Smart Grid Stability Detection via Classification

Batuhan Hangun
Independent Researcher
İstanbul, Turkey
batuhanhangun@gmail.com

Onder Eyecioglu
Computer Engineering Department
Bolu Abant İzzet Baysal Üniversitesi
Bolu, Turkey
onder.eyecioglu@ibu.edu.tr

Murat Beken
Computer Engineering Department
Bolu Abant İzzet Baysal Üniversitesi
Bolu, Turkey
murat.beken@ibu.edu.tr

***Abstract*—Smart grid stability is one of the most important factors that can be used as a criterion for assessing the usability of smart grid architecture, so testing and predicting stability under various circumstances hold great importance. As a result of the increase in residential and industrial structures, some intelligent solutions to predict stability to prevent unwanted instabilities in a future smart grid architecture are needed. In this study, we used various machine learning methods to predict smart grid stability. We approached the problem as a classification problem, we used a 4-node architecture smart grid dataset, and applied some well-known classification methods to classify the dataset into two classes which are "stable" and "unstable". For the classification part, we used k-Nearest Neighbour (kNN), neural networks (NN), support vector machine (SVM), and decision tree. All four methods were tested under different hyper parameters. Finally, the results were reported.**

***Index Terms*—smart grid, stability, machine learning, classification**

I. Introduction

To be stable, electrical networks require a balance between power supply and demand. Traditional systems achieve this balance by producing power based on demand. As a result of increasing interest in renewable energy, those traditional systems become insufficient to supply the demand. A new control method named Decentral Smart Grid Control (DSGC) was proposed by Schäfer et al in 2015 [1]. Although it was a good solution, it also requires an important inspection to detect instabilities in the system [2]. There are studies that aimed to detect and/or fix the instabilities in DSGC systems. One of the example of them is the study by Arzamasov et al. developed a novel approach for implementing demand response that would not require major changes to current infrastructure. In that study, decision tree was used to classify stable and unstable grid architectures [3]. In a more recent study, Breviglieri et al. used deep learning models to classify a simulated data set. Their proposed model has 99% accuracy [4]. In this study, we tried to detect instabilities in a simulated data set of 4-node architecture by using four different machine learning methods which are SVM, NN, kNN, and decision tree. Experiments show that we achieved 86% to 96% accuracy by using mentioned methods. The rest of this study is organized as follows. In Chapter II, we briefly explained the data set we used. Details of the method we used, and success rate evaluation is given in Chapter III and Chapter IV. Finally, in chapter Chapter V, we discuss the results

II. Data Set

The data set named *Electrical Grid Stability Simulated Data Data Set* used in this study was taken from UCI's database and it contains instances of the 4-node smart grid architecture shown in Fig. 1 Each instance consists of one producer node and three consumer nodes [5]. Attributes given in the data set are:
τ: Reaction time (τ_1: Producer)
p Nominal power consumption/production
g: Coefficient related to price elasticity
$stab$: Characteristic that defines if a system is stable or unstable ($stab < 0$: stable)
$stabf$: Categorical label ('stable' or 'unstable')

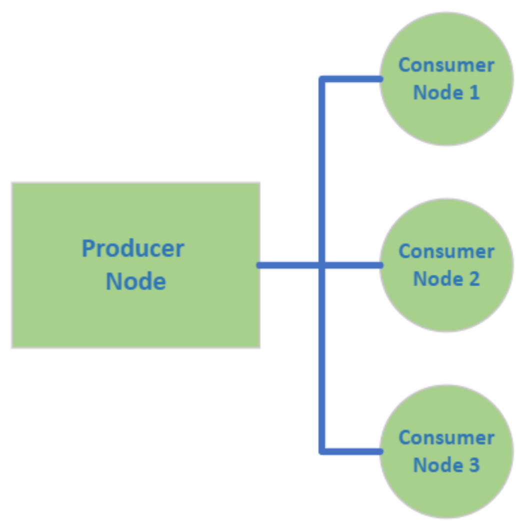


Fig. 1: 4-node architecture

III. Method

A. Classification Approach

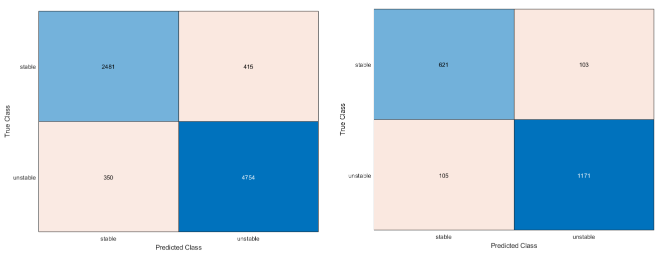
Since there are two different possible states for each node-set, it is possible to classify the data set into two separate classes and see if it satisfies the pre-defined labels. For classification, we used kNN, neural networks, SVM, and decision tree.

