197 hours out of 8784 have at least one congested line (above 85% loading)
  - 450 hours out of 1197 have at least one line at 100%
- Successful attack: attack that leads to one ore more lines exceeding their long-term rating
- Attacks are attempted on the congested hours:
  - $\text{LS} = 10\%$: 437 successful attacks
  - $\text{LS} = 15\%$: 479 successful attacks
- The loads resulting from the attacks are used to test the detectors

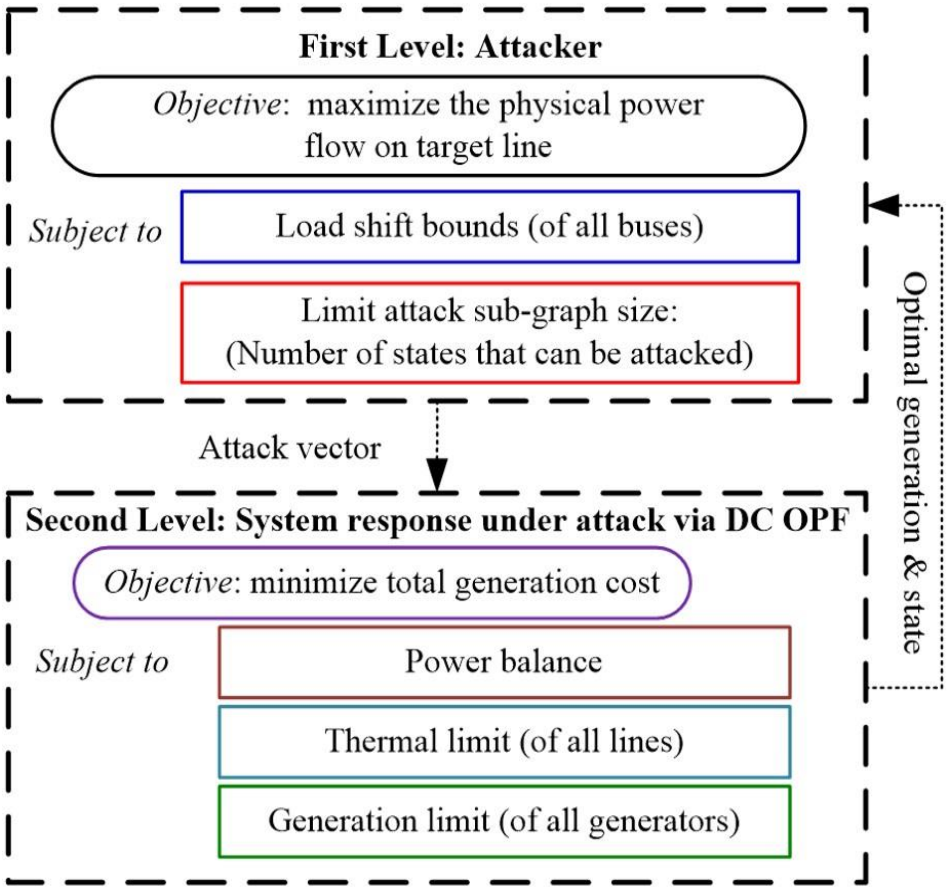


Figure courtesy of Mr. Zhigang Chu

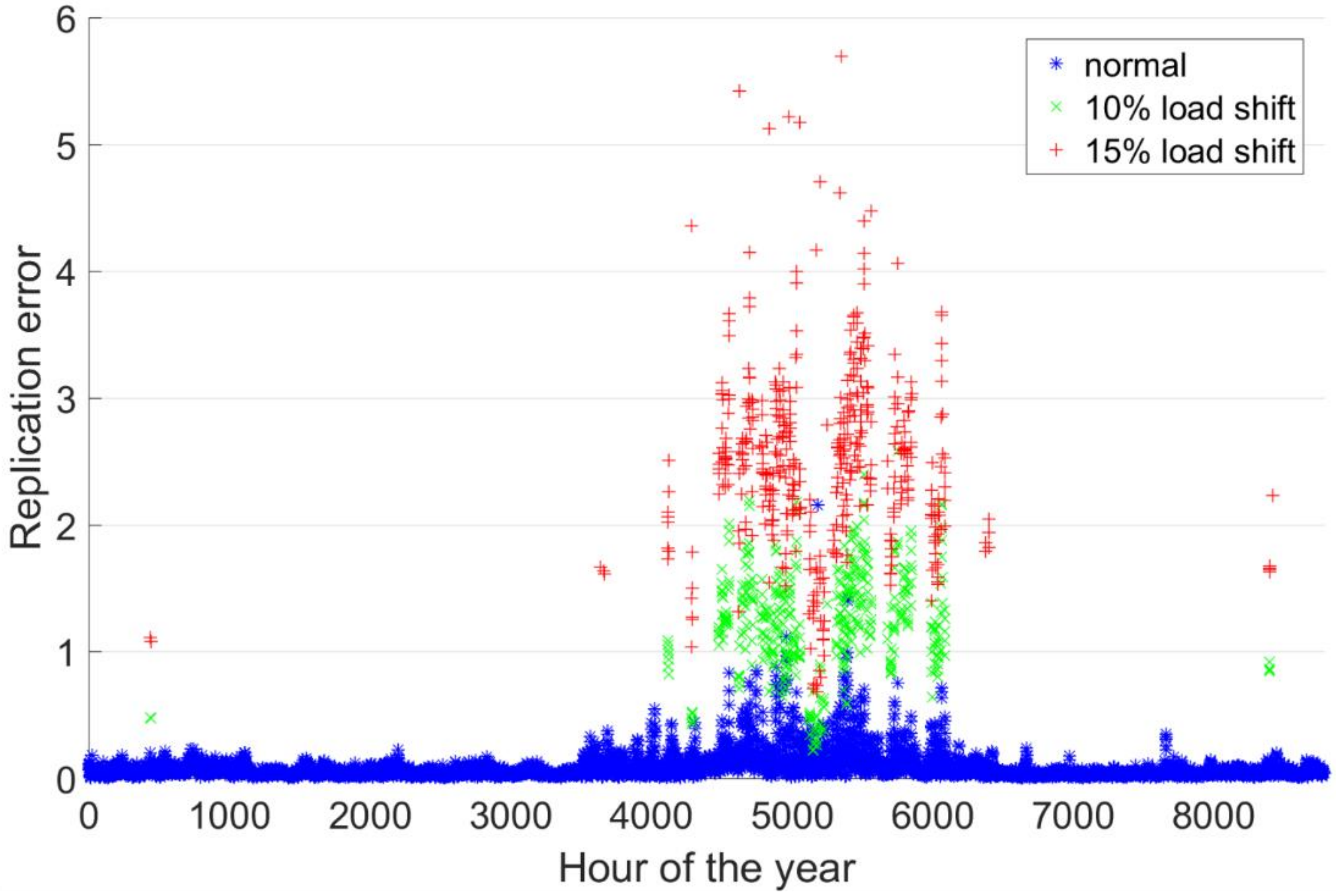
Data Summary

Normal data:
- *Training dataset:* 20 loads x 35064 hours (2012 to 2015)
- *Testing dataset:* 20 loads x 8784 hours (2016)

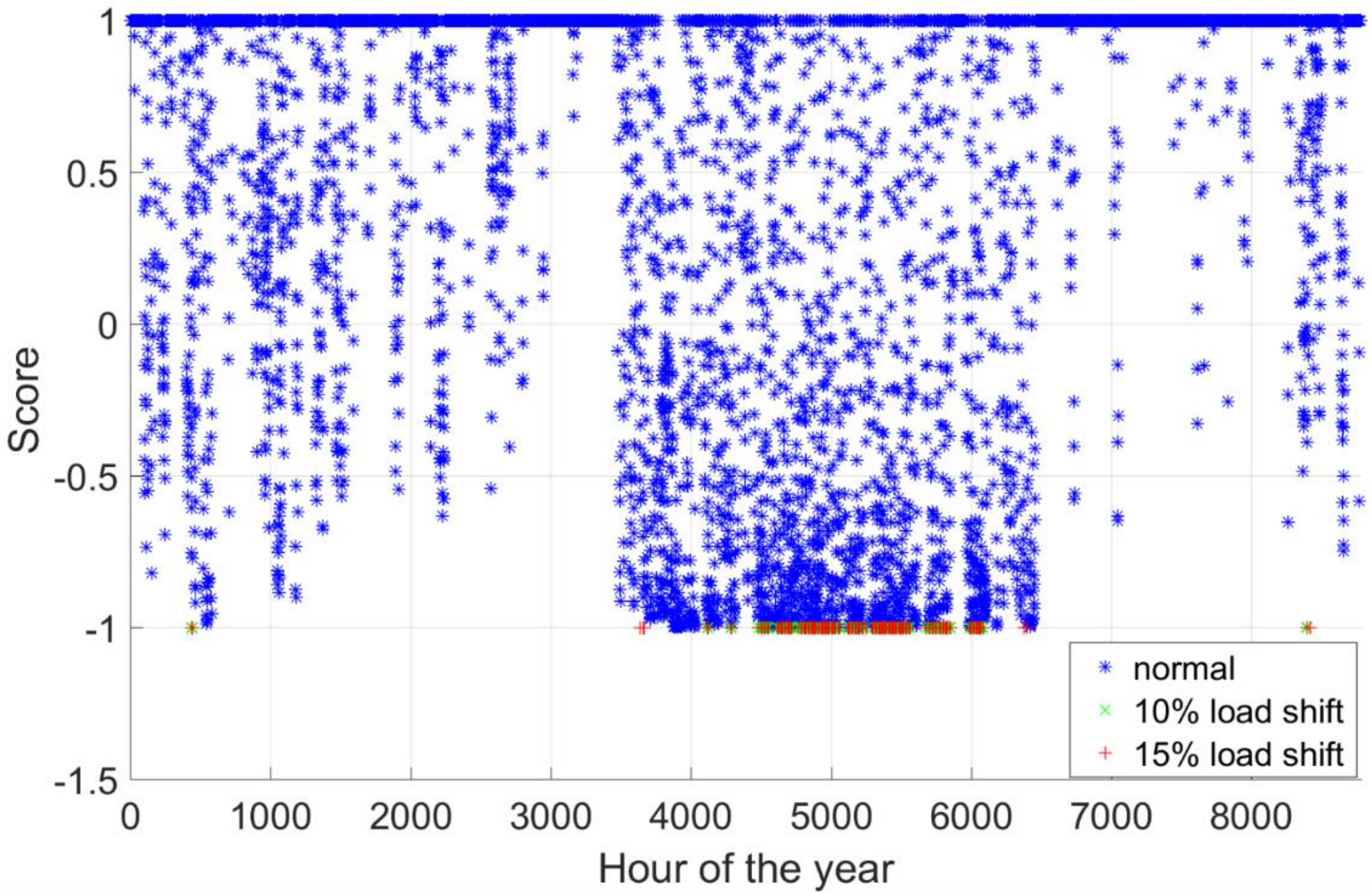
<u>Training on normal data</u>
<u>Testing on both normal and attacked data</u>

