Analysis of Machine Learning Algorithms for Classifying Electrical Grid Stability

Aaron Ogle[a,c] (Researcher)

ARTICLE INFO

Keywords:
Machine Learning
Artificial Neural Network
K-Nearest Neighbor
Logistic Regression
Smart Grids

ABSTRACT

The power grid of today is vastly different than the power grid of previous generations. With renewable energy sources joining the power grid, maintaining the stability of a power grid is a very challenging problem that is of crucial importance. When a power grid becomes unstable then blackouts can occur which can have huge negative impacts. Utilizing the data set that the authors in (1) have provided, we will apply several different types of machine learning algorithms on the data in order to classify whether the grid is stable or unstable. We will compare the Accuracy, F1-Score, Precision, and Recall, and see which one performs the best.

1. Introduction

The authors in (1) and (2) conduct research on creating models of Decentralized Smart Grid Control (DSGC); DSGC are new systems that bind the price of electrical energy to the grid frequency, this means that consumption of electricity will change based off of changes in pricing (1). The creation of new power grid models such as the DSGC are due to the need for using more renewable energy sources; how exactly renewable energy can be integrated into the power grid in an efficient way is still being debated (2). However, due to climate change, it is critical that mankind opts for clean and renewable energy sources versus continuing to rely on and use fossil fuels.

With new power grid models being created and researched it is also important to make sure that we can detect whether the grid is stable or unstable through automation; if a grid becomes unstable then the consequences can be enormous. The data that the authors of (1) have provided consists of several features which are defined below:

- tau1-tau4: tau[x] will be the reaction time of participants in the model where tau1 is the value for the energy producer; recall that the prices will be changing in the DSGC model so some of the features are the reaction times of the participants.
- p1-p4: p[x] is the power that is consumed (negative) and the power that is produced (positive).
- g1-g4: g[x] will be the coefficient that is proportional to the elasticity of the price; where g1 is the value for the electricity producer.
- Class Label (stable or unstable) - the dataset originally has the string values of "stable" or "unstable" but we have converted the values to 1.0 for "stable" and 0.0 for "unstable".

The main contributions of this work will be as follows.

1. Conduct a literature review of related work and discuss the work that is currently being done on applying machine learning models to problems related to power grids.
2. Implement the K-Nearest Neighbors, Logistic Regression, and Artificial Neural Network models on the data set described above and compare the results to see which one is able to classify the data more accurately. Metrics such as Accuracy, F1-Score, Precision, and Recall will be used in order to measure efficiency.
3. Find an optimal number of neurons for the hidden layer of the ANN to achieve higher accuracy with classifying the data.
4. Find an optimal value for K in the KNN algorithm to increase the accuracy of the algorithm.

This paper will be divided as follows. Section II will consist of Related Work, Section III will contain our Model and our exact Problem Statement, Section IV will contain the results of our experiments, and in Section V we will conclude our findings and discuss potential avenues for future research. The final section will contain the relevant references.

ORCID(s): 0000-0000-0000-0000 (A. Ogle)

2. Related Work

The authors of (1) work on constructing a Decentral Smart Grid Control model by utilizing mathematical modelling techniques along with using data mining and decision trees. In the work done in (1) they address the problem of previous researchers assuming that everyone that is participating in the electrical grid all behave in the same way. The authors are able to utilize decision trees in order to discover new insights about behavior in the grid such as having a stable system even if some participants in the model adapt the way they are consuming energy with a high delay (1). The author's mathematical model of a Decentral Smart Grid Control assumes that there is just one energy producer and three consumers; they note that they do not make any statement related to how their model will scale, nor do they know if their model would be suitable for a large country (1). This is definitely a limitation of their work, but the data they are able to generate at least gives us a starting point for applying machine learning models on power grid related data.

In (2) the authors also utilize mathematical modelling techniques in order to construct a DSGC model. The authors find in their work that if the frequency measurements of energy consumption in a DSGC are averaged sufficiently over a large time interval then the DSGC enhances the stability of a power grid system (2). In the work, the authors also utilize circle network shapes, lattice network shapes, and a four node star motif to represent the shapes of electrical grid networks (2). In the four node star motif model, this means that their DSGC model is looking at just one producer, the center of the star, and then three consumers; similar to (1), with such a simplified grid, it is hard to determine how well their methods will scale with larger sized electrical grids.