B. Training and Tests

The data set consists of 10000 instances. We split it into the train and the test by 80% and 20% respectively. We used different types of SVM, NN, kNN, and decision tree classifiers but only the ones with satisfactory results were reported. The most successful SVM was Cubic SVM with an automatic kernel scale. The most successful NN was a wide NN with ReLU as an activation layer. The most successful decision tree was a bagged decision tree and lastly, the most successful kNN classifier was 10 neighbors, equally distance weighted a kNN classifier.

IV. Results

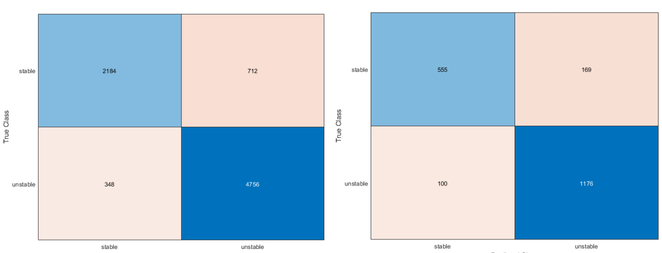
After evaluating validation and test, we obtained confusion matrix for each method. Results were given by Figs. 2, 3, 4 and 5. At decision tree classifier, out of 8000 training data, 2481 were correctly classified as *stable* and 4754 were correctly classified as *unstable*.



(a) Validation (b) Test

Fig. 2: Confusion matrix for decision tree

At kNN classifier, out of 8000 training data, 2184 were correctly classified as *stable* and 4756 were correctly classified as *unstable*.



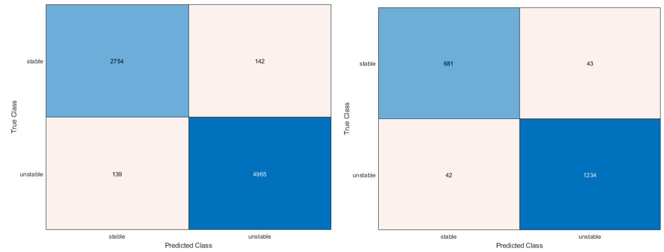
(a) Validation (b) Test

Fig. 3: Confusion matrix for kNN

At SVM classifier, out of 8000 training data, 2184 were correctly classified as *stable* and 4756 were correctly classified as *unstable*.

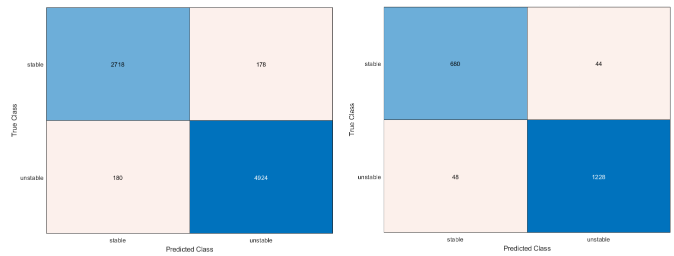
At NN classifier, out of 8000 training data, 2718 were correctly classified as *stable* and 4924 were correctly classified as *unstable*.

Results of validation and test steps were given by Table I and II. It's seen that the SVM gives the best results for both validation and test steps in terms of accuracy, precision, recall



(a) Validation (b) Test

Fig. 4: Confusion matrix for SVM



(a) Validation (b) Test

Fig. 5: Confusion matrix for NN

and F1-score, it is followed by NN and the least successful of them is kNN.

TABLE I: Classification results for validation

	Validation			
Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	96.4875	95.20	95.10	95.15
Neural Network	95.525	93.79	93.85	93.82
Decision Tree	90.4375	87.64	85.67	86.64
kNN	86.75	86.26	75.41	80.47

TABLE II: Classification results for test

	Validation			
Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	95.75	94.13	94.19	94.06
Neural Network	95.40	93.66	93.41	93.92
Decision Tree	89.60	85.66	85.54	85.77
kNN	86.55	80.49	84.73	76.66

V. Conclusion

In this study, we used four different machine learning approach to solve a classification problem based on smart grid stability. The quantitative comparison presented gave the accuracy, recall, precision, and F1 scores for each approach. We had an over 86% accuracy rate for both validation and test phases. Results showed that SVM and NN are better at successfully classifying the data set into two predefined classes that are *stable* and *unstable*.

References

[1] B. Schäfer, M. Matthiae, M. Timme, and D. Witthaut, "Decentral smart grid control," *New journal of physics*, vol. 17, no. 1, p. 015002, 2015.

[2] B. Schäfer, C. Grabow, S. Auer, J. Kurths, D. Witthaut, and M. Timme, "Taming instabilities in power grid networks by decentralized control," *The European Physical Journal Special Topics*, vol. 225, no. 3, pp. 569–582, 2016.

[3] V. Arzamasov, K. Böhm, and P. Jochem, "Towards concise models of grid stability," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, IEEE, 2018.

[4] P. Breviglieri, T. Erdem, and S. Eken, "Predicting smart grid stability with optimized deep models," *SN Computer Science*, vol. 2, no. 2, pp. 1–12, 2021.

[5] V. Arzamasov, "Electrical Grid Stability Simulated Data ." https://archive-beta.ics.uci.edu/ml/datasets/electrical+grid+stability+simulated+data, 2018.