Attacked data:
- $\mathrm{LS} = 10\%$*:* 20 loads x 437 attack cases
- $\mathrm{LS} = 15\%$*:* 20 loads x 479 attack cases

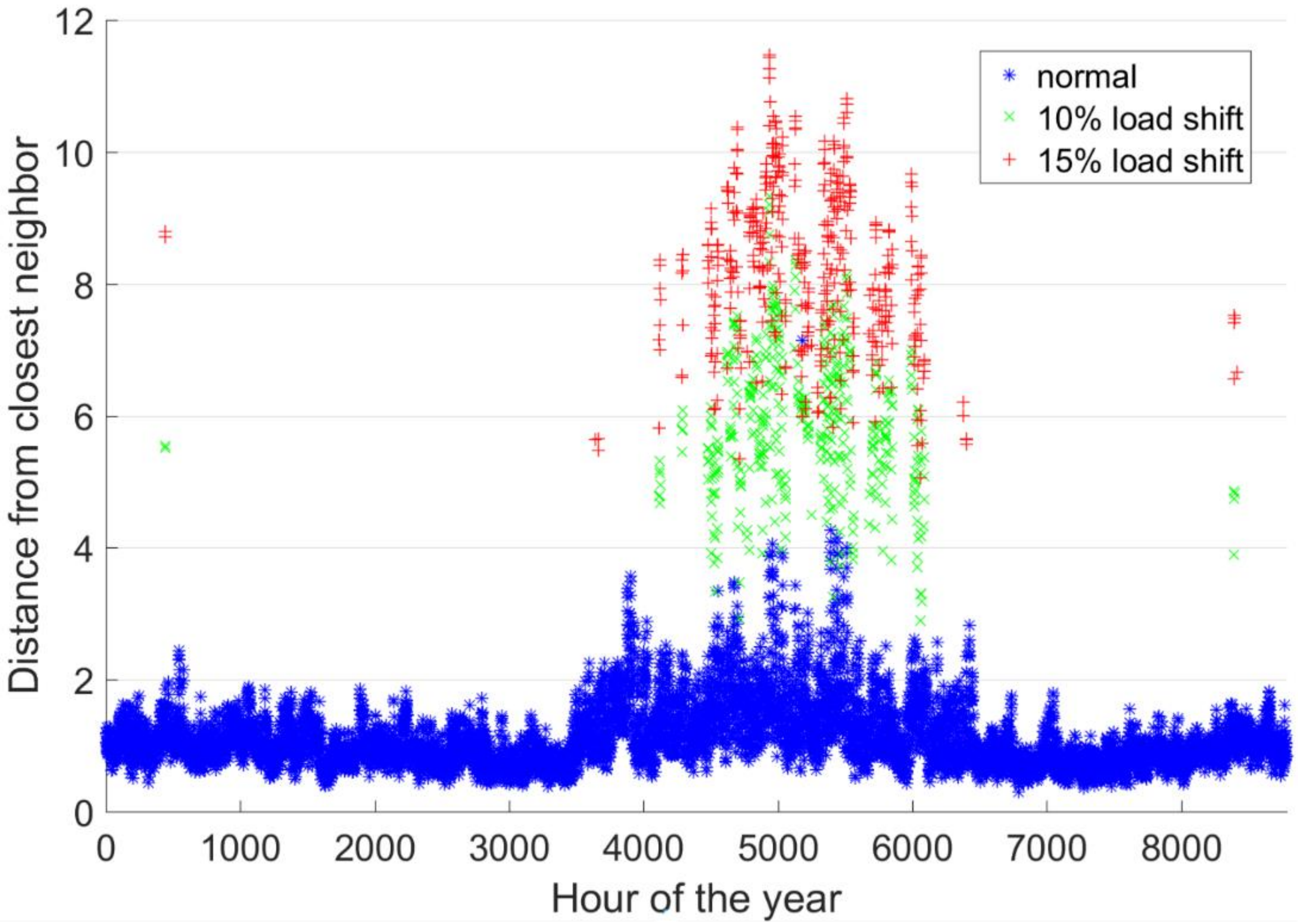
Replicator Neural Network



Support Vector Machine



Nearest Neighbor

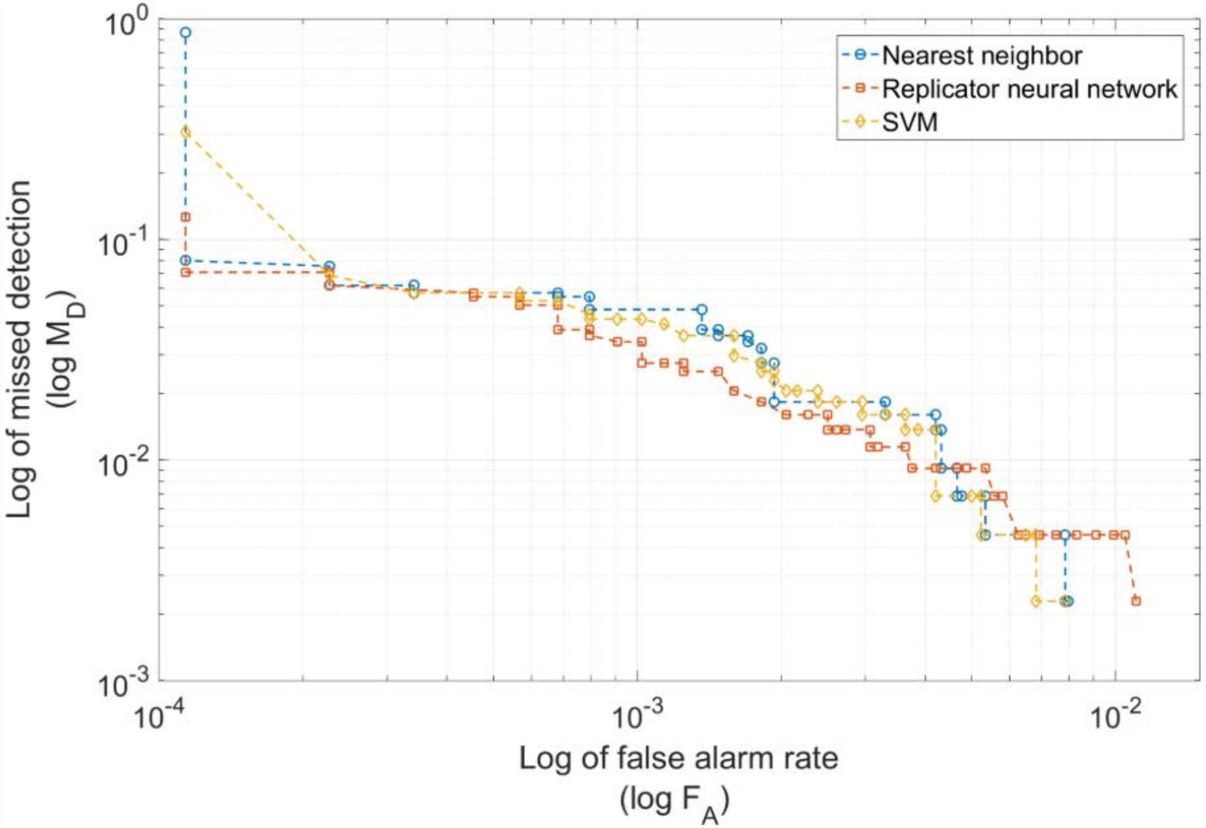


Performance Evaluation

- The performance of the detectors is evaluated in terms of:
  - *missed detection* $(M_D) = \dfrac{\text{\# attacked cases flagged as normal}}{\text{\# total number of attacked cases}}$
  - *false alarm rate* $(F_A) = \dfrac{\text{\# normal cases flagged as attacked}}{\text{\# total number of normal cases}}$
- For each detector, the threshold is varied over a wide range, thereby characterizing the tradeoff between M_D and F_A
- These results are used to plot the receiver operating characteristic (ROC)

Performance Comparison

10% Load Shift



15% Load Shift

Detector	M_D	F_A	# *of false alarms*
Nearest neighbor	0.2192	0	0
	0	1.138×10^{-4}	1
SVM	0.0021	1.138×10^{-4}	1
	0	2.277×10^{-4}	2
Replicator Neural Network	0.0125	0	0
	0	1.138×10^{-4}	1

