Real-Time Detection of Critical Generators in Power Systems: A Deep Learning HCP Approach

Bhavesh Shinde*, *Student Member, IEEE,* Shiyuan Wang*, *Student Member, IEEE,*
Payman Dehghanian*, *Member, IEEE* and Mohammad Babakmehr†, *Member, IEEE*
*Department of Electrical and Computer Engineering
The George Washington University
800 22nd St NW, Washington, Suite 5900, DC 20052, USA.
†Ford Motor Company, Global Data, Insights, and Analytics
Dearborn, Michigan, USA.
{*shindebhavesh, shiyuan1225, payman}@gwu.edu , †mbabakme@ford.com

***Abstract*—Assessment of power system transient stability is critical for a reliable continuous operation and to ensure none of the working generating units in the system go out of synchronism. Therefore, a fast and accurate surveillance of transient stability in power systems is necessary. This paper proposes a deep learning neural network framework that captures the phasor measurement unit (PMU) measurements and monitor the system transient stability in real-time. The proposed framework utilizes the convolutional neural network (CNN) with hypotheses CNN pooling (HCP) to identify the state of the system and detect the set of critical generators. The suggested CNN module for stability estimation and the robust HCP module for detecting critical generators through multi-label classification are tested on the IEEE 118-bus test system, where different types of faults at different locations and under varying system load conditions are simulated. The test results verified that our proposed framework is fast and accurate, thereby a viable approach for online system monitoring applications.**

***Index Terms*—Transient stability analysis; Convolutional neural network (CNN); Hypotheses CNN pooling (HCP); Phasor measurement unit (PMU); Deep learning**

I. Introduction

Transient stability assessment, as a mechanism to capture the power system dynamic security conditions, plays a significant role in day-to-day power system operation. This is particularly critical due to the following: (i) continuing growth in the system interconnection size and complexity; (ii) proliferation of renewables and rushing arrival of renewables; (iii) the increasing demand for electricity. As the power system operating point is reaching the stability limit and its control becomes more challenging and difficult, the instability problem is more likely to occur [1], which would potentially lead to system outage and blackouts [2], [3]. Real-time and accurate monitoring of the system transient stability is thus critical and can help determine the state of the system following large disturbances (e.g., faults, loss of load, etc.) [4], [5].

Previously, a variety of research methods have been proposed to assess the system transient stability. The traditional methods through time-domain simulations (TDS) [6], which require high computations for solving non-linear differential algebraic equations, are the mainstream Transient Stability Analysis (TSA) approach in power systems. Transient energy function (TEF) [7] and partial energy function (PEF) [8] which compare the potential and kinetic energy values of the system against the reference value have presented drawbacks in accurately estimating the actual energy values in practical scenarios. Other methods such as the extended equal area criterion (EEAC) [9] and trajectory convexity-concavity method assess the transient stability based on the characteristics of the equivalent single-machine infinite bus system (SMIB) [10], which although computationally more efficient, the performance accuracy is being compromised.

Recently, techniques based on data driven models such as support vector machine (SVM) [11]–[13], decision tree, local regression and neural networks (NN) [14]–[16] have been explored to assess the system stability performance. The latest methods proposed are based on the time-series models for time-adaptive transient stability assessment in power system. SVM is seen to produce better results while assessing transient stability; however, in certain situations, inaccurate information of post-fault conditions lead to significant degradation of the model performance. Long short-term memory (LSTM) [17] methods are effective while dealing with feature extraction in time dimension, but due to their disadvantages in parallelization, training stability and inference speed, feed forward models in time-series classification are preferred.

