KNN Algorithm for Online Identification of Power System Network Branch Events

John Woods[1], *Student Member, IEEE*, Tim Driscoll[1], *Student Member, IEEE*
[1]Real-Time Power and Intelligent Systems Laboratory
Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA

Abstract—The Smart Grid (SG) is vulnerable to both physical and cyber-attacks. The transmission network of the power grid is where the bulk movement of energy occurs. Efficient and accurate methods for identifying network branch events is critical in the security of the grid. Phasor measurement units (PMUs) are placed around the grid to provide useful measurements and aid the process of identifying network branch events. The first step in solving a problem is to identify what the problem is, which is exactly what Network Identification Methods (NIM) are designed for. Once the problem is identified, it can then be resolved. Different neural network algorithms are presented and analyzed in this paper. Multilayer perceptron networks (MLPs) use a neural network-based learning approach which creates a hidden layer of neurons that activate based on learned events using weights for the measurements. Admittance Matrix Identification is a primitive version of MLP networks where an extremely thorough calculation is done. Convolutional Neural Networks (CNN) are like MLP but layers are partially connected instead of fully connected for more efficiency in computing. Learning vector quantization (LVQ) is a hybrid supervised and unsupervised learning classification algorithm. The K-Nearest Neighbor (KNN) algorithm is a supervised learning algorithm that can be used to solve classification problems. This paper takes a more in-depth analysis on the performance of MLP, LVQ, and KNN networks. Initial tests show that KNN is a faster network and provides higher identification accuracies. Suggested improvements for all tested network implementations are provided.

Index Terms - PMU, NIM, MLP, CNN, LVQ, KNN

I. INTRODUCTION

The power system is integrating more smart grids (SG) as the technology improves because their use of digital two-way communication makes them objectively superior to traditional power grid systems. Each part of the SG can communicate with the other parts, which means the system can adjust for changes like lower demands or even failures much more quickly. However, the new use of digital technology means the SG is vulnerable to cyber-attacks in addition to physical attacks, so the protection systems in place must also be capable of dealing with these vulnerabilities. These components, incorporating the applicable criteria that follow.

The evolution of the grid promoted the integration of new monitoring technologies, most notably phasor measurement units (PMUs). PMUs provide grid operators with voltage and current measurements that are synchronized across the grid [6]. The measurements are synchronized by time in accordance with a Global Positioning System (GPS) [4]. These measurements provide grid operators with valuable information but need neural network algorithms to optimize the use of the data and security of the grid.

Currently PMUs are located across all areas of the grid from generation to distribution. This paper specifically focuses on the application of online branch event identification within the transmission network of the grid. The transmission network of the grid is where a high volume of electricity is transmitted and if disrupted can result in large scale outages. The transmission network is often referred to as the backbone of the grid, connecting the entire body [4].

Fortunately, digital technology can inform the system of attacks by using Network Identification Methods (NIMs), which inform the system of failures or attacks by detecting abnormal network branch events. Examples of NIMs include Multilayer Perceptron (MLP) Networks, Convolutional Neural Networks (CNN), learning vector quantization (LVQ) and Admittance Matrix Identification (AMI).

II. THE IMPORTANCE OF NETWORK IDETIFICATION METHODS

A. Network Identification Introduction

NIMs are a crucial part of the SG because they are an integral part of securing the grid and analyzing the collected information. As noted previously the power grid is arguably the most important system in the world. Without the power grid it would make it impossible for most other systems to function properly. Network identification methods are extremely important because they can prevent small to large scale blackouts that could occur due to attacks on the transmission network. A smart grid is a cyber-physical system leaving it vulnerable to both physical and cyber-attacks and problems [2].

B. Protection Benefits of NIMs

Protecting the system from failures and attacks is a crucial role that NIMs play in a SG. These methods will identify an abnormality which usually corresponds to a failure or attack, so the appropriate systems know what the problem is and how to fix it. A near instantaneous change in a voltage or current detected by PMUs is almost certainly the result of a system failure for example. If any part of the system does something that isn't normal according to the network identification method being used, it will inform the appropriate parts of the system to remedy the situation. These protection methods will hopefully promote the prevention of failures and blackouts as well as aid in the quickness of recovery if a failure were to happen.

III. Methods of Online Branch Event Identificiation

There are several different types of NIMs which have their advantages and disadvantages. Generally neural network algorithms are implemented and trained with PMU data to be able to identify specific online branch events. Methods that will be covered in this report are admittance matrix identification (AMI), multilayer perceptron networks (MLP), convolutional neural networks (CNN), and learning vector quantization (LVQ).

A. Admittance Matrix Identification (AMI)

AMI is a primitive NIM which uses the input and output voltages and currents as variables to calculate the admittance matrix. This is calculated continuously, and the change is compared to a threshold. If the change exceeds that threshold, then the appropriate part of the SG is informed to address the issue. AMI is rarely used today because the calculations required are extremely high and it doesn't prioritize the areas where abnormalities are much more likely to occur. This makes the system slow which is why other NIMs are used in favor.