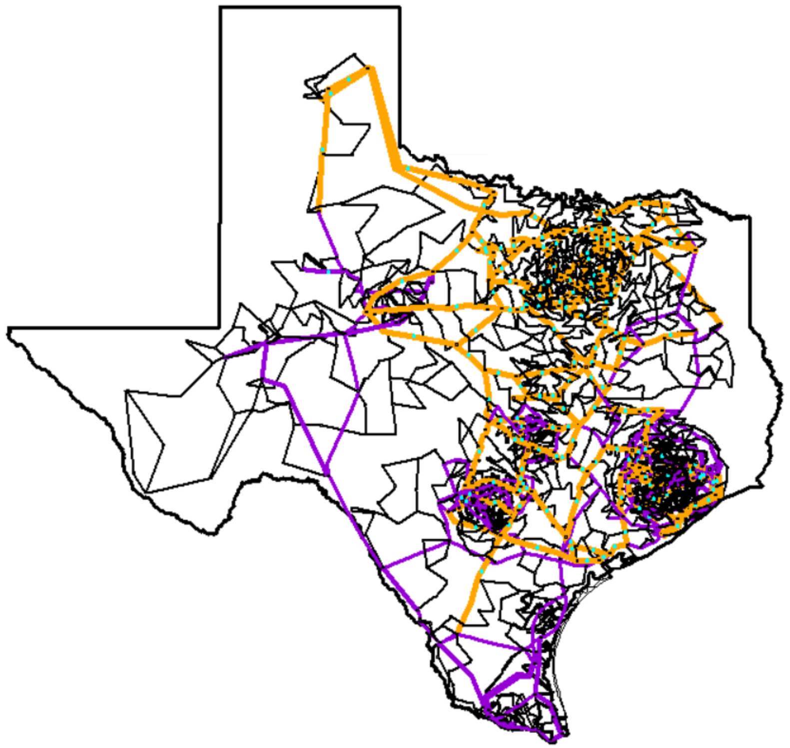
Large Scale Systems

The three techniques show comparable detection performance

The nearest neighbor-based detector is chosen for its computational efficiency and simplicity

Synthetic Texas system [1]

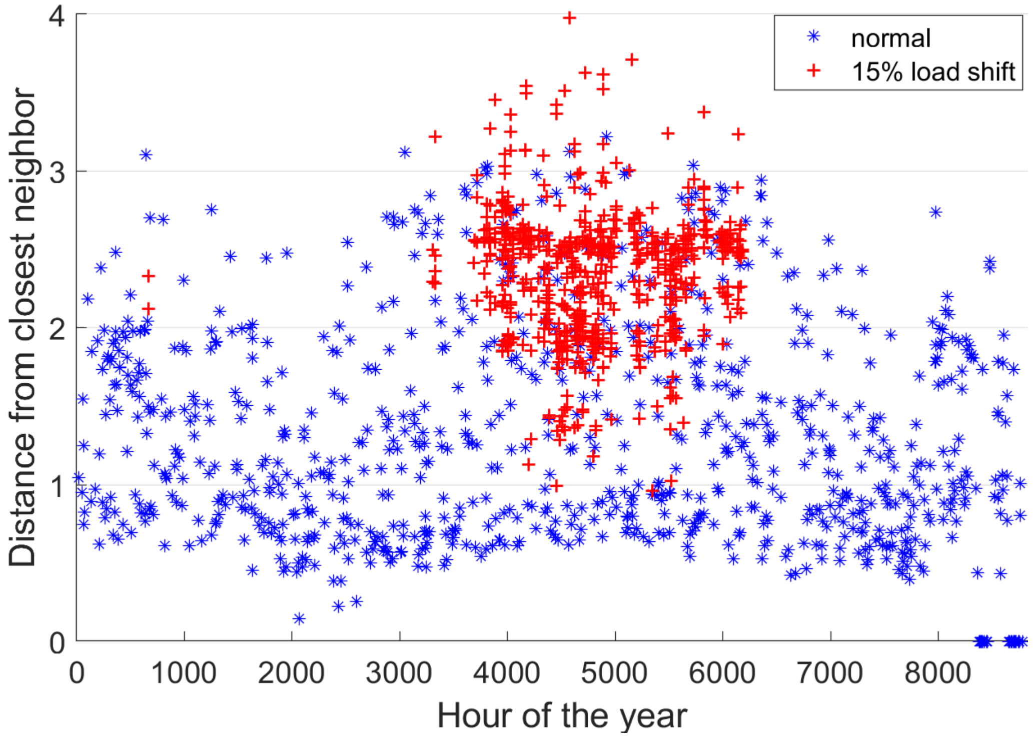
- Synthetic grid model on the footprint of the state of Texas
- 2000 buses and 1125 loads
- One year of bus-level hourly load data



[1] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258–3265, July 2017

Improvements required

- The 30-bus system has 20 loads, the Texas system 1125
- On a large system, attacker needs to only target a small, localized subset of the loads
- Measuring the Euclidean distance between 1125th dimensional vectors is not optimal when we care about the change of 100 loads or less



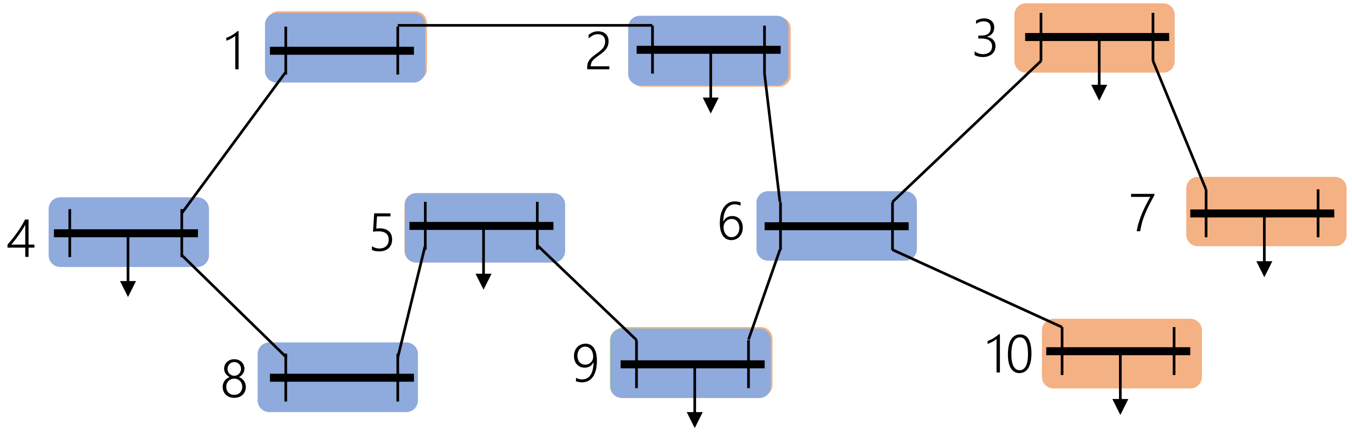
How to overcome this problem?

Grouping strategy

We divide the loads into groups which are analyzed independently

Groups are created by:

1. Starting from the largest load, define a group containing its neighbors within a radius r_g
2. Take the next largest load, if it's not contained in the previous group, create a new one with its neighbors
3. Repeat until 35 groups are created



Example

$r_g = 3$

4 : 4 – 2 – 5 – 9

9 : skip

3 : 3 – 7 – 2 – 9 – 10 – 5

⋮

Detection model

Given a load vector p, the nearest neighbor distance is calculated for each load group j

$$d_j = \min_{r=[1:n_h]} \left\| \boldsymbol{p}^j - \boldsymbol{h}_r^j \right\|_2$$

The vector p is labelled as attacked if $n_a \geq 1$, where

$$n_a = \sum_{j=1}^{n_g} \mathbb{1}(d_j > \tau_j)$$

$\boldsymbol{p} \in \mathbb{R}^n$, where n is total number of system loads

n_a: number of anomalous load groups

n_g: number of load groups

d_j: minimum distance for group j

τ_j: threshold for group j

$\boldsymbol{p}^j$: vector of loads in group g_j

$\boldsymbol{h}_r^j$: subset of loads in group g_j from r^{th} historical vector

n_h: the number of historical load vectors

Example of attack detection with load grouping

- 10 loads
- 3 groups: yellow, green, and pink
- 4 normal cases and 2 attacked cases

Normal cases

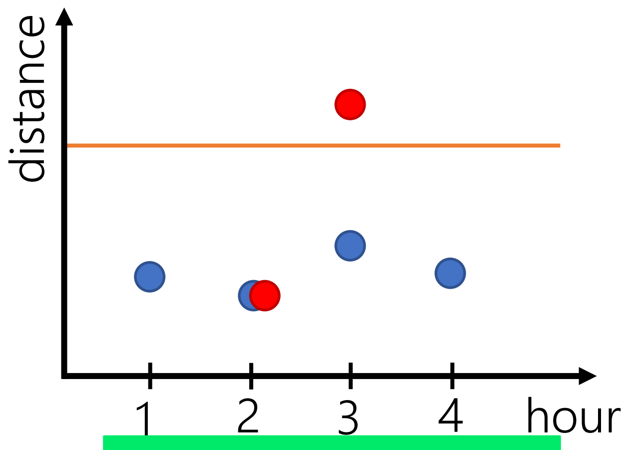
LOADS

HOURS										
1	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
2	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
3	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
4	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}

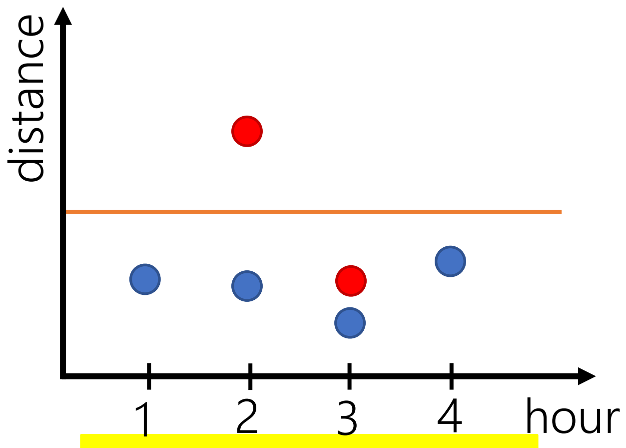
Attacked cases

$\cancel{L_1}$	$\cancel{L_2}$	L_3	L_4	$\cancel{L_5}$	L_6	L_7	L_8	L_9	L_{10}
L_1	L_2	L_3	$\cancel{L_4}$	L_5	L_6	$\cancel{L_7}$	$\cancel{L_8}$	L_9	L_{10}

