Feature Studies for PV Fault Classification Using Nonlinear Principal Component Analysis

Maxwell Yarter
School of ECEE
Arizona State University
yartermt@gmail.com

Gowtham Muniraju
School of ECEE
Arizona State University
gmuniraj@asu.edu

Andreas Spanias
School of ECEE
Arizona State University
spanias@asu.edu

Yiannis Tofis
KIOS center
University of Cyprus
tofis.n.yiannis@ucy.ac.cy

***Abstract*—Solar fault classification neural networks have shown promise in recognizing fault conditions and improving power output of solar arrays in conjunction with Smart Monitoring Devices (SMD). These SMD's contain sensors for recording** 10 **solar features on a solar panel and apply dynamic switching to maximize power output under a given fault. [1] The sensor data can be used to classify the common solar array faults soiling, arc faults, degraded modules, and shading. In this study solar feature data from is manipulated using kernel principal component analysis (KPCA) and autoencoder neural networks to reduce its dimension. The modified data is used to train fault classification neural networks with various dimensions of input features to compare classification accuracy. Through this process it was determined that the feature set had a significant amount of redundancy, but no nonlinear relationships that could be manipulated to improve fault classification accuracy.**

***Index Terms*—Feature analysis, Nonlinear PCA, Neural networks, Machine Learning, PV modules, Kernel PCA, Autoencoder.**

I. Introduction

Solar energy has tremendous potential to alleviate global energy insecurity as a source of clean renewable electricity [2]. Developing solar infrastructure comes with a host of problems related to maintenance and consistency that need to be overcome to ensure reliability. Solar arrays are subject to several common fault conditions that reduce their power output and require a technician to identify and solve in order to restore peak operating conditions. These faults range from soiled panels that require cleaning to arc faults that can be hazardous to technicians and the array. [3] It would be more cost effective and efficient to remotely monitor the arrays status and autonomously detect faults than manually inspecting an array for fault conditions. This type of autonomous detection has been successfully implemented using a combination of Smart Monitoring Devices (SMD) and an Artificial Neural Network (ANN) for classification [1].

Previous work in SenSIP lab addressed several problems in solar array monitoring, control and optimization [4]–[13]. Initial work was reported in [4] where traditional statistical methods were proposed. Later machine learning methods [5] were considered including PU Learning [6]. Fault detection using neural nets was reported in [7], [8] and optimization methods were reported in [9]. PU learning for fault detection was reported in [10] and a recent study including neural net fault detection experiments and simulations on a quantum computer simulator was published in [11]. Training neural networks on quantum computer simulators is complicated, time consuming, and the difficulty scales with the number of features and size of the data set being used. A major impetus for this project was alleviating these problems through dimensional reduction of the training data set.

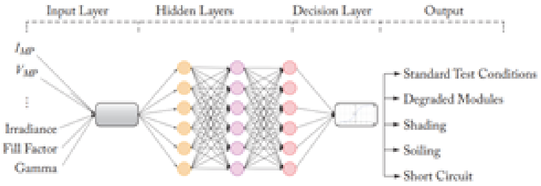

Fig. 1. Solar Fault Classification Neural Network [1].

This autonomous fault detection system uses 10 features relevant to solar power production such as voltage, current, irradiance, and temperature. Initial attempts at redesigning the ANN to use fewer input features resulted in significant reduction of the networks fault classification accuracy. It would be desirable to reduce the number of features the model is dependent on for computational efficiency, and to reduce the number of sensors needed for each SMD.

One method of dimensional reduction is principal component analysis (PCA). Principal component analysis allows us to determine the importance of variables and the correlation between variables [14]. More specifically, if there are unique first degree relationships between variables in a system then, PCA produces a condensed set of variables where the variable with greatest variance is the most important to the system. [15] Traditional PCA is an inherently linear method of data analysis and fails in the face of systems with nonlinear feature sets. PCA can be elaborated on by using Nonlinear principal component analysis (NLPCA) techniques. The advantage of NLPCA is revealing if a data set exists in a nonlinear manifold that can be made linear. [16].