B. Multilayer Perceptron (MLP) Networks

The MLP network uses a hidden layer of neural nodes to collect information in addition to the input and output node layers. These neural nodes have weights to them which prioritize certain variables based on which is most likely to be abnormal [7]. This is determined by random cases which teach the neural nodes which variables to prioritize in addition to what has happened in the history of the network. While MLP is great for small scale networks, the fact that all neurons need to be connected will slow the system down significantly in a large-scale network.

C. Learning Vector Quantization (LVQ)

LVQ is a type of artificial neural network that implements a hybrid of supervised and unsupervised classification algorithm. LVQ works by forming classifications and in the network each neuron that appears in the first layer is assigned to a specific class [5]. It common that multiple neurons in the first layer will be assigned to the same class. The neurons in the first layer are then all assigned to a neuron that is in the second layer. The number of neurons represented in the first layer is represented by S1 and is greater than or equal to the number of neurons in the second layer S2 [5]. Figure 1 below shows a visual representation of the neural network and neuron connections.

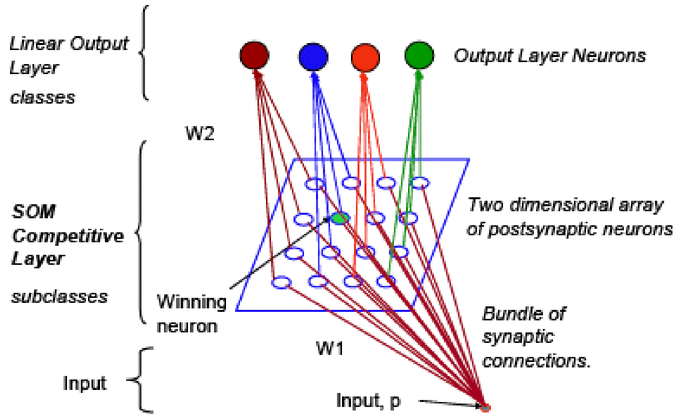

Fig. 1 Learning vector neural network [5].

Like MLP at the beginning of training random weight vectors are created. With each iteration of training the weight vectors are adjusted so that the input vector can be classified to the correct output [4]. The neuron whose corresponding weight vector is closest to the output a one and all the other neurons will output a zero [5]. This shows the winner take all methodology of LVQ. The winning neuron represents a subclass, and there may be multiple subclasses that compose each class. The second layer of the network is used to combine the subclasses using the output weights W2 [5].

D. K-Nearest Neighbors Algorithm (KNN)

KNN or K-Nearest Neighbors is a sorting algorithm that takes a point, finds the k data points with the most similar inputs and assigns that point the output that the most neighbors have. KNN can also be weighted, meaning instead of simply counting the number of neighbors with each output, the output with the highest weight sum is chosen. As new data points are inputted, they are then classified by finding the points K closest points. This is a simple algorithm that allows for very efficient set up times and performance with classification problems.

1) Example KNN Calculation

Let X1={(5,2),R}, X2 = {(4,6),B}, X3 = {(1,5),B} be the nearest neighbors, let I=(5,3) be the input, let N_R be the sum of the R weights and let N_B be the sum of the B weights. This case will have the inverse of the Euclidean distance as the weight. The equations for both N_R and N_B can be observed below.

$$N_R = \frac{1}{\sqrt{(5-5)^2+(3-2)^2}} = 1$$

$$N_B = \frac{1}{\sqrt{(5-4)^2+(3-6)^2}} + \frac{1}{\sqrt{(5-1)^2+(3-5)^2}} = 0.53983$$

Thus, the output for I will be R.

E. Convolutional Neural Networks (CNN)

CNNs are one of the most widely used options for SGs because they are like MLP networks, but they perform much better in large scale networks. The key difference is the hidden layer uses smaller and shared weights so there isn't a need to share data with every node in the system. Needing less communication will make the system much more efficient and much easier to teach.

F. Cellular Computaional Network (CCN)

Like convolutional neural networks, cellular computational networks are another method for reducing the computational expense of training neural networks. When implementing AI based transmission branch event detection and methods it's important to look at the efficiency of the model to produce a scalable method. When looking at applications such as the smart electric power grid there will be many inputs and outputs increasing the training time and expense. CCNs are a type of sparsely connected dynamic recurrent networks (DRNs) [1]. This allows for the training of the network to decentralized linking pieces together to be independently computed. CCN will allow for large scale problems and I/O sets to be broken down scaling the speed and efficiency of training greatly [3].