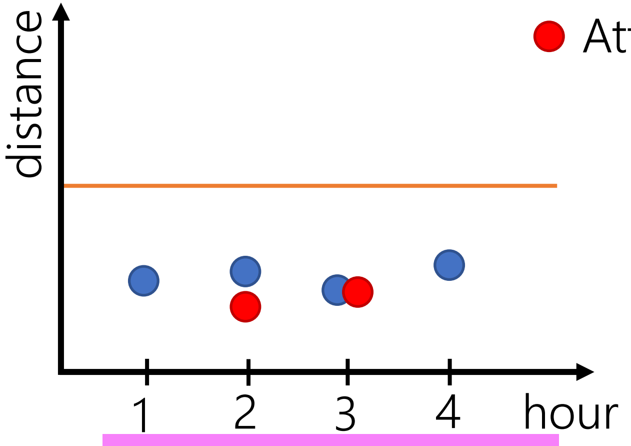
Threshold
Normal vector
Attacked vector



Group green



Group yellow



Group pink

Testing procedure

Normal data

- Hourly load data for the year 2016 → 1125 loads over 8784 base cases

$$\boldsymbol{P}_N \in \mathbb{R}^{1125\times 8784}$$

Attacked data

- Attacks are computed for every congested line (flow on a line > 90%), across every hour;
- Load shift factors ranging from 1% to 15% are used;
- A total of 8861 successful attacks are computed

$$\boldsymbol{P}_A \in \mathbb{R}^{1125\times 8861}$$

Testing procedure

To compute detection probability and false alarm:

- $\boldsymbol{P}_N$ divided into three subsets:
  - $\boldsymbol{P}_N^{hist}$ historical dataset 70% of the columns of $\boldsymbol{P}_N$
  - $\boldsymbol{P}_N^{train}$ training dataset 20% of the columns of $\boldsymbol{P}_N$
  - $\boldsymbol{P}_N^{test}$ testing dataset 10% of the columns of $\boldsymbol{P}_N$

- For group j, the distance between each vector i in $\boldsymbol{P}_N^{train}$ and $\boldsymbol{P}_N^{hist}$ is calculated:

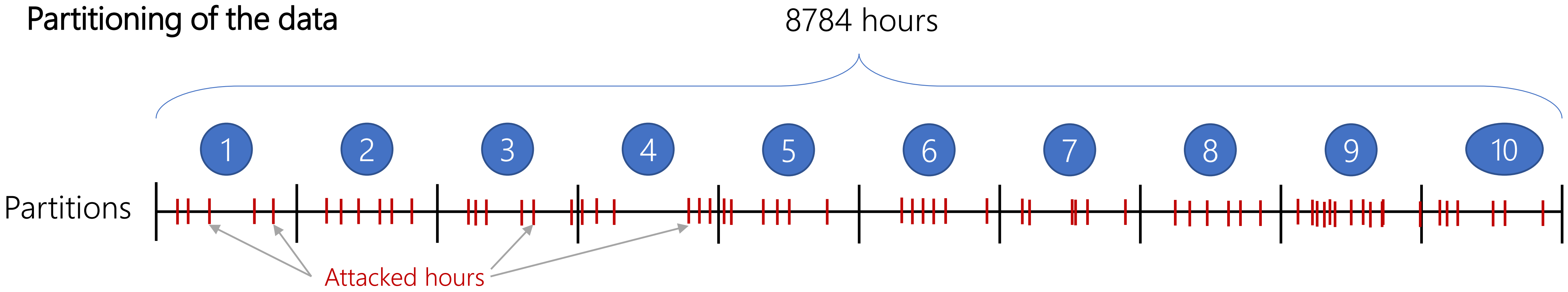
$$d_{i,j} = \min_{r=[1:n_h]} \left\| \boldsymbol{p}_i^j - \boldsymbol{h}_r^j \right\|_2$$

- Threshold τ_j is calculated as

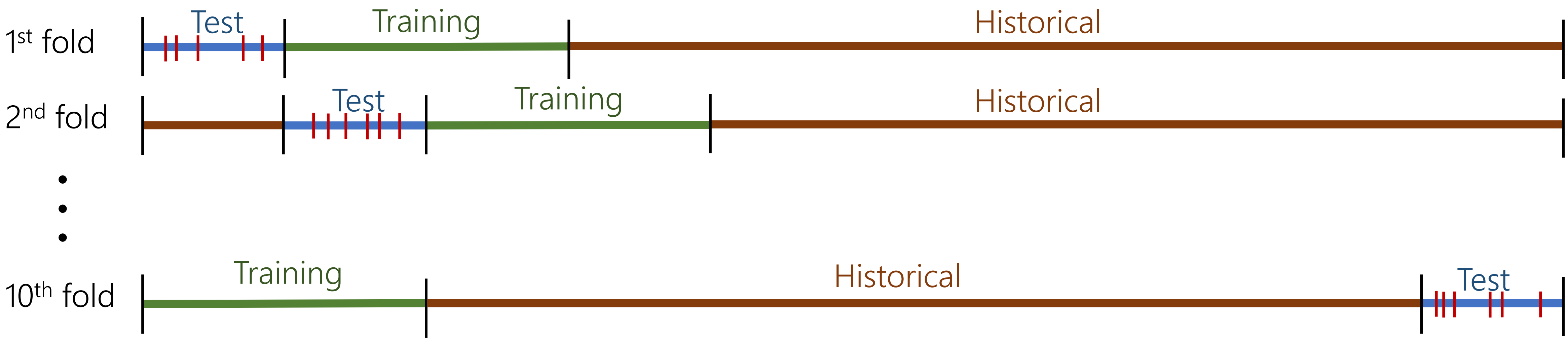
$$\tau_j = \alpha \left(\max_{i \text{ in } \boldsymbol{P}_N^{train}} d_{i,j} \right)$$