Now that we have looked into papers that construct DSGC models, let us turn our attention to research that is being done on applying machine learning techniques to power grids and smart grids. From (3), power grids can be transformed into smart grids by enhancing it with information, communication, and machine intelligence technologies. The authors in (3) review several different applications of machine learning approaches that try to enhance factors such as stability and reliability in a smart grid (3). The authors discuss that power grid operators utilize forecasting of electricity demand in order to adjust how much power is being generated so that they can meet the demand of customers, but also make sure that the power system does not become overloaded (3). In order to help with this kind of energy dispatching problem, the authors discuss work that has been done with applying reinforcement learning based energy management algorithms that can make optimal decisions in order to reduce the cost of energy for consumers (3). A Q-learning method was developed which learns to make better energy dispatching decisions through experiences; experiments utilizing this method showed that the algorithm can reduce total energy cost based only on current information (3).

In (4), the authors use three different machine learning models, support vector machine, random forest, and artificial neural network, in order to estimate the stability of power grid synchronization. The authors train their models using synthetic data that they have created and find that their models can transfer their knowledge and be applied to real data from power grids in Great Britain, Spain, France, and Germany in order to predict the stability of the power grids. Just as in this work, the authors in (4) utilize the ReLU activation function in the hidden layers of their artificial neural network. The authors compare their results on using both heterogeneous and homogeneous data; by this we mean what kind of power distribution model the algorithms used (4). The author's models perform well when it comes to classifying the data which shows that their models were able to learn from synthetic data and were then able to transfer that knowledge to real power grid data.

The authors in (5) implement a multi-directional long short-term memory model for predicting electrical grid stability. In their work, the authors use the same data set that were are utilizing for our work. The main contribution that is made in their paper is a novel multi-directional long short-term memory model that is able to classify electrical grid stability data with a very high accuracy. The author's note that recurrent neural networks use more network layers, which provides challenges on gaining information about the parameters in previous layers; to address this, long short term memory models are used (5). LSTM models have a chain structure that is similar to RNN and has multiple neural network modules (5). With their model, the authors are able to achieve 99.07% accuracy when it came to classifying the data which was 3% higher than the other traditional deep learning algorithms that they compared their work to such as a recurrent neural network model (5).

Voltage stability prediction by using active machine learning is researched in (6); the authors main contributions are a pool based active learning model that uses power system measurements to be used to classify voltage stability (6). By active learning, the authors mean that the technique builds knowledge bases instead of having to use an exhaustive simulation method; the proposed model will search for operating points where inaccurate predictions might occur, and then it will perform simulations that will help with creating new mappings for the data and will then add that data to an existing pool that contains the training set data, this will help find behavior that was not originally represented in the training set data (6). Discussing this further, the authors utilize a pool based active learning model that uses ANN, SVM, and Decision Trees, where these usual machine learning models are employed in a "active learning" methodology; the active learning part of these models consists of having a pool of unlabeled data be labeled by an "oracle" which is able to label the data by using simulations, once the data is labeled then the ANN, SVM, and Decision Tree models can utilize the data for classification (6). From their work, the authors found that a pool based active learning approach can help with building data sets that will allow for machine learning models to be trained more efficiently; their approach also enhances the models by finding operating points where the model's classification contradicts with reality, and adds labeled data sets around these points to the knowledge base with the aim that this will help with the model's accuracy (6).

3. Models and Problem Statement

The main problems we are trying to address in this paper are as follows; out of the ANN, KNN, and Logistic Regression Machine Learning Models, which one is able to classify the electrical grid stability data with the highest accuracy. We also want to experiment with different values for K in the KNN algorithm to see which value leads to a higher classification accuracy. Then, with the ANN model, we want to find an optimal number of neurons for the hidden layer so that we can obtain a high classification accuracy.

Before we use the models on the data, we first standardize the data which consists of transforming each dimension of the data using the below equation:

$$F(v) = \frac{(v - mean)}{std}$$

where std means standard deviation. It is worth noting that using other normalization methods could lead to higher accuracy and lower accuracy for some of the models, but in order to keep things "fair" we choose to standardize the data and have each of the models use that data in order to get a better comparison. The data is also split into groups of 80% for training the model and 20% for testing the model.

In our work, we have implemented the KNN and Logistic Regression algorithms from "scratch" and we have also implemented them utilizing the scikit-learn library. We will discuss the accuracy results of our KNN and Logistic Regression implementations vs the scikit-learn implementation of them; our version of the models will be denoted and discussed as v1 and the scikit-learn version will be denoted as v2. For the ANN, we have used the Keras library in order to implement the algorithm, and along with investigating which number of neurons for the hidden layer leads to better classification accuracy, we will also investigate which activation function, Sigmoid vs ReLU (Rectified Linear Unit) leads to higher accuracy results.