IV. Implementation and Analysis of Online Branch Identification

In this report a survey of a variety of different methods of online branch event identification was done as well as the analysis of two different methods. The methods that were further tested were MLP, LVQ, and KNN. Results were produced to determine which method would be more accurate and ultimately with the goal of improving one of the methods.

Test were performed to determine which AI based transmission branch event detection and identification method would perform better in energy control centers. Test data was collected for both training and testing the method on a modified IEEE 2 Area 4 Machine system. Within this system a sequence of seven different transmission lines were toggled to go down representing the branch events. Two online simulations were run, and inputs and outputs were collected. The first simulation was used to train the two different AI approaches. The second simulation was used for testing inputs to the two different AI approaches and the outputs could be used to verify results.

A. Multilayer Perceptron (MLP) Performance

First, we will analyze the training time performance of the MLP network. The MLP network that was trained and tested in this report was not at all distributed resulting in slower overall training times. The training times greatly varied from test to test. This variance can be attributed to the fact that the initial weight values are randomly assigned. Thus, the weights could require a varying range of adjustments and test iterations. The time it took to train the network took anywhere from 5 minutes to 45 minutes. This is an area that would likely need to be addressed and improved if this network were to be introduced to a large-scale system such as our power grid.

The second area of the MLP performance that was analyzed was the identification accuracy. As previously noted, the test case that data was collected for represented the sequential outage of seven different transmission lines. At any given time in the test there could only be one downed transmission line. The goal of the MLP network was to accurately report the status of all the lines. From the tests it was clear that the MLP network could follow the general trend of the transmission lines. The network would show when the given power line when out and when it was active. Although the general trend was accurate the fine details of the prediction was not. As shown by figure 2 below there was a lot of unpredictable spikes and inaccuracies in the prediction.

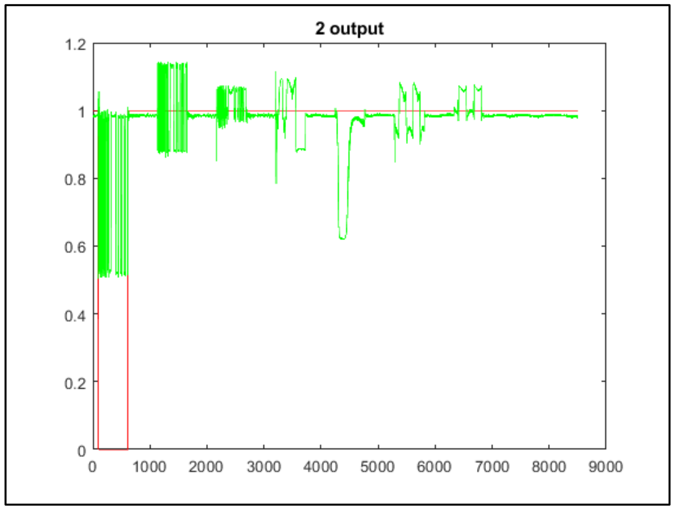

Fig. 2 Example transmission line testing results from MLP network.

Figure 2 above represents one of the transmission lines that went down during the simulation. The red line shows the real output data where the transmission line goes down at about 0 and is restored at about 500. The green line represents the MLP networks branch event prediction. It shows that the green line follows the general trend but, would be hard to follow and detect branch events with the constant spikes.

Ultimately to use an MLP network approach it would need to be combined with a form of distribution and filtering. If the network was combined with something like a CNN or CCN it would greatly improve the training efficiency. The network would then need to be paired with a filter to smooth out the spikes in the data and emphasize the correctness of the general trend.

B. Learning Vector Quantization (LVQ) Perfromance

The second approach that was applied to our testing scenario was a learning vector quantization network. Like the MLP network the LVQ network was not distributed resulting in unoptimized training times. Following a similar pattern to the MLP network the training times generally varied from test to test. This randomness can again be attributed to the fact that the initial weights of the network were randomly generated and assigned. Ultimately though on average the training time of the LVQ network was quicker than the MLP network. The time it took to train the network took anywhere from 2 minutes to 20 minutes. Again, this that would need to be addressed to properly scale the network and use this as a real method for online branch identification.

The identification accuracy was the second area of the LVQ network that was analyzed to determine the performance of the network. The LVQ network was tested with the same data that was used to test and train the MLP network. This allowed for an accurate and fair comparison between the two networks. The LVQ was developed and trained, but ultimately performed poorly in the identification of branch events. Like MLP the LVQ network did not have any issues simulating an uninterrupted branch but displayed sporadic spikes when a transmission line would go out. Unlike the MLP network though and represented by figure 3 it was hard to follow a general trend from the LVQ network.