In this paper, a real-time framework is proposed for online monitoring of the power system transient stability. The proposed monitoring system utilizes available measurements—voltage and current magnitudes, voltage and rotor angles, and system frequency—obtained from phasor measurement units (PMU) distributed across the grid. The deep learning network used in the proposed system is the convolutional neural network (CNN) [18], which takes the heat-map representation of the aforementioned variables measured from PMUs and rearranged over a 3D matrix as the input [19]. The proposed network is a Y-net architecture [19] which detects the operating status of the system and the critical generators in case of unstable scenarios. This involves two different classifier networks, i.e., multi-class classification for predicting the state of the system and multi-label classification for identifying the critical generators. To enhance the performance of the second classifier , a flexible hypotheses CNN pooling (HCP)

978-1-7281-4436-8/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.

[20] approach is proposed to produce ultimate multi-label predictions to aggravate accurate identification of the system critical generators over all other previously proposed models.

The rest of the paper is structured as follows: Section II introduces a background on the CNN and HCP models. Section III introduces the proposed framework consisting of the data collection and processing, and the deep learning neural network model architecture. Section IV presents the numerical results and a comparison of results with other models. Finally, conclusions are provided in Section V.

II. Background on CNN and HCP Models

Within the family of neural networks, and to train the data with grid-like topology such as images etc., deep CNN has been one of the greatest breakthroughs [21]–[23]. CNN consists of convolutional layer, pooling layer and fully connected layers (FCs). When applied to single-label (multi-class) image classification, CNN can handle well-aligned images very well [24]; however, for multi-label image classification, there arises complexity of miss-alignments and occlusion which would lead to relative inaccuracy in the prediction of such multi-label classification. On the other hand, HCP additionally consists of max pooling layer compared to CNN. It is a flexible deep CNN structure which can help alleviate these issues, since HCP takes segment hypotheses as the input which are generated by object detection techniques, and then connects a shared CNN to each hypotheses, finally aggregating single-label predictions from different hypotheses into the multi-label results [20].

Typically, the features are extracted from the input data via the convolutional layer by the convolution kernel, defined by,

$$O_C^j = f\left(\sum_{i\in M_j} x_i * k_{ij} + b_j\right) \tag{1}$$

where O_c^j is the j^{th} feature map of the convolution layer, x_i is the input image set, k_{ij} is the convolution kernel corresponding to O_c and x_i, b_j is the bias. $*$ stands for the convolution operation and f is defined as $f = max\{0, x\}$.

The pooling layer extracts the important features from the convolutional layer, the output of which can be formulated as

$$O_P^j = f\left(\beta_j \mathrm{down}(O_C^j) + b_j\right) \tag{2}$$

where *down* sampling function is represented and β_j is the j^{th} multiplier of the pooling layer. The fully connected layer may consist of multiple hidden layers and its output can be generally represented in the following form

$$\hat{y} = \sigma(\omega O_P + b_j) \tag{3}$$

where ω is the weight and σ is the activation function. Suppose v_i is the output vector of the i^{th} hypotheses from the shared CNN and v_{ij} is the j^{th} component of v_i. The max pooling layer can be then formulated as

$$v^{(j)} = \max\left(v_1^{(j)}, v_2^{(j)}, \cdots, v_m^{(j)}\right) \tag{4}$$

where v_j can be considered as the predicted value for the j^{th} category of a given image.

For a multi-label classification, $y_i = [y_{i1}, y_{i2}, \cdots, y_{ic}]$ is the label vector of the i^{th} image. If the image is annotated with class j, then $y_{ij} = 1$ or else $y_{ij} = 0$. The probability vector for i^{th} image is defined as $\hat{p}_i = y_i/||y_i||$. The cost function to be minimized is then defined as

$$J = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{c}(p_{ik} - \hat{p}_{ik})^2 \tag{5}$$

where M stands for the number of images

The final output of the first classifier is obtained through a softmax operator. Softmax function calculates the estimated probability scores for each individual class. These scores are useful in deciding the most probable class for each input pattern. The activation function used at the output layer for the second classifier is the sigmoid function. In sigmoid activation function at the output layer, the neural network models probability of a class as a Bernoulli distribution. Sigmoid function [25], unlike softmax, do not give a probability distribution around different classes as the output, but provides independent probabilities. Multi-label classification tasks are not mutually exclusive and each class is independent; therefore, this function allows for such types of classifications.