For our metrics, we use the below equations which can be found in (5) as well:

$$Accuracy = \frac{(TruePositive + TrueNegative)}{TotalInstances}$$

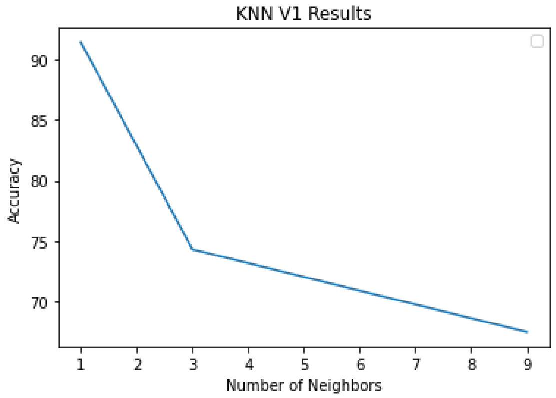
$$Precision = \frac{TruePositive}{(PredictedInstances = True)}$$

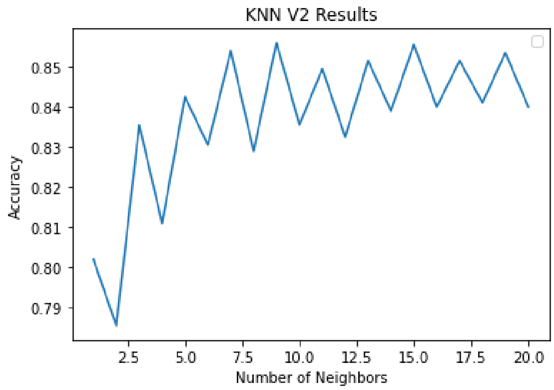
$$Recall = \frac{TruePositive}{ActualNumberofinstancesasTrue}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

4. Experimental Analysis

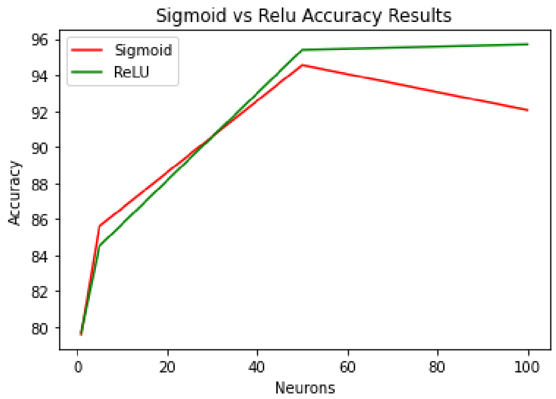
For the first part of our experimental analysis, we have implemented our own version of the KNN algorithm and will compare its accuracy with the scikit-learn implementation of the algorithm. When utilizing our implementation, v1, we found that when K=1, we obtained the highest accuracy of 91.45%, whereas with the scikit-learn version of the algorithm, v2, we found that K=9 gave higher accuracy results than other neighbors and obtained an accuracy of 85.6%. For the v2 version we used values from 1-20 for our number of neighbors. The plots on the next page show the results we obtained. Interestingly, for our implementation of the algorithm, the accuracy drops as the number of neighbors increases, but for the scikit learn version, the accuracy increases for odd numbers of neighbors until we reach 9; after this point, the algorithm dips in accuracy. We will discuss the results for F1-Score, Precision, and Recall at the end of our experimental analysis when we compare these values for all three algorithms together.





For our implementation of the Logistic Regression algorithm, unfortunately, we are only able to obtain an accuracy of 68.55%; we are not sure what the problem is with our model, and after trying different things we were unable to increase the accuracy of the model. It may just be that Logistic Regression is just not the best algorithm to use for classifying this data, or different normalization techniques should be utilized on the data in order to increase accuracy, however, we want to utilize standardization since that is what the other models will be using. The scikit learn version is able to obtain an accuracy of 80.05%.

For our Artificial Neural Network, we use the following numbers for the number of neurons in the hidden layer: 1, 5, 50, 100. We utilize the mean square error as our loss function and then we will compare the results of using the Sigmoid activation function vs the ReLU activation function in the hidden layers. The results that were obtained are plotted below.