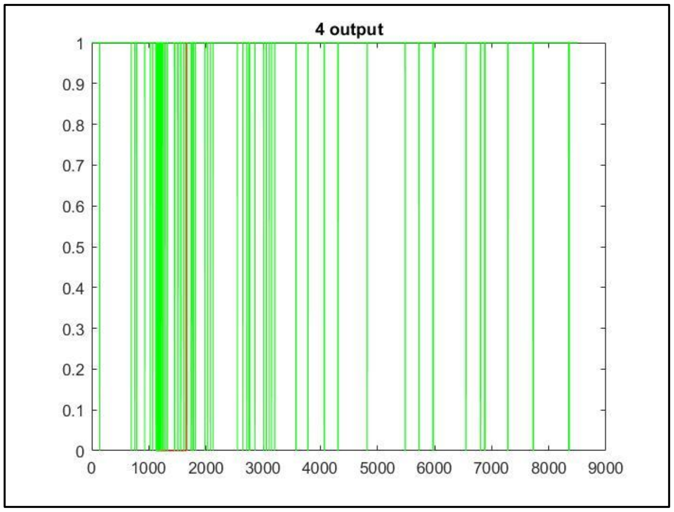

Fig. 3 Example transmission line testing results from LVQ network.

Figure 3 above represents one of the transmission lines that went down during the simulation. The red line shows the real output data where the transmission line goes down at about 1000 and is restored at about 1500. The green line represents the LVQ networks branch event prediction. The inaccuracies in the performance of the LVQ network's identification are clear in figure 3. There is a random on off oscillation when there should only be a small window of 0 prediction. Although this figure shows inaccuracies it can be noted that there is a greater density of 0 predictions and oscillations over the period that represented when the transmission line was down.

The LVQ network has potential to be a very strong method for accurate online branch event identification. The LVQ network represents a classification algorithm and the problem being addressed is a classification problem. Although this is true improvements to LVQ algorithm need to be made first. Similar to the MLP network combining the LVQ approach with some form of distribution such as a CCN or CNN would greatly improve the training time and make the network scalable. The LVQ will also require revisions to its implementation to improve that accuracy to a point where general trends could be determined. After reaching this point like the MLP network filters could be applied to fine tune the accuracy.

[1] *K-Nearest Neighbor (KNN) Perfromance*

The third approach that was applied to our testing scenario was the K-nearest neighbor approach. Like both MLP and LVQ approaches this method was not distributed, this was less critical because of how the algorithm works. This approach does not require training at all, but simply to create a model. Creating the model for the data only took several seconds compared to taking an upwards of several hours for the MLP network depending on the number of epochs used.

Like the previous methods the identification accuracy of branch events was the second area that the algorithms performance was tested. The same data was used to set up and test all three algorithms which allowed us to produce controlled and comparable results. When the K value of the KNN was adjusted to seven, representing the number of neighbors used to classify new inputs the most accurate results were produced. The classification results ended up being about 99.95% accurate. Like the other algorithms there were no issues simulating an uninterrupted branch and showed promising identification of interrupted branches. As shown by figure for the identification of a branch event on output 2 was about 100% accurate.

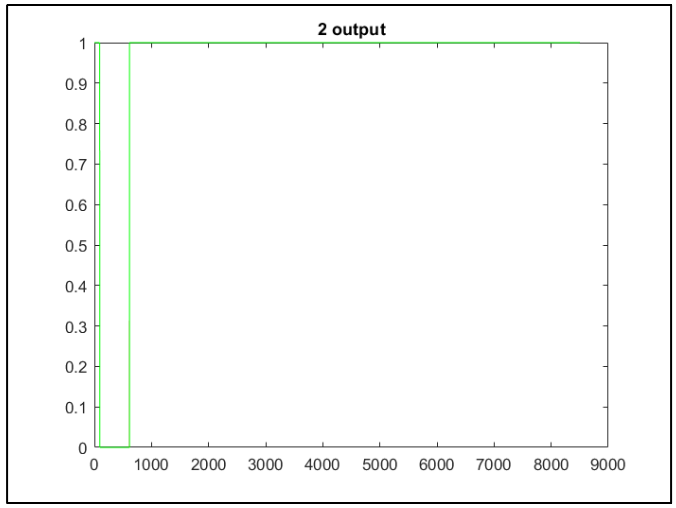

Fig. 4 Example transmission line testing results from KNN network.

Figure 4 above represents one of the transmission lines that went down during the simulation. The red line shows the real output data where the transmission line goes down at about 0 and is restored at about 500. The red line in this figure is covered by the mirroring green line. The green line represents the KNN network's branch event prediction. It shows that the green line follows the red line exactly and predicts the branch event with a very high precision. There were instances where the KNN did not produce results with the exact precision modeled by figure 4.

The KNN showed great performance in both speed and detection accuracy. However, the detection accuracy is still behind MLP at high epochs, but the cutoff point for the number of epochs where MLP pulls ahead will take around an hour or more to train compared to the almost instantaneous model creation with KNN. To continue to improve some of the detection cases where the accuracy was not exact the KNN would require larger training sets. Unlike a major drawback with both the MLP and LVQ networks increasing the training set size would not result in a large increase of the training time. Thus, the KNN network algorithm showed the best results in speed and performance. The KNN network also proved that it could be improved with the simplest modification when compared with the other networks.