In this paper we focus on two methods of NLPCA to reduce the dimension of training required to achieve high fault classification accuracy. Additionally, we attempt to increase ANN fault classification accuracy beyond a roughly 90 percent threshold previously established [1]. The first method of NLPCA is Kernel Principal Component Analysis (KPCA).

Fig. 2. SenSIP Solar Testbed at ASU Research Park

KPCA can unravel nonlinear relationships in a feature set by passing the feature vectors through a selected kernel function [17]. This method alters the original feature space and compresses it to a selected number of dimensions. An advantage of this method is that, in nonlinear systems, it can help separate the labeled data allowing for a more accurate decision boundary.

The second NLPCA method explored in this paper is the autoencoder neural network configuration. The autoencoder neural network consists of a hidden layer with three distinct components which will be referred to as the encoding, bottleneck, and decoding layer. The objective of this network architecture is to recreate input feature data at the network output after compressing the features in the bottleneck layer [16]. The extent of the compression or dimension of features in the compressed space is determined by the number of neurons in the bottleneck layer. By applying reinforcement scoring to the network output the network is forced to learn the most important aspects of a data set and represent them through its encoding.

II. NREL Data Set

The labeled solar data set used in this research was recorded by the National Renewable Energy Laboratory [18]. The data set consists of ten features and five labels. The ten features are DC power, maximum voltage, maximum current, temperature, irradiance, fill factor, gamma ratio, maximum power, open-circuit voltage, and short-circuit current. The five labels consist of a standard test condition (STC) and four faults; degraded, shaded, soiled, and short circuit.

Data was recorded at hour intervals over the course of one year. The standard test condition data is defined by any given days maximal power output corresponding to the days temperature and irradiance conditions. Shaded faults were labeled as data points with reduced irradiance values while soiled modules were classified by standard irradiance measurements with reduced power output. In other words, if irradiance was at the standard test condition level, but the power output was low then it was considered a soiled fault. A degraded module was classified by reduced open-circuit voltage or reduced short-circuit current measurements [3].

III. Kernel PCA

Kernel PCA is conducted by passing a data set through a kernel function to map it in a higher-dimensional space [19]. PCA is then used on the modified data set to assign weight to the modified features. [17]. The data set was, separately, passed through five different kernel functions: polynomial, cosine, radial bias function, sigmoid, and linear. The linear kernel is identical to standard PCA and can be used as a baseline for determining the other kernel function's ability to capture nonlinear structures in the data set. After generating a modified feature space using the kernel function, we have a full set of features and labels that can be used to train a fault classification neural network.

Kernel PCA searches the data set for nonlinear structures but must still be applied to a fault classification network to determine the optimal number of principal components. Principal components is used here to refer to the number of KPCA variables being used to train classification networks. Fault classification networks were trained using 2-10 of the kpca generated features for each kernel function to observe the changes in classification accuracy with increasing principal components.

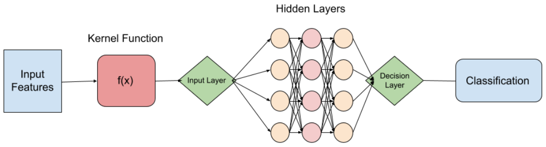

Fig. 3. KPCA Schematic Diagram.

IV. Autoencoder Neural Networks

Autoencoder neural networks operate by compressing a feature space into a specified number of neurons then expanding back to the original size. This compression scheme represents a non-linear generalization of PCA that captures high-order correlations between the features in the input layer [20]. If the autoencoder is successful at replicating feature data, then it was able encode the full feature set into a smaller one. A successful compression implies that the encoder has learned the non-linear structure of the features and that there is enough redundancy in the data set to represent all relevant information in a condensed form. Once trained, passing solar feature data into the network creates an encoded space in the bottleneck. The data output from these bottleneck neurons becomes our principal components. This means we can specify any number of principal components less or greater than the dimension of the original feature space. These principal components generated by the autoencoder are then used to train the solar fault classification neural network. As with the KPCA method 2-10 principal components were used as training data to compare classification accuracy. Additionally, several autoencoders were given more than 10 bottleneck neurons to expand the feature set beyond 10 dimensions. These larger feature spaces were also used to train the fault classification network.