Testing procedure

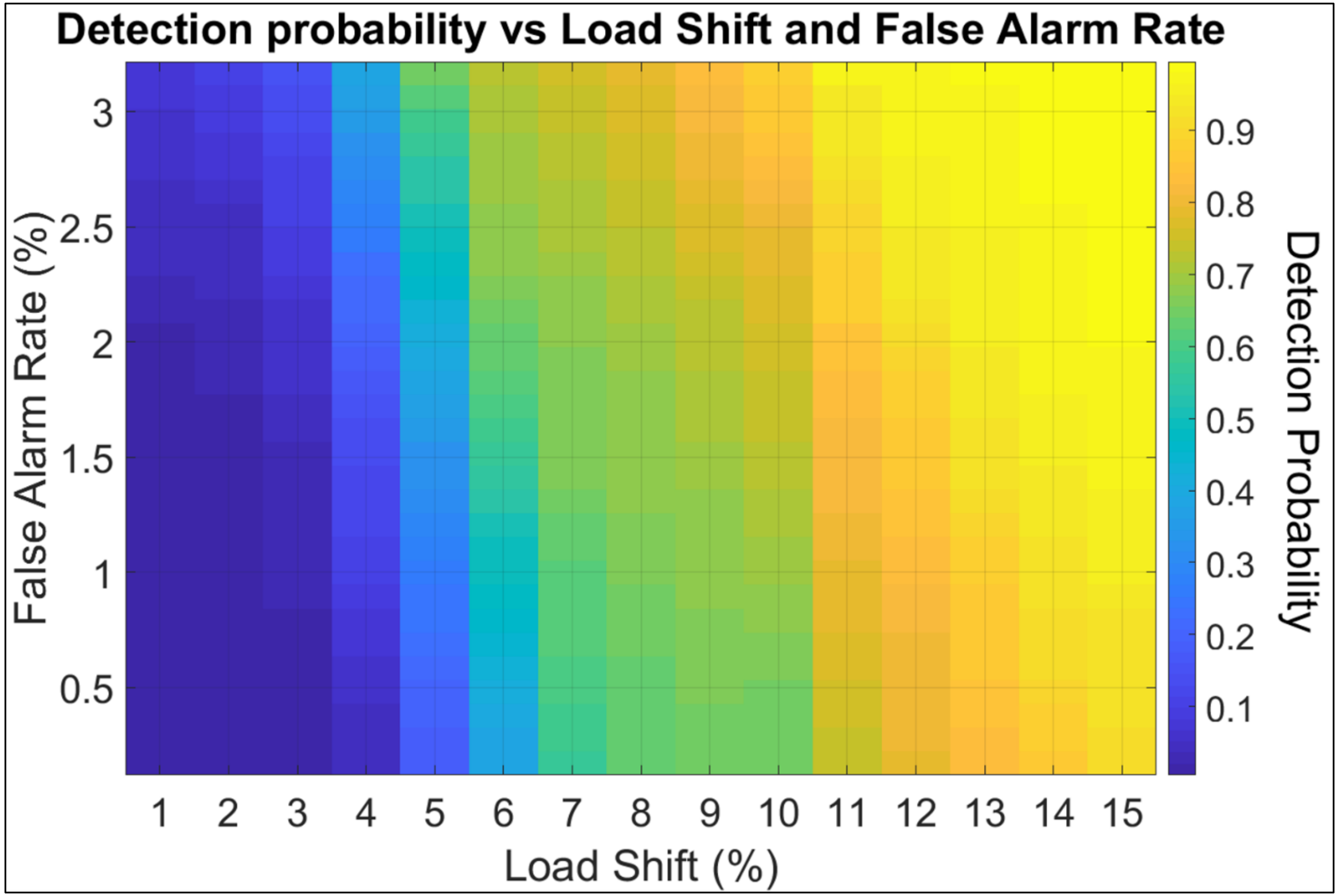
Partitioning of the data



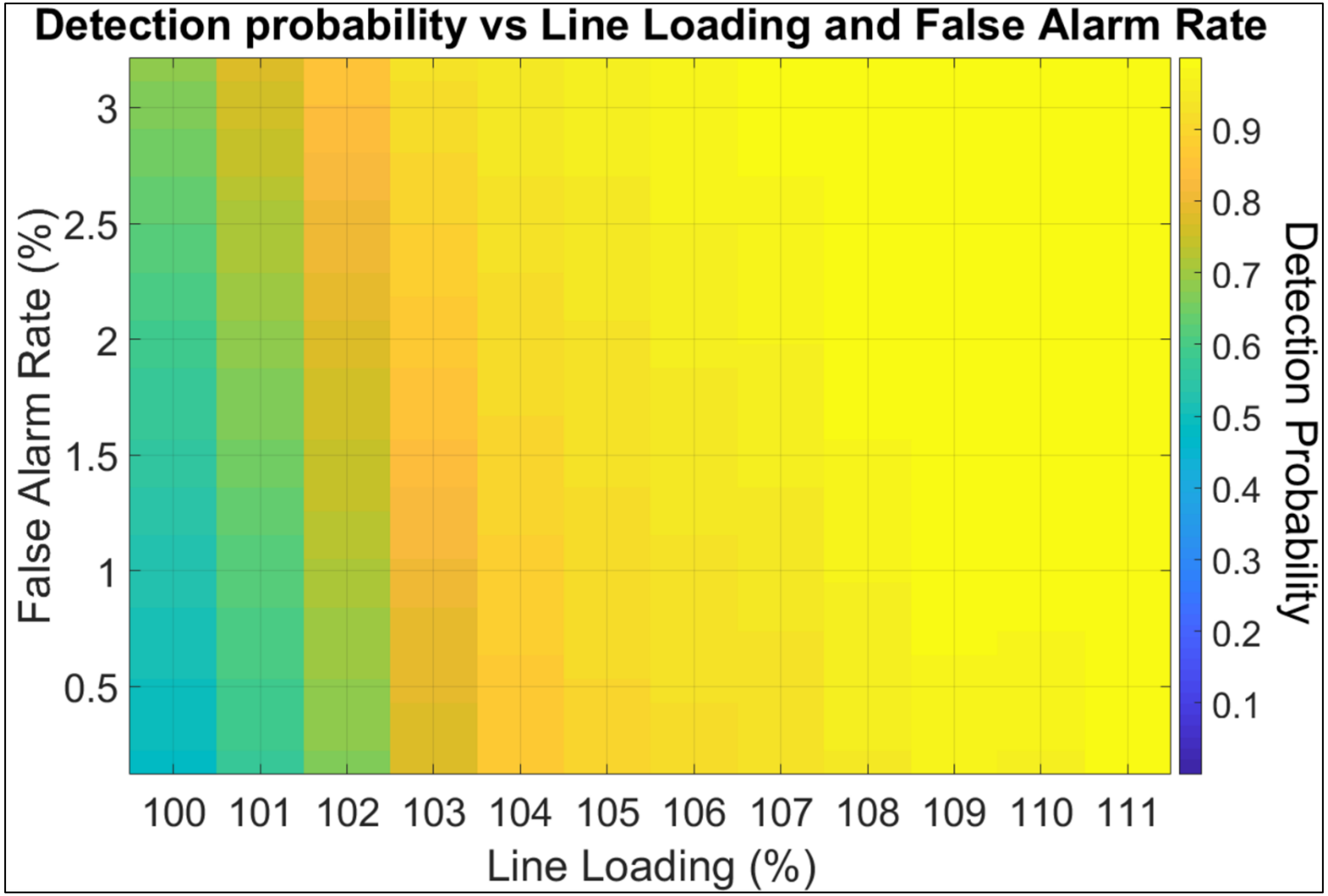
Folding of the testing partitions



Results


Detection probability vs Load Shift and False Alarm Rate
False Alarm Rate (%)
3
2.5
2
1.5
1
0.5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Load Shift (%)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Detection Probability

Results


Detection probability vs Line Loading and False Alarm Rate
False Alarm Rate (%)
3
2.5
2
1.5
1
0.5
100 101 102 103 104 105 106 107 108 109 110 111
Line Loading (%)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Detection Probability

Ongoing and future work

How well does the detector perform against "random" load redistribution attacks?

- Sensitivity analysis with varying:
  - Load shift
  - Attack subgraph size

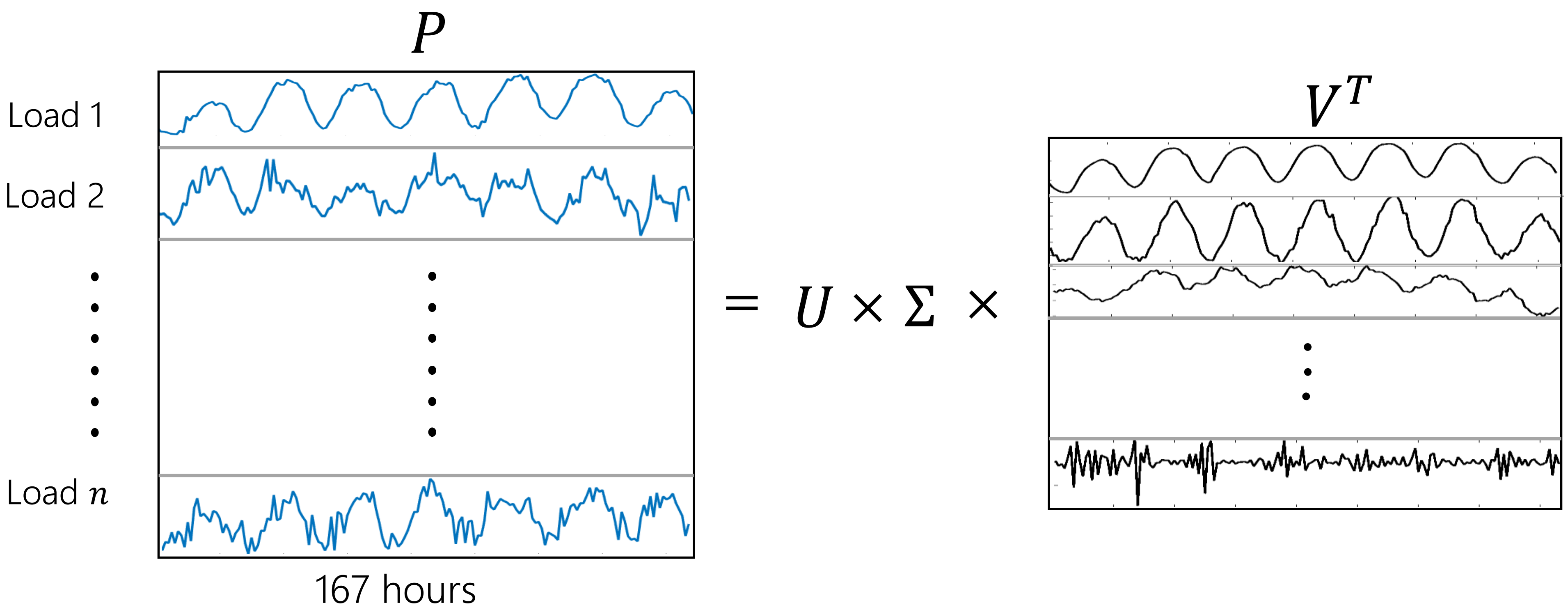
How can we leverage the detector for more secure operations?

- Attack localization
  - Determining which areas and measurements can be securely controlled
- Likelihood of each load of being attacked
  - Improved decision making tools that consider load uncertainty