This KNN network used a single model with all the PMU readings unlike the planned version which will use a model for each breaker and only use the PMU ratings directly related to the breaker location. The reason for this is to avoid having data that has little to do with if a breaker is out contribute to the distance and therefore classification of if a breaker is out or not. This is suspected to be the main cause for inaccuracies in the performance.

C. Comparative Results Analysis

After taking a detailed look at each model the results were directly compared to gain a comparative look at each models' performance. In order do this the models were trained with the same data and given the same input data. The estimated output data was then compared to the actual outputs for a reference. The decision was made to remove the LVQ model data from the comparative graphs because of its distracting and completely inaccurate predictions. The 7 graphs below represent each one of the 7 branch outages that were focused on in this study. In each one of the graphs the title represents the transmission line, the red and green lines represent the KNN and MLP model respectively and the dashed black line represents the actual output.

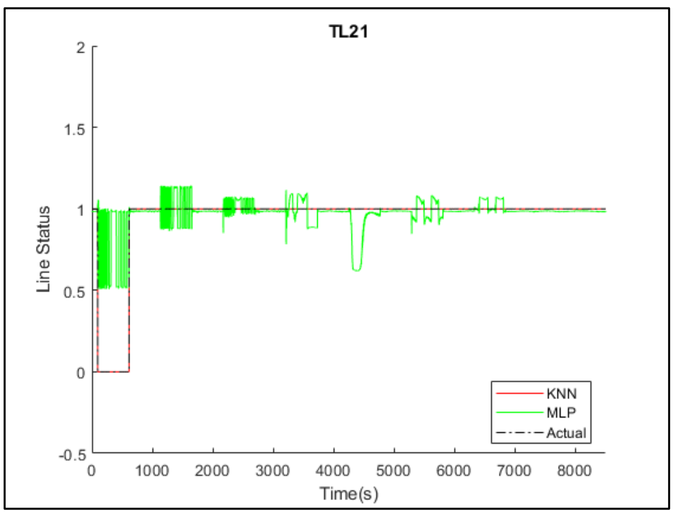

Fig. 5 KNN, MLP and Actual outage detection on transmission line 21.

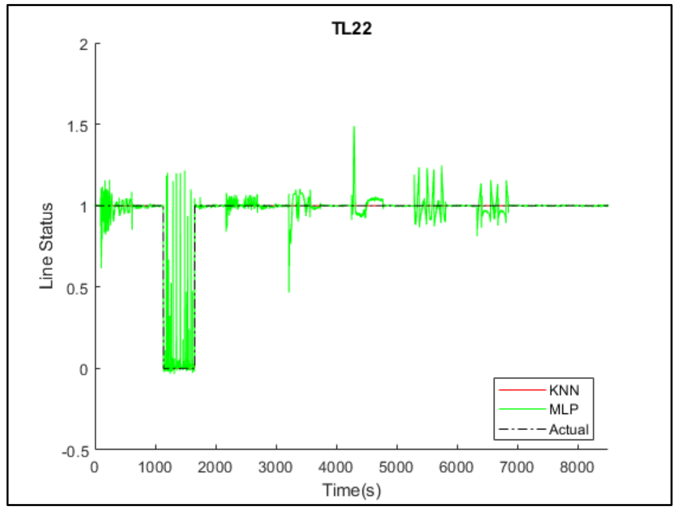

Fig. 6 KNN, MLP and Actual outage detection on transmission line 22.

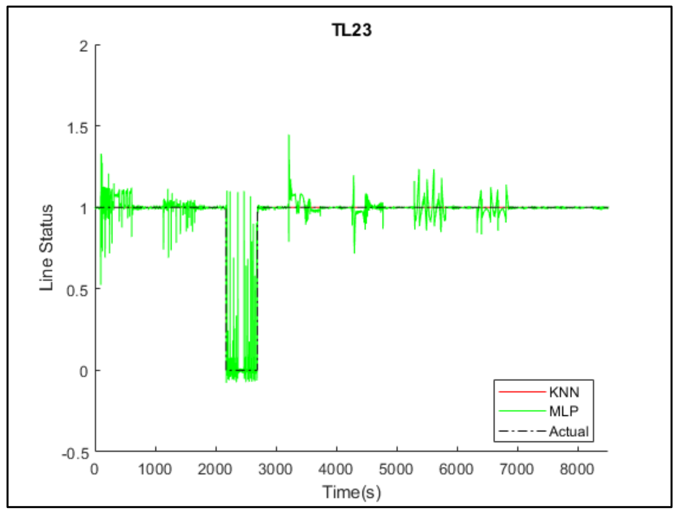

Fig.7 KNN, MLP and Actual outage detection on transmission line 23.