III. Proposed Framework

The proposed framework for online power system stability surveillance is illustrated in Fig. 1. The data obtained from PMUs is first used for offline training of the pre-built hybrid deep learning model. The trained model is then used for online detection of the power system stability and to identify sets of critical generators in real-time.

A. Training Data Acquisition

The parameters used for training the model are the current and voltage magnitudes, rotor angle, voltage angle, and frequency. The training data is collected from PMUs located at all generator buses across the network, the data on which is obtained through TSA simulations on the IEEE 118-bus test system in the PowerWorld software environment. The IEEE 118-bus test system comprises of 118 buses, 54 generating units, 99 load points, and 177 transmission lines. The TSA simulations are conducted for various types of faults (3-phase balanced faults on each bus and on each transmission line at three different locations of 25%, 50% and 75% of the line length) and under varying loading conditions in the system.

Each simulation is run for a period of 20 seconds and a time-step of 0.02 seconds is used throughout, such that 1000 timestamp recordings are available for each contingency. For each contingency, the fault is created exactly at t = 1s and the fault is simulated for 8 cycles (i.e., 0.1333 seconds), after which it is cleared. All such contingencies including bus faults and transmission line faults are repeated for several load levels in the system (base load and -3%, -2%, -1%, +1%, +2%, +3%, +5% and 7%), for which the load flows are converged.

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.

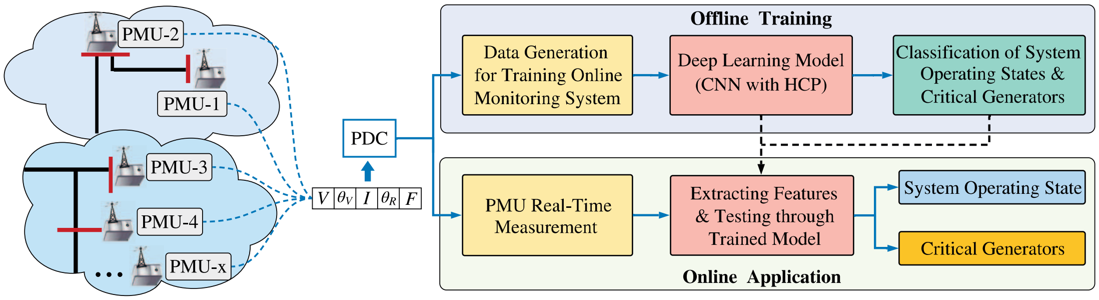

PMU-2
PMU-1
PMU-3
PMU-4
PMU-x
PDC
V θ_V I θ_R F
Offline Training
Data Generation for Training Online Monitoring System
Deep Learning Model (CNN with HCP)
Classification of System Operating States & Critical Generators
PMU Real-Time Measurement
Extracting Features & Testing through Trained Model
System Operating State
Critical Generators
Online Application

Fig. 1. The proposed framework for power system online stability surveillance.