Ongoing and future work

Generation of synthetic time-series load data

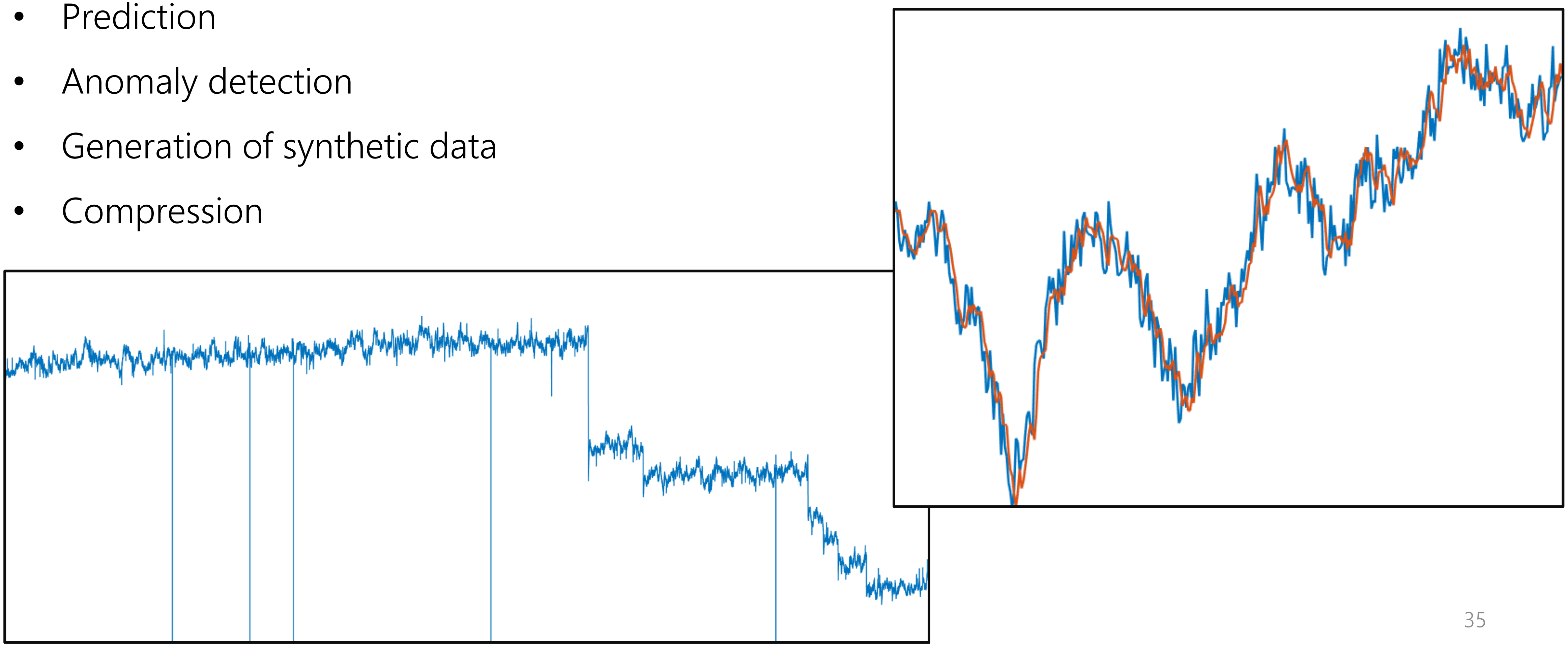
- Singular value decomposition (SVD) to extract typical load patterns
- Learning of the spatio-temporal behavior of loads



Ongoing and future work

Leveraging the proposed learning techniques for the study of PMU data:

- Prediction
- Anomaly detection
- Generation of synthetic data
- Compression



Publications

Conference papers

- A. Pinceti, L. Sankar, and O. Kosut, "Load Redistribution Attack Detection using Machine Learning: A Data-Driven Approach," *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 2018
- A. Pinceti, O. Kosut, and L. Sankar, "Data-Driven Generation of Synthetic Load Datasets Preserving Spatio-Temporal Features", accepted *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA
- Z. Chu, A. Pinceti, R. Sen Biswas, O. Kosut, A. Pal, and L. Sankar "Predictive Filters Cannot Detect Gradually Ramping False Data Injection Attacks Against PMUs", SmartGridComm 2019, **to be submitted**

Journal papers

- A. Pinceti, L. Sankar, and O. Kosut, "Detection and Localization of Load Redistribution Attacks on Large Scale Systems", **to be submitted**

Thank you!

Questions?

Thank you!

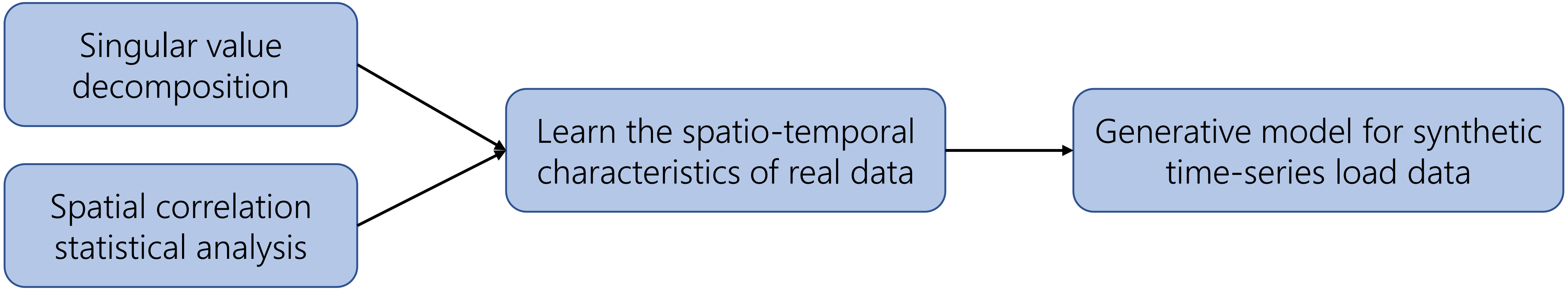
Questions?

Generation of Synthetic Time-series Load Data

Synthetic load data

Motivation

- Historical load data crucial in almost every field of power systems
  - Traditional studies (stability, transmission expansion planning, multi-temporal unit commitment and economic dispatch, etc..)
  - Emerging topics (machine learning applications, cybersecurity, etc..)
- Limited availability of public time-series load data



Dataset description

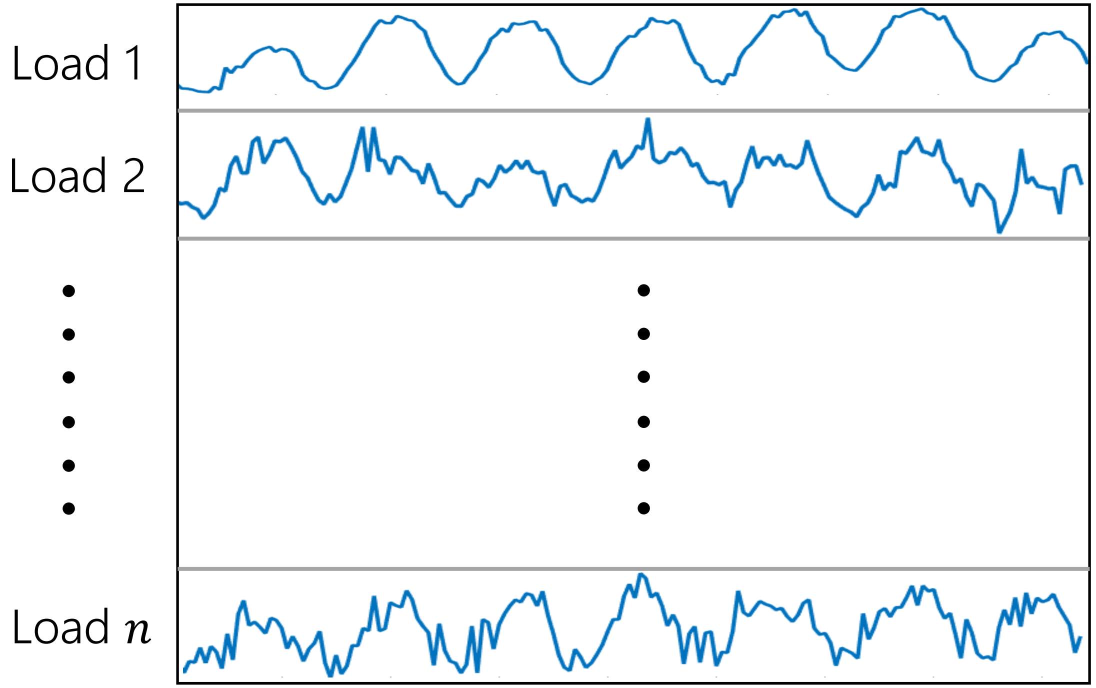
The learning phase requires:

- Historical load data $P \in \mathbb{R}^{n \times t}$
  - $t = 167$ hours
  - $n \sim 3500$ loads
- System topology

The generative phase requires:

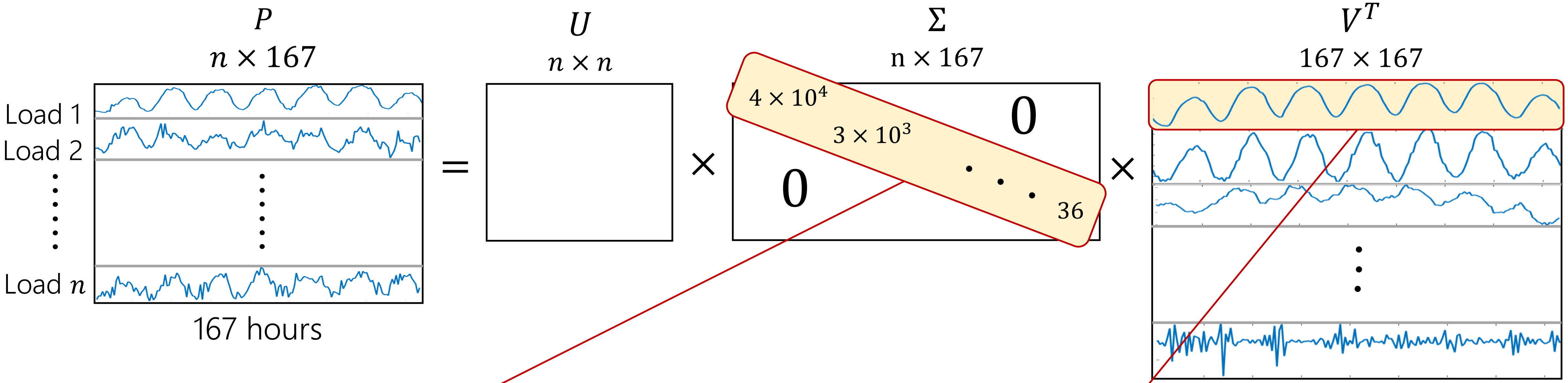
- System topology of the new system

$P =$

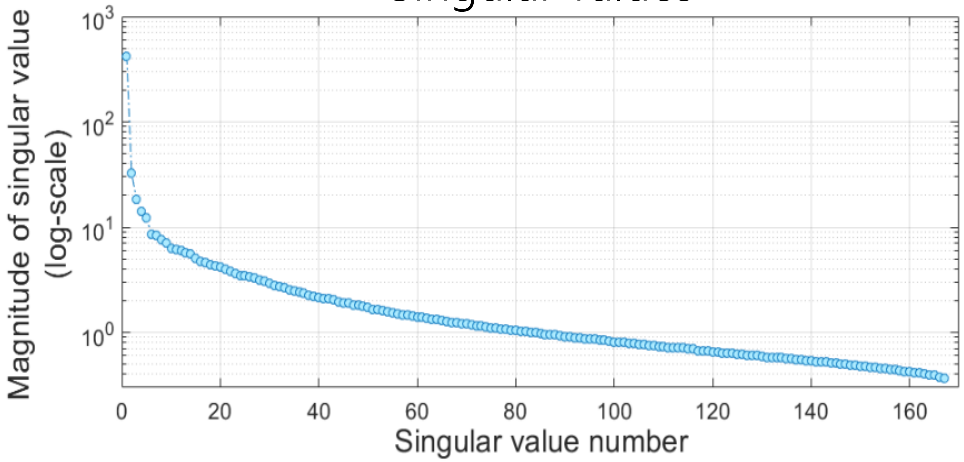


167 hours

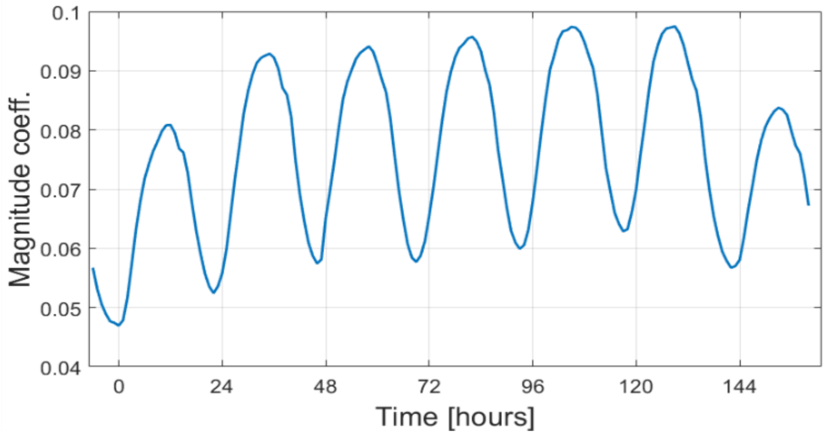
Singular value decomposition



Singular values

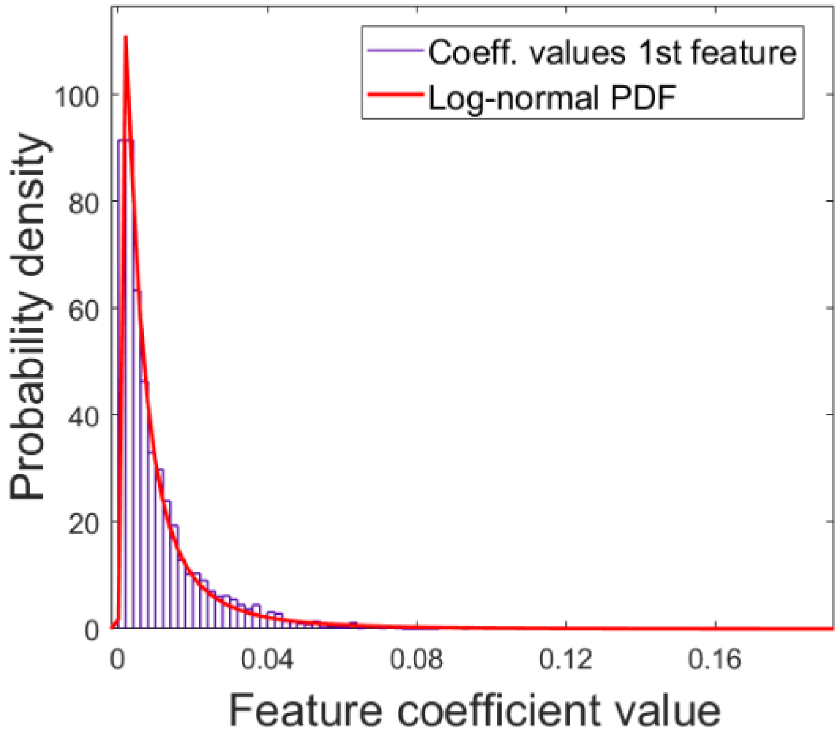


First feature

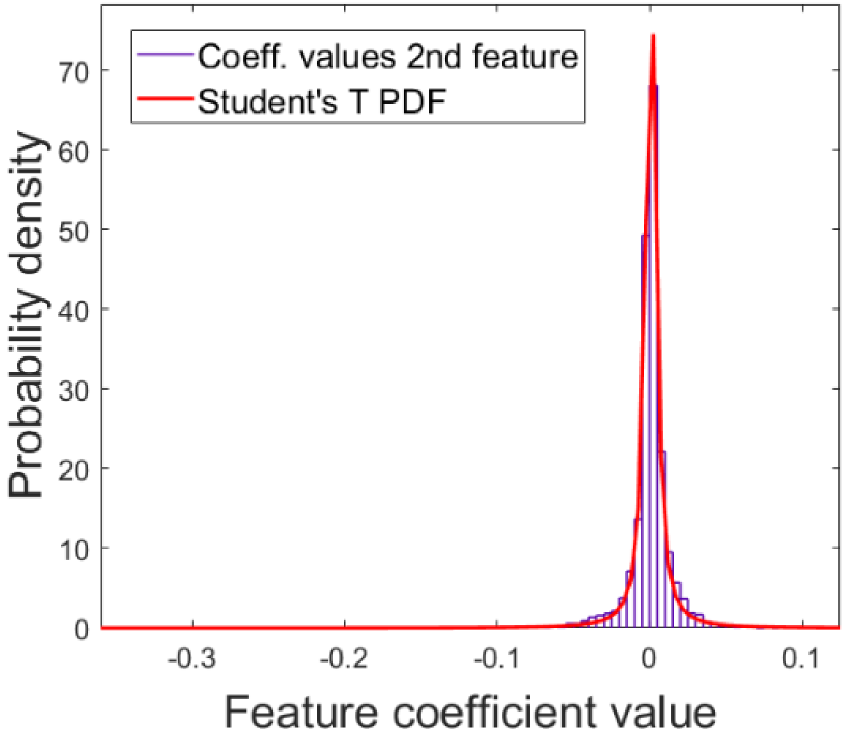


- Synthetic load profiles are linear combinations of the features (rows of V^T)
- Generation process:
  - Learn distribution of coefficients of U (by column)
  - Sample from distributions to create U_{new}
  - New load profiles $P_{\text{new}} = U_{\text{new}}\,\Sigma\,V^T$