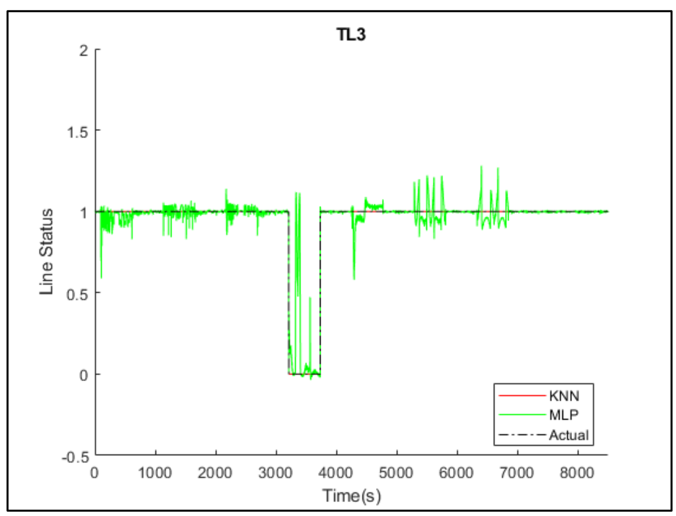

Fig. 8 KNN, MLP and Actual outage detection on transmission line 3.

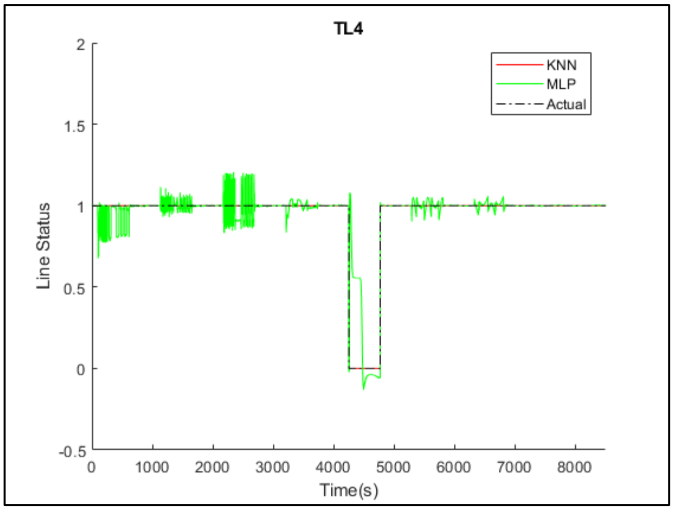

Fig. 9 KNN, MLP and Actual outage detection on transmission line 4

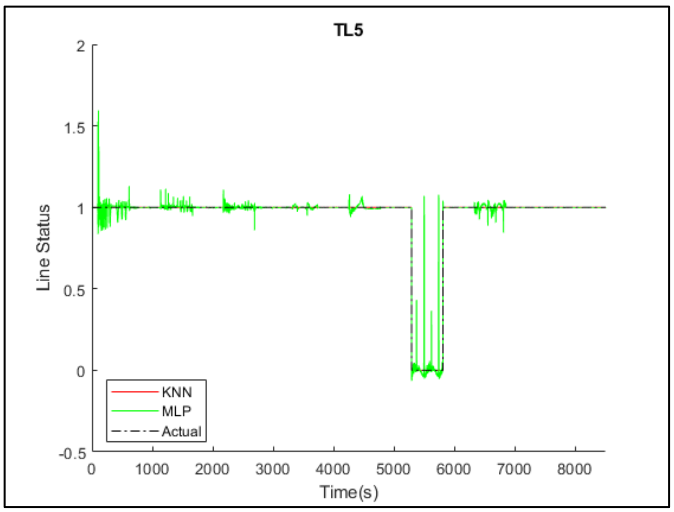

Fig. 10 KNN, MLP and Actual outage detection on transmission line 5.

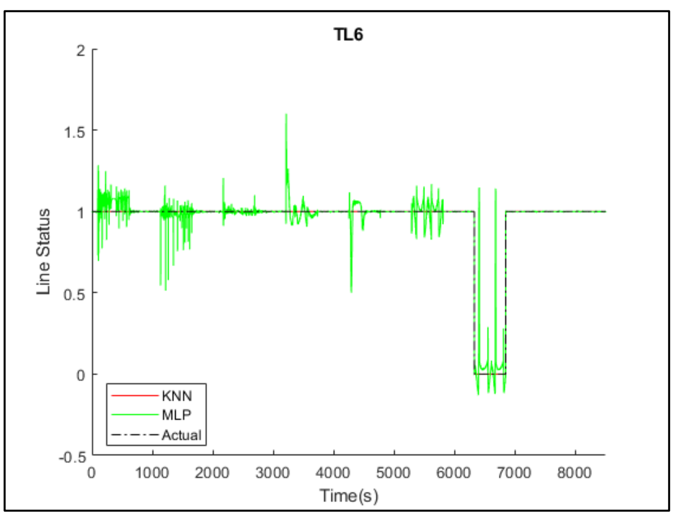

Fig. 11 KNN, MLP and Actual outage detection on transmission line 6.