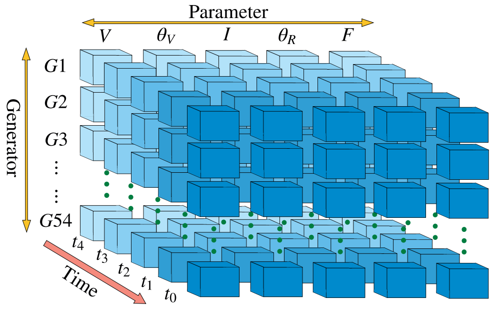
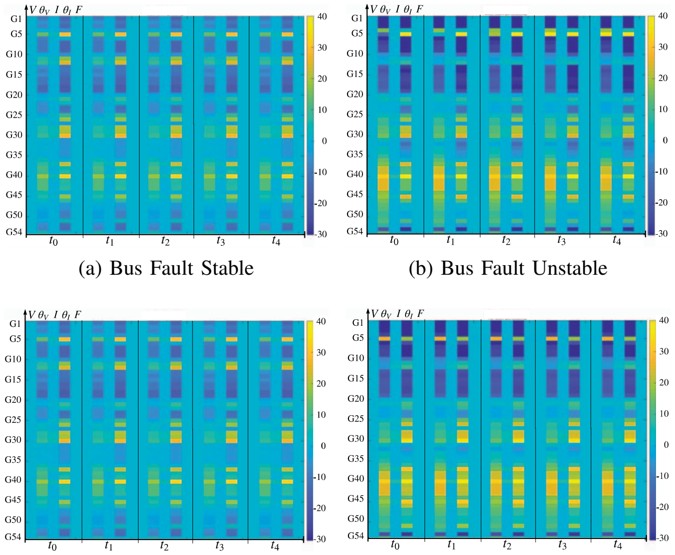

Parameter
V θ_V I θ_R F
Generator
$G1$
$G2$
$G3$
$G54$
t_4 t_3 t_2 t_1 t_0
Time

Fig. 2. 3D data matrix representation in the proposed framework.

B. Data Pre-Processing

In order to monitor the system transient stability in real-time, the surveillance system needs to continuously analyze the power system parameters over few time-steps. All parameters described in Section III-A are observed over a sliding window of time stamps, lets say t time-stamps. Therefore, at every sampling instant, the sliding window consists of $(t-1)$ past measurement recordings and one current measurement recording. The observed raw data is rearranged and transformed into a three-dimensional vector (timestamp, generator number, parameter) as shown in Figure 2, where color shades are assigned over time such that the darkest shade is the newest time-stamp entry into the data matrix. There are 54 generators ($G1$ to $G54$) on *Generator* axis and 5 parameters in all on *Parameter* axis. Also, a range of timestamps exist on the *Time* axis. The length of each observation window is 5 timestamps and the sliding step is 1 timestamp (note: these two settings are recommended by [19]).

A heat-map image of this three dimensional matrix is created for each sample, i.e., the data matrix for each sample is rendered a color image of size $T \times N \times P$, wherein T is the length of the observation window, N is the number of generators, and P is the number of parameters. Therefore, the size of each heat-map image for any particular fault scenario is considered constant and it is 5×54×5. The representation of stable and unstable cases for bus and transmission line fault is shown in Figure 3. The demonstrated heat-maps are obtained by rearranging the data from 3D into a 2D matrix form (54 × 25) through stacking all 5 timestamps on the *parameter* axis.



(a) Bus Fault Stable

(b) Bus Fault Unstable

(c) Transm. line fault Stable

(d) Transm. line fault Unstable

Fig. 3. Visualisation of the extracted features from data matrix in both stable and unstable cases.

C. Rotor Angle Estimation

Apart from the measurements directly obtained from the PMU units, generator rotor angle is a parameter that is used in the data matrix which is not available from the PMUs directly, but can be estimated by various dynamic state estimation methods [26]. During transient conditions, the rotor angle estimation is not as straightforward as in the case of steady-state conditions. Among various methods rotor angle estimation during transient conditions, the estimation using the damper current measurements is here pursued [27].

The electric torque of a generating unit is commonly expressed using the known parameters of the generator machine as given in (6) [27]:

$$T_e = (L_d - L_q) i_d i_q + k M_F i_F i_q + (k M_D i_q i_D - k M_Q i_d i_Q) \quad (6)$$

and the swing equation of the generating unit is given by the following equation:

$$\frac{2H}{\omega_s}\frac{d\omega}{dt} = T_m - T_e = T_a \quad (7)$$

where T_m, T_e, and T_a are the mechanical, electrical, and accelerating torque of the generator, respectively. $k = \sqrt{3/2}$, L_d and L_q are the stator inductances; i_d, i_q, i_D, i_Q, and i_F are the currents in the individual damper windings; M_D, M_F,

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.

and M_Q are maximum values of the stator to field, stator to q-axis damper winding, and stator to d-axis damper winding mutual inductances and ω and ω_s are the speed of the rotor and synchronous speed of the machine, respectively.