1st column of $\boldsymbol{U}$



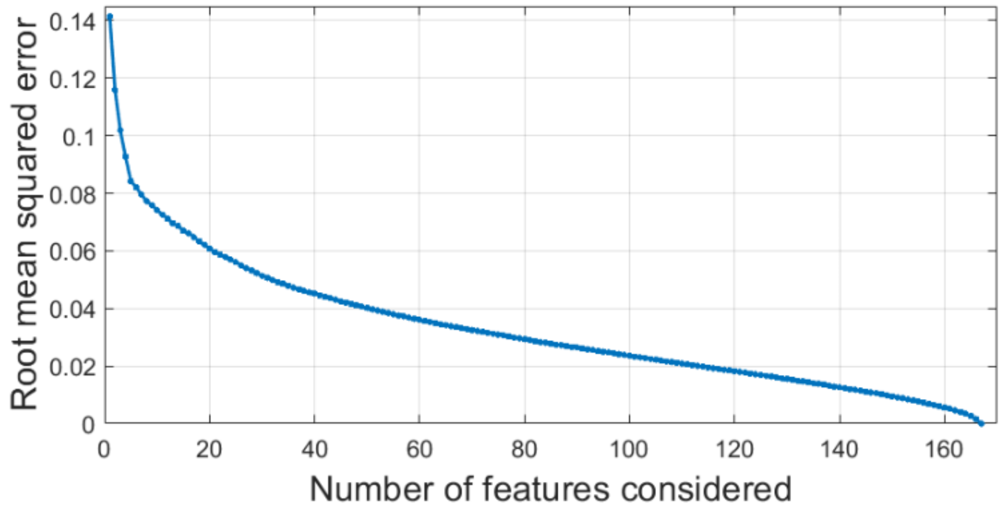
2nd column of $\boldsymbol{U}$

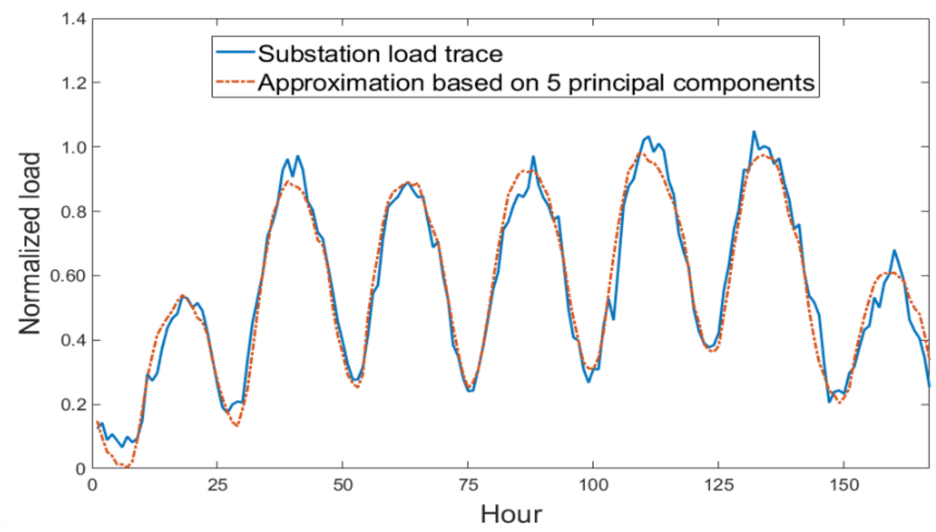


Feature selection

Root mean square error between P and $\hat{P} = U^f \, \Sigma^f \, {V^f}^T$

First five features are selected





Sample load trace and its approximation using first 5 features

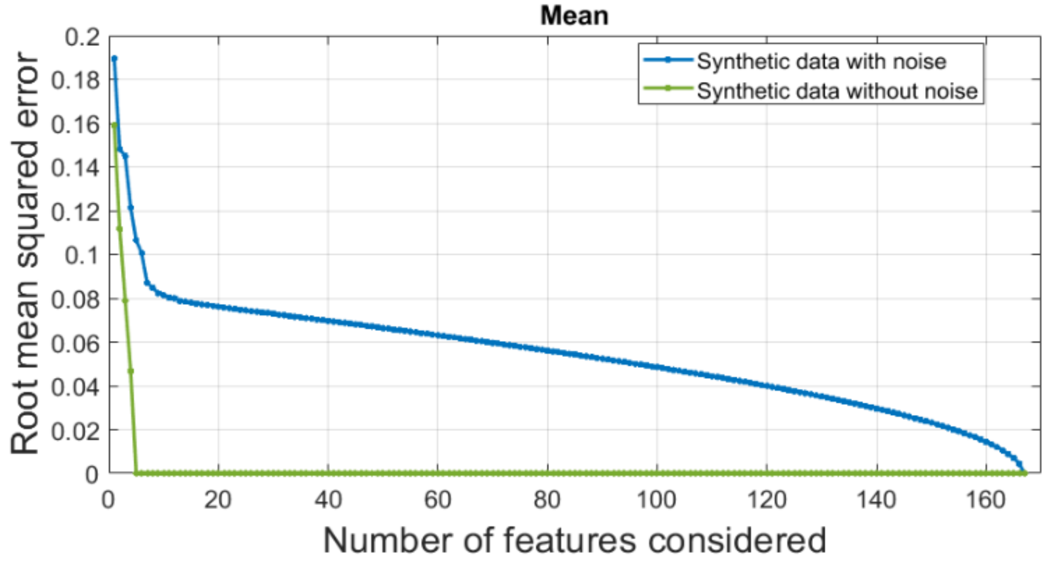
Notation: U^f indicates the first f columns of U, Σ^f first f columns and f rows of Σ, and V^f first columns of V

Feature selection

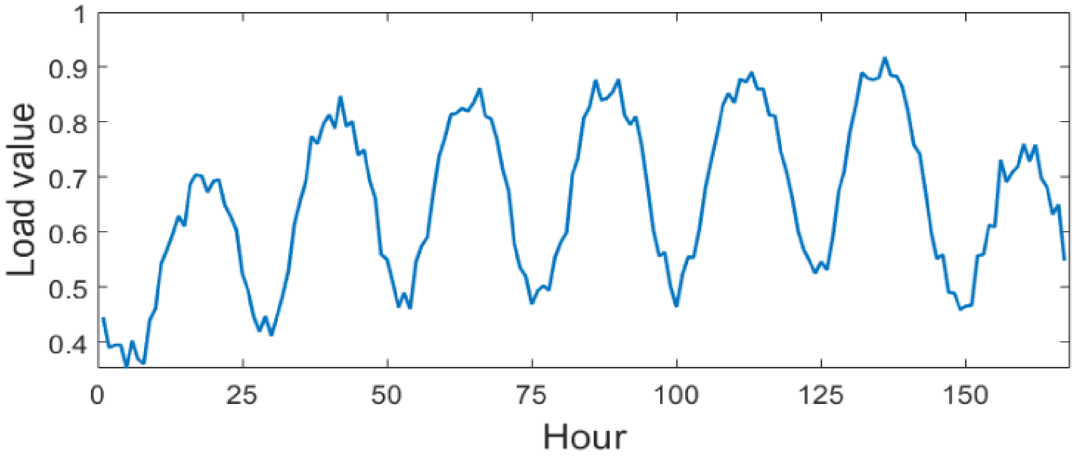
Synthetic profiles generated for 1000 loads as:

$$P_{\text{new}} = U_{\text{new}}^{5}\, \Sigma^{5}\, {V^{5}}^{T} + W$$

where W is the noise matrix



Sample load trace extracted from P_{new}

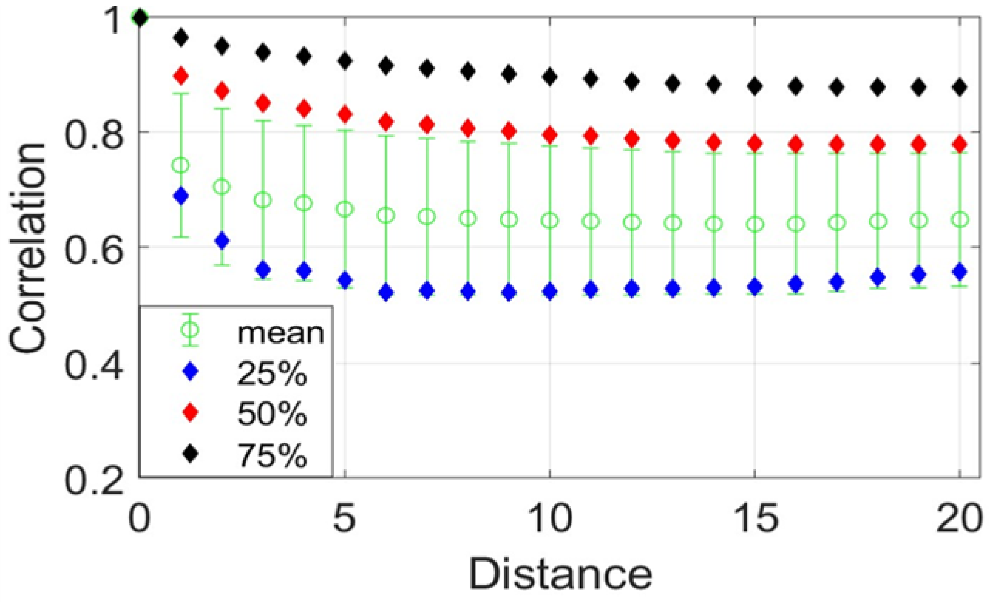


Notation: U^f indicates the first f columns of U, Σ^f first f columns and f rows of Σ, and V^f first columns of V

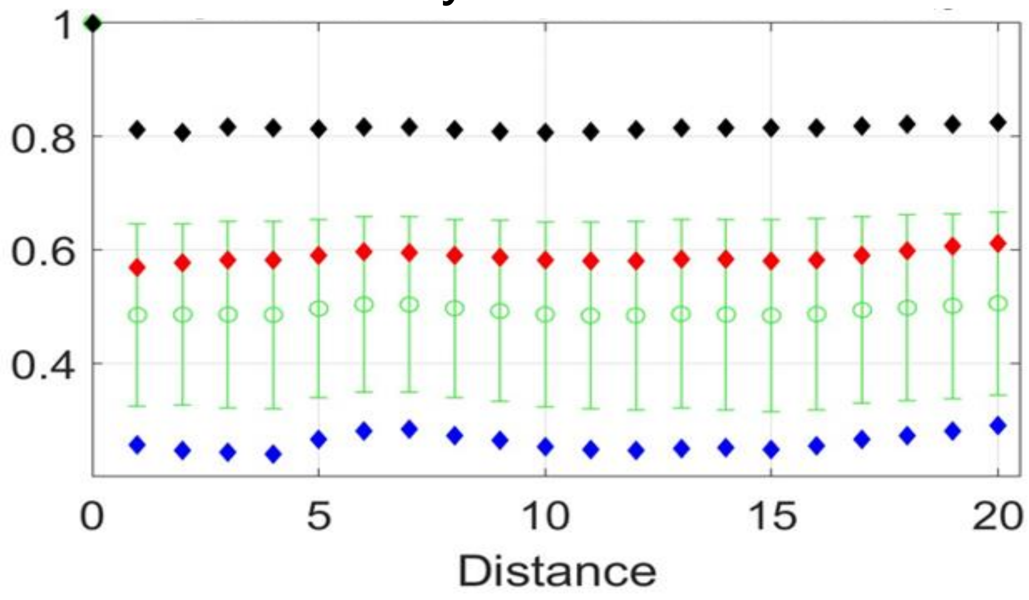
Correlation between load profiles

- Load profiles are spatially correlated
- Spatial correlation given by:
  - Load composition (residential, commercial, industrial..)
  - Geography-dependent factors (e.g. weather conditions)

Real data



Synthetic data



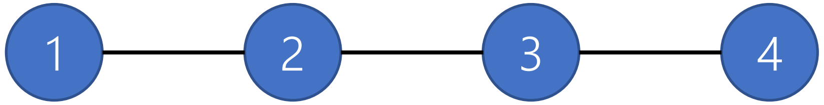
Modified generative model

Independently generated coefficients are adjusted according to coefficients of neighboring loads

$$P_{new} = (DU_{new})\Sigma V^T + W = U'_{\text{new}}\Sigma V^T + \text{W}$$

Where, each entry $d_{i,j}$ of D is given by:

$$d_{i,j} = \begin{cases} 1, & \text{if } \text{i} = \text{j} \\ e^{-2dist_{i,j}}, & \text{if dist}_{\text{i,j}} \leq 3 \text{ and } \text{i} \neq \text{j} \\ 0, & \text{otherwise} \end{cases}$$



$$U_{\text{new}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}, \text{ where } u_i \in \mathbb{R}^{1\times 5}$$

$$u'_1 = u_1 + d_{1,2}\, u_2 + d_{1,3}\, u_3$$

Modified synthetic data