From our results we see that for the model that uses the Sigmoid activation function the accuracy increases with each addition of more neurons in the hidden layer until we utilize 100 neurons; the accuracy then starts to drop. For the model that uses the ReLU activation function the model continues to increase as more neurons are added to its hidden layer. Our best results that we were able to obtain is an accuracy of 95.7% when utilizing the ReLU activation function in the hidden layer and also using 100 neurons in this layer. Due to this being the highest accuracy, we will use these same settings for when we discuss the values for the overall comparison between the algorithms.

Now that we have investigated the accuracy of the algorithms, an ideal value for K in our KNN algorithm, and an optimal value for the number of neurons in the hidden layer of our ANN, let us look into the F1-Score, Precision, and Recall of the algorithms by utilizing scikit learn and keras functionality. The below table shows our percentage results for Accuracy, F1-Score, Precision, and Recall for each of the algorithms. For KNN, we use 9 neighbors, and for the ANN we use 100 neurons in the hidden layer with the ReLU activation function.

Algorithm	ANN	KNN	Logistic Regression
Accuracy	95.7	85.6	80.05
F1-Score	93.8	77.4	71.5
Precision	95.4	90.5	74.6
Recall	92.2	67.6	68.7

From our results we can see that the Artificial Neural Network model performed the best in all of the different metrics. The KNN came in second with the best performance, and the Logistic Regression model came in last out of the models. Although the Logistic Regression model came in last, it did perform slightly better when it came to recall that the KNN algorithm did, however, all of the other metrics were lower for the Logistic Regression model. The ANN was able to obtain a high classification accuracy of 95.7%; if a decentralized smart grid control model were to be implemented, and a process was needed for a machine learning model to monitor whether or not the grid was stable or unstable, the ANN would be a good candidate model for this task.

5. Conclusion and Future Work

The stability of electrical grids is of crucial importance, and with additional renewable energy resources being added to the grid it is becoming an increasingly difficult problem to maintain. From this work, we have seen that machine learning techniques can be utilized in order to predict whether or not the grid is stable based off of different features.

In our work, we have investigated three different machine learning algorithms and have found that the Artificial Neural Network model performs the best out of the three with the highest accuracy classification percentage of 95.7%. We investigated different numbers of neurons to input into the hidden layer and found that the optimal setting was having 100 neurons in the hidden layer with the ReLU activation function. We also investigated the KNN algorithm to see what an ideal value for "k" would be in order to acheive the highest accuracy percentage. From our implementation of the algorithm, we found that when k=1 we acheived the highest accuracy percentage, but when utilizing the scikit learn library we found that the optimal number of neighbors is 9.

Unfortunately, we were unable to implement a Logistic Regression model that was able to classify the data efficiently, however the scikit learn library version of the algorithm was able to have a decent classification accuracy of 80.05%. With this model having the lowest accuracy of the other models it may just not be the best choice to use on this data set.

For future work, we would like to investigate deep learning techniques such as the ones used in (5) to see if by adding additional layers to our ANN model if we can have an increase in the classification accuracy. We would also like to learn more about, and implement, other machine learning algorithms such as the support vector machine and random forest models to see if either of these methods are able to achieve a higher classification accuracy.

6. Bibliography

References

[1] V. Arzamasov, K. Böhm and P. Jochem, "Towards Concise Models of Grid Stability," 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2018, pp. 1-6, doi: 10.1109/SmartGridComm.2018.8587498.

[2] Schäfer, Benjamin, et al. "Taming instabilities in power grid networks by decentralized control." The European Physical Journal Special Topics 225.3 (2016): 569-582.

[3] S. Azad, F. Sabrina and S. Wasimi, "Transformation of Smart Grid using Machine Learning," 2019 29th Australasian Universities Power Engineering Conference (AUPEC), 2019, pp. 1-6, doi: 10.1109/AUPEC48547.2019.211809.

[4] Yang, Seong-Gyu, et al. "Power-grid stability predictions using transferable machine learning." Chaos: An Interdisciplinary Journal of Nonlinear Science 31.12 (2021): 123127.

[5] M. Alazab, S. Khan, S. S. R. Krishnan, Q. -V. Pham, M. P. K. Reddy and T. R. Gadekallu, "A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid," in IEEE Access, vol. 8, pp. 85454-85463, 2020, doi: 10.1109/ACCESS.2020.2991067.

[6] V. Malbasa, C. Zheng, P. Chen, T. Popovic and M. Kezunovic, "Voltage Stability Prediction Using Active Machine Learning," in IEEE Transactions on Smart Grid, vol. 8, no. 6, pp. 3117-3124, Nov. 2017, doi: 10.1109/TSG.2017.2693394.