The series of graphs painted a clear image as to which model was producing more accurate predictions. In each one of the seven different line outages the KNN model mirrors the actual output. The MLP model generally followed the trend of the actual output but had a lot of noise and often failed to completely predict the outage correctly. The MLP model ultimately showed inconsistent results that would have not functioned well if implemented in the online model. The KNN model produced extremely accurate results as shown in the results graphs. For each one of the seven outages the KNN model reliably predicts the correct time where the line status is down. When compared to the actual data the KNN method was 99.95% accurate and the MLP mode was only 89.78% accurate.

V. KNN Online model Implementation

A. Overview

This KNN implementation uses a model for each transmission line of interest using the normalized and distance weighted values of the currents on the transmission line and the adjacent bus voltages as inputs. The predicted output is determined by finding the inverse distance sums of the inputs corresponding to each output and choosing the one with the highest inverse distance sum. The data normalization first finds the norm vector which is a r_{input} x 1 vector that is the mean of the input absolute values with outputs having all transmission lines online. The normalized input is then calculated by dividing each row by the number on the norm vector's row corresponding to it. The models use input matrices with the columns containing the normalized line current measurements multiplied by the current weight at the top and the normalized adjacent bus voltages on the bottom. The estimated output is found by using the predict function on the test input rows of interest for each model. This method is extremely fast compared to MLP and is also accurate. As stated previously the accuracy in the test case was 99.95%.

B. The Norm Vector

Finding the norm vector is a two-step process. The first step is finding the sum vector of the raw input columns absolute values corresponding to all transmission lines online output's, as well as keeping track of how many of those inputs were added to the sum. The second step is to find the mean by dividing the sum vector by the number of inputs with all transmission line online outputs found. The reason only inputs corresponding to all transmission line online outputs are used to calculate the norm vector is for consistency as the norms will be different if a different percentage of inputs with all transmission line online outputs are used which will skew the accuracy.

C. Models

The models will use only inputs directly related to the breaker and k will be relatively low but not too low to avoid noise corrupting the data and to stop false transmission line online positives that would result with k being too high. Each model input will only include the normalized currents the transmission line is on times 2 as well as the normalized adjacent bus voltages and the outputs will be the row with the transmission line's status. A function is implemented to create the input matrix used for the models with the parameters being the normalized input matrix and the 2Nmatrix with the indices of interest on the first row and their weights on the second where N is the number of input rows of interest. This does not require training at all, so it is much faster than other networks like MLP.

D. Predicting Output

The estimated output is found for each transmission line of interest individually by using the predict function using the normalized, distance weighted and appropriate inputs. This means appropriate input matrices need to be found with the test input just like the training input, though the same norm vector is used for the test inputs. Any transmission line that is not of interest is assumed to be always online meaning their rows are hardcoded to 1s. It is an option to create models for all 16 transmission lines, but it is pointless and only wastes data and processing power since we were only focusing on 7 different lines.

VI. VISUALIZATION

Visualization is a key element in realizing the benefits of network branch event identification. After a method is developed and tested for identifying network branch events there needs to be a component allowing operators to see this identification. Data representation is defined as a graphical representation of information and data. This visualization will provide operators with a way to see trends, outliers, and important data. It is important that the visualization is not overly complicated and highlights the most important areas. Ultimately visualization will be a tool of situational awareness that can be used by operators in energy control rooms.

A. Visualization Method

The KNN network that was discussed in this report was developed using MATLAB software. This design choice resulted in the visualization GUI to be developed using MATLAB software as well. MATLAB provides a wide range of tools for visualization using its App Designer application. Although MATLAB provides great tools for visualization, its major downfall is the speed of its communication. There tends to be a delay in the communication between the RTDS and the MATLAB software.

B. Visualization Representation

The visualization was designed with an emphasis on clearly showing when an outage occurred and at which transmission line it occurred. The goal of the user interface was to provide a simplistic yet effective visualization of branch outage events that occurred during online testing. Ultimately the visualization was designed with the wants and needs of real grid operators. The visualization will provide the grid operators with a situational awareness of the grid by showing them statuses of all the transmission lines. A visual representation of the interface can be seen below in figure 5.