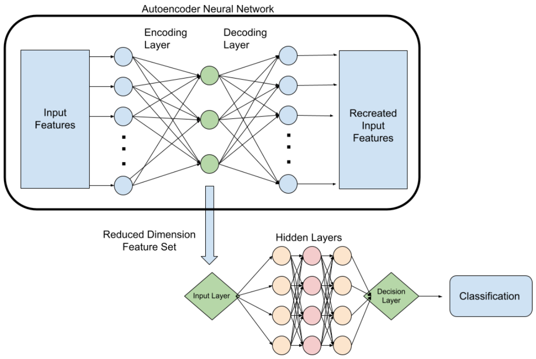

Fig. 4. Autoencoder Schematic Diagram.

V. Results

The use of kernel principal component analysis shows that the linear kernel function produces the most accurate classification with this data set. Additionally, the network achieved maximum accuracy using only 6 principal components and saw diminishing returns on accuracy after 4 principal components. The cosine and sigmoid kernels showed promise and generally saw an increase in accuracy with increasing principal components, but accuracy was less than that of the linear by several percent. The radial bias function and polynomial networks achieved 80 percent classification accuracy with 4 components but became less accurate when increasing the number of features.

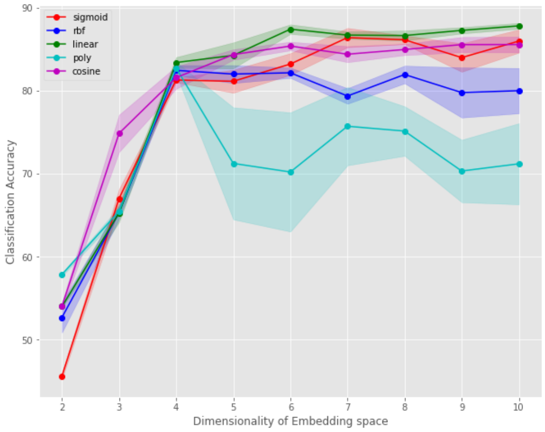

Fig. 5. KPCA Average Accuracy

The most telling result of the KPCA is that the ”linear” kernel function, a regular PCA, was most successful in dimensional reduction. This indicates that the original feature space was not composed of nonlinear relationships. As a result, none of the kernel functions were able to produce a feature space that enabled greater than 90 percent classification accuracy. The neural networks trained with KPCA generated features were able to obtain high accuracy using only 4 components demonstrating a large amount of redundancy in the data set. Additionally, the largest limitation to increasing classification accuracy appears to be the ”shaded” and ”STC” fault classifications. The confusion matrix for a 5-feature linear KPCA shows the majority of misclassification is between these two classes.

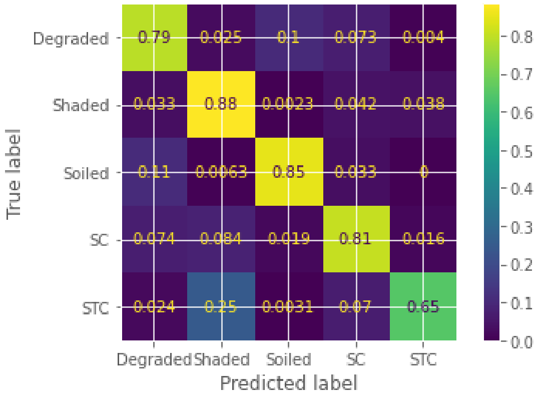

Fig. 6. 5-Feature Confusion Matrix Showing Classification Percent Accuracy.