The accelerating or decelerating torque at the n^{th} instant on the rotor can be found from the following equation:

$$T_a(n) = T_m - T_e(n) \tag{8}$$

Inserting (8) in equation (7) and integrating it, the speed of the rotor can be assessed using the following equation,

$$\omega = \int \frac{T_a}{2H\omega_s} dt \tag{9}$$

The rotor angle can be eventually achieved by integrating the speed of the rotor as follows,

$$\delta = \int (\omega - 1) dt \tag{10}$$

The above equation helps in estimating the rotor angle of the system generators using PMU measurements during a transient operating condition (e.g., post disturbances).

D. Transient Stability Index

Transient stability assessment in power systems is captured using Transient Stability index (TSI). If a disturbance occurs and is cleared exactly after 8 cycles (t = 1.1333s), then the state of the system following the contingency can be theoretically determined via the TSI, which is defined as

$$\eta = \frac{360° - |\Delta\delta|_{max}}{360° + |\Delta\delta|_{max}} \tag{11}$$

where $\Delta\delta_{\text{max}}$ is the maximum rotor angle separation between any two generators following the fault. The system stability profiles obtained through the simulations are classified stable or unstable based on the value of η. A system is considered stable if $\eta > 0$, otherwise the system would be labelled as unstable. If a case is classified as an unstable case, and the angle separation of certain generators from the rest of the generators is more than 360°, then that set of generators is classified as critical for that particular contingency.

The system operating states are distinguished into six different classes as shown in Table I.

TABLE I
POSSIBLE SYSTEM STATE LABELS BASED ON OBSERVED DATA MATRICES

Class 1 : No Disturbance	Class 4 : Fault Clearance
Class 2 : Fault Occurrence	Class 5 : Return to Stable State
Class 3 : Fault Duration	Class 6 : Unstable State

The system operating state is differentiated based on various events of interest taking place in the system. A detailed description of these events and the classification of these six different classes are described as follows:

- ***Class 1:*** All the observed data matrices belong to the pre-fault operating time.
- ***Class 2:*** Any observed data matrix that covers the instant timestamp of the fault occurrence.
- ***Class 3:*** All the observed data matrices which cover exactly the timestamps that lie between fault occurrence and fault clearance (without instant timestamps of either fault occurrence or fault clearance).
- ***Class 4:*** Any observed data matrix which covers the instant timestamp of the fault clearance.
- ***Class 5:*** During post-fault clearance period, all the observed data matrices which reveal the stable state.
- ***Class 6:*** During post-fault clearance period, all the observed data matrices contain the instant timestamp of unstable states and all the timestamp afterward. Each data matrix is here associated with a set of critical generators.

The training data is generated follows Section III-A, and is classified and labeled accordingly based on Table I. It is then used to train the deep learning model presented next.

E. Proposed Hybrid CNN + HCP Architecture

The suggested Y-net CNN architecture is shown in Figure 4, where firstly smaller candidate windows are selected within the input image as hypotheses by a hypotheses selection process. The selected hypotheses are fed into two convolutional layers to compress the split image into feature maps. After extraction of features from the data matrix, the network is divided into two different branches. The upper branch in the network, shown as Classifier 1, works as a multi-class classifier which detects the system stability. The lower branch shown as Classifier 2 performs a multi-label classification which identifies the system critical generators. Within Classifier 1, the extracted feature is fed into Relu and fully connected layers, softmax is then used to give the output. For Classifier 2, the extracted feature acts as an input to the shared CNN and fuses individual hypotheses scores with a max pooling operation. Unlike Classifier 1, Classifier 2 uses a sigmoidal activation function. To integrate both branches, errors in both classifiers are back propagated together during the training process.