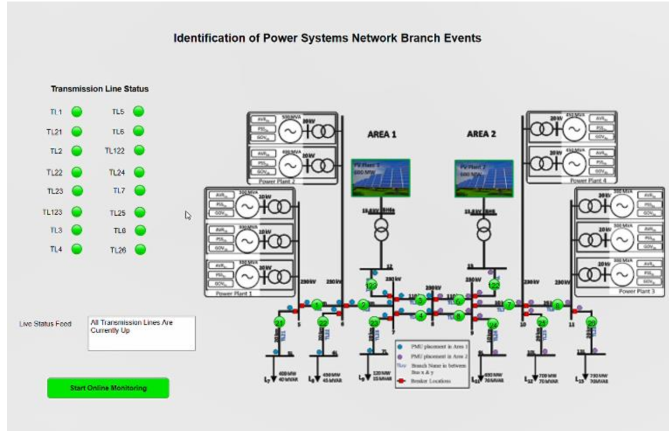

Fig. 12 Identification of power system network branch events visualization model.

There are three main components that make up the visualization model. The first component is the 16 lights that make up the transmission line status panel. These lights will be used to toggle from green to red depending on the status of the corresponding line. If the line is down, then the corresponding breaker status light will be red. If the line is active the light will remain green, as shown in figure 5. The second component is represented by the green lights that are placed on the different lines of the grid model. All these lights correspond to a single light on the transmission line status panel. This will allow grid operators to also have a live view of where in the system a line is down. The last component that is included in the visualization is a “Live Status Feed”. This status feed will be used in sync with all the lights to provide the grid operator with written based updates on the status of the grid. This live feed will generally display two different messages to the grid operators. These messages include “All Transmission Lines Are Currently Up” and “Transmission Line _ Down. Contingency plan in effect.”. The line in the status feed regarding a contingency plan was used to indicate that at this point an alert could be sent by the system to indicate that a transmission line is down. In combination these components will combine to create a simplistic but effective data visualization for any grid operator.

VII. CONCLUSION

As the development and transition to a smart grid continues to grow so will the need for new supporting technology. One of the key features and benefits of the smart grid is the increased amount of available data, information, and monitoring. This data is largely coming from measurements that are produced by PMUs. Although this data is currently being collected it is not being properly used by power grid system operators. System operators don’t currently have the needed technology to process and information and turn it into situational intelligence. This is the area where artificial intelligence-based transmission branch even detection and identification methods come into play. These methods will allow for grid operators to optimize the collected and improve the security of the grid. There are currently many different network identification methods that need to be tested and further developed. Some of these methods include MLP networks, KNN networks, and LVQ networks. All three methods were tested and showed potential in their abilities to produce accurate identification of branch events. The networks also presented many challenges and areas for improvement. Most notably the networks needed to be filtered to improve accuracy and distributed to improve scalability and efficiency. Lastly comparing the three networks that were more closely analyzed the KNN network showed the best results. Its simplistic approach provided very fast set up times and accurate identification results. The only drawback with KNN is that it is somewhat difficult to optimize compared to the other methods. Similar to the research showing the scalability of the LVQ network the KNN has the potential to be quickly scaled and applied to use cases within the smart grid. The modular approach that was utilized in the KNN implementation proved its ability to be easily scalable and very efficient.

The KNN model was further adapted to become compatible with the RTDS and designed visualization software. The KNN model was adapted to have individual models for all 16 of the outputs in the power system model. These outputs are then being represented through a visualization model. The visualization model uses a simplistic design but provides the operator with situational awareness and clear view of the system's status.

REFERENCES

[1] B. Luitel and G. K. Venayagamoorthy, "Cellular computational networks – a scalable architecture for learning the dynamics of large networked systems," Neural Networks, vol. 50, pp. 120 – 123, 2014. [Online].

[2] D. Madurasinghe, P. Arunagirinathan and G. K. Venayagamoorthy, "Online Identification of Power System Network Branch Events," 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, 2019, pp. 1-5, doi: 10.1109/ISGTEurope.2019.8905508.

[3] D. Madurasinghe, P. Arunagirinathan and G. K. Venayagamoorthy, "Distributed Identification of Power System Network Branch Events," 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC,2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281775.

[4] D. Madurasinghe and G. K. Venayagamoorthy, "LVQ Neural Network for Online Identification of Power System Network Branch Events," 2020 Clemson University Power Systems Conference (PSC), Clemson, SC, USA, 2020, pp. 1-7, doi: 10.1109/PSC50246.2020.9131340.

[5] G. K. Venayagamoorthy and M. Zarghami, "Wide area power system protection using a learning vector quantization network," Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, Arlington, VA, USA, 2005, pp. 5 pp.-, doi: 10.1109/ISAP.2005.1599286.

[6] J. De La Ree, V. Centeno, J. Thorp, and A. Phadke, "Synchronized phasor measurement applications in power systems," in IEEE Transactions onSmart Grid, vol. 1, no. 1, pp. 20-27, June 2010.

[7] Uniqtech, "Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning," Medium, 13-Jun-2019. [Online].