The overlap between these two classes is so prevalent that it can be seen in a 2D plot of any combination of 2 features from the original data set. To improve classification accuracy there would need to be an addition of features to these classes. Another solution to misclassification would be to redefine the shaded class to separate the two classes more. Irradiance is the feature directly afflicted by a shaded fault. The most reasonable adjustment to the shaded fault definition would be to reduce the threshold of irradiance from less than 75 percent [3] of the daily maximum to a lower percentage.

The Autoencoder neural network approached 80 percent accuracy but suffered from sudden drops in accuracy at greater than 4 principal components. Doing multiple Monte Carlo trainings to obtain average accuracy produced drops in average accuracy at different numbers of principal components. This inconsistency among trials, and lower accuracy than the KPCA trials, indicates several things. When autoencoders are given too much space to operate in they can learn to copy input data instead of learning relevant features [21]. This would explain why the large drops in accuracy seem to occur when the network is tasked with creating more than 4 principal components. The accuracy being lower than PCA methods is a more difficult problem to address but is likely due to a lack in volume of training data to sufficiently train the autoencoder.

A similar result was obtained by expanding the bottleneck layer to contain more features than the original data set. Classification accuracy was inconsistent and generally sat at the 80 percent accuracy mark.

The autoencoder suffers from a similar inability to classify between the STC and shaded classes. This is visible in a 3D representation of the shaded and STC feature spaces using both the autoencoder and KPCA. The autoencoder result was consistent with that of the KPCA in that there is diminishing returns on accuracy when exceeding 4 principal components.

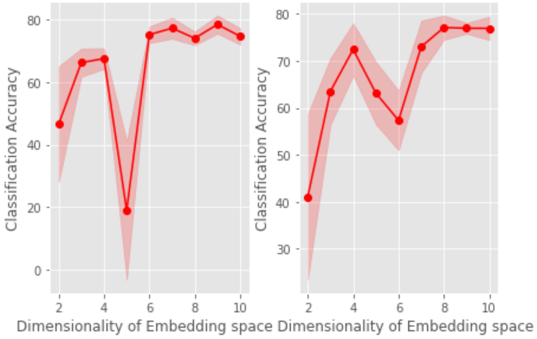

Fig. 7. Two Examples of Autoencoder Average Accuracy.

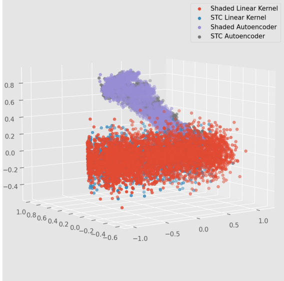

Fig. 8. Shaded and STC Overlap Using KPCA and Autoencoder Features.

VI. Conclusion

In conclusion, this study utilized nonlinear principal component analysis techniques to develop a better understanding of the features in the NREL solar data set. Both KPCA and autoencoder neural networks were able to show redundancy in the feature set. In each case the fault classification networks trained with modified features saw an ”elbow” in their accuracy curve at 4 principal components. This demonstrates a large amount of redundancy in the feature set’s ability to classify solar faults. Perhaps the most relevant conclusion of this study is that the feature set does not have a set of nonlinear redundancies that can be analyzed to improve fault classification accuracy. Linear principal component analysis was sufficient in reducing the feature set dimension, and was the most accurate method overall. The autoencoder was the least accurate method of dimensional reduction likely because there wasn’t enough training data or features for the network to successfully learn the features important qualities.

Based on this information, improving the solar fault classification network can be achieved by using the 4 most relevant principal components in combination with a solution to the STC and shaded fault misclassification. This could be in the form of new solar features in addition to the top four already in use, or by redefinition of the fault conditions.

VII. Acknowledgements

This work was supported in part by NSF Grant No. 1854273. Special thanks to Dr. Gowtham Muniraju and Dr. Andreas Spanias for their instruction and guidance that enabled this project.