During the operation of the proposed framework, if Classifier 1 is detected as Class 6, then the output of the Classifier 2, i.e. the multi-label classification, is produced in parallel to the final output; if the output of Classifier 1 is detected as any class other than Class 6, the output of Classifier 2 is a null set. One should note that the proposed framework can only be used after training based on the simulated electrical system.

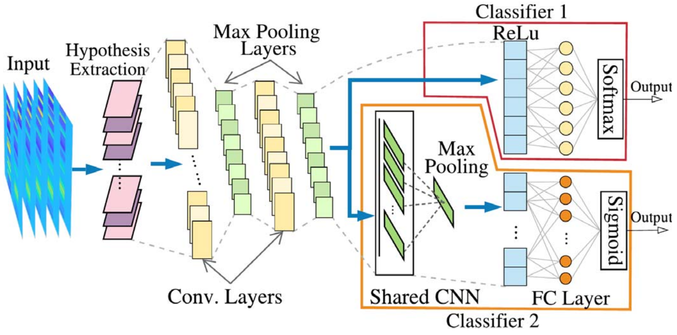


Fig. 4. The proposed CNN architecture with Hypotheses pooling.

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.

IV. Numerical Results and Analysis

The IEEE 118-bus test system is used as the test platform, where a total of 5652 contingency scenarios are simulated including different types of faults under varying loading levels. The data set is randomly split into the training and validation test sets, and the represented results are averaged over these trials. The number of sample windows in ***Class 5*** (stable post transient disturbance) is much larger than the other classes, whereas, in ***Class 2***, ***Class 3*** and ***Class 4*** the number of samples is relatively lower. Hence, to balance the data for training the NN, sub-sampling without replacement is used to represent the classes in equal proportion in the training dataset [19]. During sub-sampling, each sample is only sampled once and the training and validation dataset are mutually exclusive. Additionally, the input data for the training datasets in all the unstable cases of ***Class 6*** are modified with the associated matrix file containing the information about critical generators for each particular unstable case. The performance of the proposed work has been tested under no noise consideration.

The implementation of the CNN algorithm is achieved in Tensorflow 1.14.0 with NVIDIA GeForce RTX, 64GB GPU (CUDA 10.0) support. The NN is trained with Adam Optimizer with a batch size of 64. The data matrix of the PMU readings generated through simulations in PowerWorld Simulator are used as the input to train the NN. The size of the input data matrix is very small (5x54x5)—the equivalent heat-map image size is (54x25)—compared to the normal image size (300x300) or (512x512). Therefore, while training the CNN module, a comparable image-size and a similar kernel size is used. The consideration of smaller image size works better as there is a need to look for global features in the data matrix and not local features. A similar kernel size to the size of the input image reduces additional computational burden.

In our online monitoring system, along with the accuracy of the model, another critical factor is the time taken for computation of the outputs in both introduced classifiers. The time required for computing the final output of a given sample window should be extremely low for the system to be considered viable in real-time.

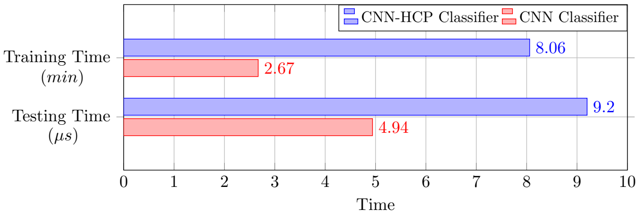


Fig. 5. Training and testing time for two different models.

The bar graph in Figure 5 represents the training time and the online monitoring time computed on two different models. The first model is the CNN model [21] (without HCP Framework) and the second model is the proposed flexible CNN-HCP framework used for detection of system critical generators. The time calculated for the final output of both models is 4.94μs and 9.2μs, respectively. Thus, although the proposed model is a bit slower in monitoring, it is still efficient enough to work in real-time. Furthermore, the training time for this model is relatively higher than that in traditional CNN model which is 2.67 minutes. The time required for training the proposed CNN-HCP network for the studied 118-bus test system is 8.06 minutes and the training is done offline.

