ELEC97077: Smart Grid Tech Coursework on Detecting False Data Injection Attack in Power System

Presented by Dr.Wangkun Xu
wangkun.xu18@imperial.ac.uk

Nov. 2024

Introduction

Background

The cyber-physical smart grid (CPSG), which is powered by advanced communication and digitalisation techniques, is vulnerable to malicious cyberattacks. Recently, False Data Injection (FDI) attack has drawn great attention due to its high stealthiness and adverse impacts on the state estimation (SE) of CPSG. The consequences of falsified SE include economic losses, transmission line overflow, system instability, and blackout. To enhance the resilience of the power system, it becomes critical to develop fast and accurate detection for such attacks.

Overview of the Coursework

- The main purpose of this coursework is to get familiar with the *state estimation* and *bad data detection (BDD)*. Due to the stealthy nature of FDI attacks, BDD in the control center likely fails. Therefore, you are asked to develop machine learning (ML) algorithm(s) to forecast the system state and detect FDI attacks as well.

- You will be provided with a dataset of sensor measurements, simulated by DC optimal power flow (DC-OPF) on realistic loads. You must choose from `Matlab` or `Python` as your programming language for scientific computation in this coursework. For the ML tasks, it is recommended to use well-developed ML packages, such as Matlab's Statistics and Machine Learning Toolbox[1] or Python's scikit-learn[2], etc. For Python user, you can use Google Colab to avoid managing package dependency on your local machine. However, your submission **must** be in Python `.npy` format.

- You will work in a group of **two**. The maximum length of the report is 3 pages. Extra pages will not be marked. You need to attach your code in the appendix (not included in the 3-page limit). For all tasks, you need not only to provide the numerical results, **but should briefly report the procedures**, in the format of pseudo-code, descriptions, or mathematical equations, etc. Some tasks may require knowledge slightly beyond the lecture note, but can be easily attained online.

- There will be five tasks in this coursework. The first three will assess your knowledge and analytical skills for power system state estimation and bad data detection. The last two tasks focus on basic machine learning applications. The purpose of this coursework is not to develop advanced and complex algorithms, but rather to evaluate your insight on how to develop a successful machine learning pipeline.

[1] `https://uk.mathworks.com/products/statistics.html`
[2] `https://scikit-learn.org/stable/`.

- For any questions on the coursework, please contact Dr. Wangkun Xu via MS teams or by email: `wangkun.xu18@imperial.ac.uk`.

Courseworks

Task One: Get Familiar with the Dataset and the Power Grid [10/100]

Go to website [3] and download the `train_dataset.npy` and `dataset_test_feature/`.

In the `dataset_train/`, you have the training dataset `train_dataset.npy` for Python user containing both the training feature and labels. The training features are the measurements (in per unit) generated by simulating the DC-OPF on the IEEE bus-14 system under realistic load and renewable profiles.

In the `train_dataset`, there are three classes (labels), including

- 0: for normal measurements;
- 1: for FDI attack measurements;
- 2: for random attack measurements.

All the three classes of data have **equal** size. Load the dataset to your working space and answer the following questions based on `dataset_train`.

1. What is the size of the training dataset? How many samples do you have for each class and how many sensor measurements are collected? [2/100]
2. Referring to the IEEE bus-14 description[4], the topology of the bus-14 system can be configured from the `branch data` where the `fbus` and `tbus` represent the from-side and to-side bus index for each transmission line. Based on the information, construct the incidence matrix $\boldsymbol{A}$ of bus-14 system. [3/100]

   *Hint: you must construct the incidence matrix following the **same** sequence in the website. For example, the first row of your incidence matrix has 1 at the first element and -1 at the second element.*
3. Referring to the IEEE bus-14 description, the line susceptance $\boldsymbol{b}$ can be calculated as $1/\boldsymbol{x}$. Based on the information, calculate the branch susceptance matrix $\boldsymbol{B}_{pf} = \text{diag}(\boldsymbol{b})\boldsymbol{A}$ and the bus susceptance matrix $\boldsymbol{B}_{pi} = \boldsymbol{A}^T\text{diag}(\boldsymbol{b})\boldsymbol{A}$. Observing the structure of the $\boldsymbol{B}_{pi}$ matrix, what can you find? [5/100]

Task Two: State Estimation [15/100]

The measurements are composed of branch power flow $\boldsymbol{P}_{pf} \in \mathbb{R}^{20}$ for all 20 branches and bus power injection $\boldsymbol{P}_{pi} \in \mathbb{R}^{14}$ for all 14 buses. For each sample, they are packed as a vector $[\boldsymbol{P}_{pf}^T, \boldsymbol{P}_{pi}^T]^T \in \mathbb{R}^{34}$.

It is assumed that the measurement noise on power flow follows independent normal distribution with mean $\mu_{pf} = 0$ and standard deviation (std) $\sigma_{pf} = 0.01$. The measurement noise on power injection follows independent normal distribution with mean $\mu_{pi} = 0$ and std $\sigma_{pi} = 0.02$. The first bus is the reference bus with bus angle equal to $\boldsymbol{\theta}_{ref} = 0$.