References

[1] S. Rao, S. Katoch, V. Narayanaswamy, G. Muniraju, C. Tepedelenlioglu, A. Spanias, P. Turaga, R. Ayyanar, and D. Srinivasan, “Machine learning for solar array monitoring, optimization, and control,” *Synthesis Lectures on Power Electronics*, vol. 7, no. 1, pp. 1–91, 2020.

[2] E. Kabir, P. Kumar, S. Kumar, A. A. Adelodun, and K.-H. Kim, “Solar energy: Potential and future prospects,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 894–900, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117313485

[3] R. Platon, J. Martel, N. Woodruff, and T. Y. Chau, “Online fault detection in pv systems,” *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1200–1207, 2015.

[4] H. Braun, S. Buddha, V. Krishnan, C. Tepedelenlioglu, A. Spanias, S.i Takada, T. Takehara, M. Banavar, and T. Yeider., Signal Processing for Solar Array Monitoring, Fault Detection, and Optimization, Synthesis Lectures on Power Electronics, Morgan Claypool, Book, 1-111 pages, ISBN 978-1608459483, Sep. 2012.

[5] U. Shanthamallu, A. Spanias, C. Tepedelenlioglu, M. Stanley, ”A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications,” Proceedings 8th International Conference on Information, Intelligence, Systems and Applications (IEEE IISA 2017), Larnaca, August 2017.

[6] K. Jaskie and A. Spanias, ”Positive and Unlabeled Learning Algorithms and Applications: A Survey,” Proc. IEEE IISA 2019, Patras, July 2019.

[7] S. Rao, A. Spanias, C. Tepedelenliglu, ”Solar Array Fault Detection using Neural Networks”, IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Taipei, May 2019..

[8] S. Rao, G. Muniraju, C. Tepedelenlioglu, D. Srinivasan, G. Tamizhmani and A. Spanias, ”Dropout and Pruned Neural Networks for Fault Classification in Photovoltaic Arrays, IEEE Access, 2021.

[9] V. Narayanaswamy, R. Ayyanar, A. Spanias, C. Tepedelenlioglu, ”Connection Topology Optimization in PV Arrays using Neural Networks’,” IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Taipei, May 2019.

[10] K. Jaskie, J. Martin, and A. Spanias, “PV Fault Detection using Positive Unlabeled Learning,” Applied Sciences, vol. 11, Jun. 2021.

[11] G. Uehara, S. Rao, M. Dobson, C. Tepedelenlioglu and A. Spanias, ”Quantum Neural Network Parameter Estimation for Photovoltaic Fault,” Proc. IEEE IISA 2021, July 2021.

[12] H. Braun, S. T. Buddha, V. Krishnan, C. Tepedelenlioglu, A. Spanias, M. Banavar, and D. Srinivansan, “Topology reconfiguration for optimization of photovoltaic array output,” Elsevier Sustainable Energy, Grids and Networks (SEGAN), pp. 58-69, Vol. 6, June 2016.

[13] G. Muniraju, S. Rao, A. Spanias, C. Tepedelenlioglu, M20-254P Dropout and Pruned Neural Networks for Fault Classification in Photovoltaic Arrays Provisional US 63/039,012, 06/15/2020.

[14] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[15] K. P. F.R.S., “Liii. on lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. [Online]. Available: https://doi.org/10.1080/14786440109462720

[16] W. W. Hsieh, “Nonlinear principal component analysis by neural networks,” *Tellus A*, vol. 53, no. 5, pp. 599–615, 2001.

[17] H. Heiko, “Kernel pca for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[18] A. P. Dobos, “Pvwatts version 1 technical reference,” 2013.

[19] B. Schölkopf, A. Smola, and M. Klaus-Robert, “Kernel principal component analysis,” *Lecture Notes in Computer Science*, vol. 1327, 1997.

[20] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006. [Online]. Available: https://science.sciencemag.org/content/313/5786/504

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org