Based on the information above, answer the following questions:

1. Construct the covariance matrix $\boldsymbol{R}$ for the measurement. *Hint: your covariance matrix should be diagonal of size* 34×34. [5/100]

[3] `https://imperiallondon-my.sharepoint.com/:f:/g/personal/wx3418_ic_ac_uk/Ej7dn0ociE5MjHpmQLanlNkBjWBtqF5Gtp7ToMVXY5VOSQ`

[4] `https://github.com/MATPOWER/matpower/blob/master/data/case14.m`

2. Brief describe what is state estimation in power system and why it is important to power system operation. [5/100]

3. For `train_dataset`, do state estimation for class 0, class 1, and class 2, individually, to get the bus angles using the weighted least square (WLS) approach. Report the formula for state estimation and briefly discuss how you did it. [5/100]

   Hint: remember to remove the column of the reference bus in the measurement matrix.

Task Three: Bad Data Detection [25/100]

1. Calculate the **residual vectors** for the measurements in `train_dataset`, based on the bus angles you calculated in the previous task. Report the formula and briefly discuss how you did it. [5/100]

2. Define what is true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR) **for detecting of attacks in your own words**. [5/10]

3. Assume that the system operator wants to control the false positive rate on the normal measurements to be around 5.0%. Calculate the BDD threshold γ accordingly and describe briefly how you did it. [5/10]

   Hint: On the normal measurement, the residual follows the χ^2 distribution approximately. First, you need to determine what is the 'degree of freedom' is in this problem. Then use Matlab's `chi2inv` or Python SciPy's `scipy.stats.chi2.ppf` to find the inverse cdf as the threshold. Search online for extra information.

4. **Before calculate the residuals, give a guess on** the alarm rates on normal, random attack, and FDI attack datasets, based on your knowledge from the lecture and explain why. Then for each class, calculate the residuals on the datasets. What are the alarm rates for these dataset and do they follow your intuition? You need to briefly discuss on how you achieve this. [10/100]

Task Four: Predict State Estimation Results Using Machine Learning [10/100]

In this task, you are asked to build a machine learning model to predict the state estimation results from the previous tasks. In particular, on the `training_dataset` with label 0, build a **linear regression model** to predict the state estimation results. Based on your experiment, you need to answer the following questions,

1. What is the learning task (regression or classification) in machine learning? [2/100]

2. How can you describe the training dataset, e.g., how many samples do you have and what are the feature and label? [2/100]

3. Report the accuracy of your ML model in mean squared error (MSE). What can you find and why? [6/100]

Task Five: Detecting FDI Attacks using Machine Learning [45/100]

In this task, you need to design machine learning algorithm to distinguish between **normal** measurement and **FDI attack** measurement. You need to train and tune your ML detector on the `train_dataset`. Finally, report your predict label for the measurement in `test_feature`.

1. What is the learning task (regression or classification) in machine learning? [2/100]

2. Briefly discuss what are the roles of train-validation split in a machine learning task in your own words. [2/100]

3. Define what is F1 score. [1/100]

4. To visualize the high-dimensional data, dimension reduction techniques, such as principal component analysis (PCA), can be used. Use Matlab's `pca` or Python's `sklearn.decomposition.PCA` to project the class 0 and class 1 measurements in `train_dataset` into 2-dimension at the same time. Answer the question below [5/100],

   - Briefly discuss the idea behind PCA.
   - Then draw scatter plot to visualize them (make sure you include the figure in your report!).
   - Discuss what you have found and how the findings can guide you design the machine learning algorithms.

5. Using **different** machine learning techniques to train a detector to classify the normal measurement and FDI attacks on the `train_dataset`. [15/100]

   Note that

   - There is no restrictions on what machine learning techniques you use. It is suggested to start from the classic ones that are covered in the lecture, such as linear support vector machine, nonlinear (kernel) support vector machine, decision trees, etc. It is also recommended to use Matlab's build-in functions or Python's scikit-learn package.
   - Briefly describe how you develop your machine learning models and how you choose the hyperparameters.
   - Report the performance of your classifier in the format of TPR, FPR, and F1 score, on the `train_dataset`. Compare different methods and discuss any findings you have.

6. Report your detection results on the `test_feature`. Note that you can only report the results from **one** detector you believe to have the best performance. You must report it in `group_no.npy` format. Use the same convention, i.e., class 0 for normal measurement and class 1 for FDI attack measurement. [15/100]

   Note that

   - Using Group 1 as an example, you are reporting a list of 0 or 1 in `group_1.npy`.
   - Your sequence of the results should match the sequence of `test_feature` assigned to you.
   - Your results will be assessed by the F1 scored.

   For Python user, you can save your prediction by (for reference only)

   ```
   np.save(f"group_{group_index}.npy", y_pred)
   ```

   To make sure that your submission can be assessed, use the following code to see if you have a legit result (for reference only)

   ```
   def assess(y_pred):
       assert np.all((y_pred == 0) | (y_pred == 1))
       assert len(y_pred.shape) == 1
       assert y_pred.shape[0] == 1200
   ```

7. Use one paragraph to conclude what you have found in this coursework. [5/100]