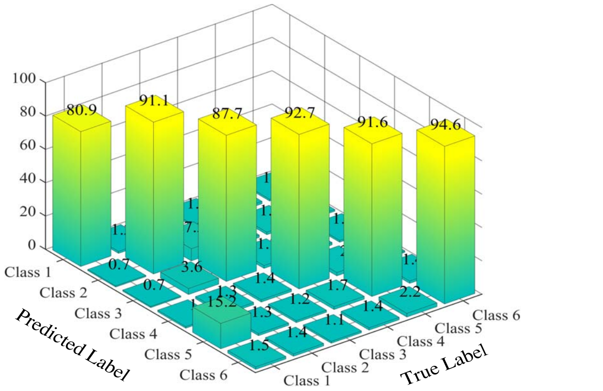


Fig. 6. Confusion matrix presenting accuracy of the proposed hybrid framework considering all loading conditions.

The confusion matrix in Figure 6 shows the overall accuracy in detecting the system operating state for different load conditions during the testing process. For each class, 3000 samples are taken for testing purposes. The labels in the *True Label* stand for the true class of the testing data and the *Predicted Label* stands for the classified results of the CNN-HCP model. As the result shows, 91.1% samples are correctly classified to detect ***Class 2*** and 1.4% of the samples are mistakenly classified as ***Class 6***.

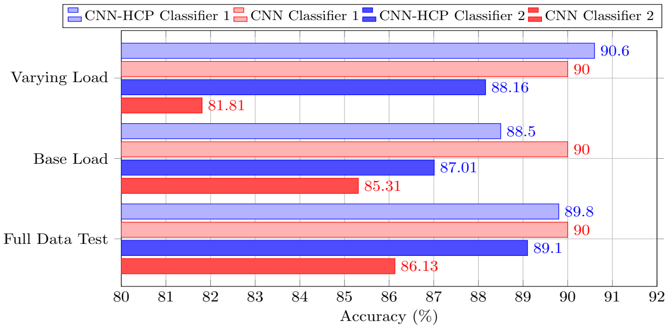


Fig. 7. Detection accuracy at base load and varying load conditions.

The greater training and testing time comes with an increased accuracy rate for the suggested hybrid model in this paper. The bar graph in Figure 7 represents a similar accuracy rate for classifying the state of the system and a considerable increase in accuracy rate while detecting the critical generators when the system is trained for the entire dataset (including all types of faults at all varying load conditions specified in Section III-A). Secondly, the proposed model is tested on the base and varying load conditions separately and compared with the CNN model in [21]. In all the given test cases, the results represent that the suggested CNN-HCP model outperforms in all conditions as compared to the previously studied CNN model in [21] which does not use the HCP framework for multi-label classification. For the base case load, when trained for 3-phase faults on buses and transmission lines at three different locations, the total number of 628 contingencies

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.

are considered. For the varying load conditions (±2 and 3%), a total number of 2512 contingencies are modeled for training, which includes all different types of faults. In both load conditions, the results of the Classifier 1 obtained from both frameworks reveal a similar accuracy rate. However, the accuracy rate of Classifier 2 from the proposed framework shows a significant improvement, compared with that from [21], which verifies that the proposed framework can be more robust when it comes to detecting the set of critical generators in power systems following a transient disturbance.

V. Conclusion

This paper presents an advanced deep learning framework for online detection of unstable operating states in power systems and real-time identification of system critical generators following disturbances. The proposed framework utilizes phasor measurements from PMUs at the generator buses and classifies the events based on the features extracted from the measurements. In the proposed framework, CNN is used to classify two different outputs simultaneously, which consists of multi-class and multi-label classifications, followed with a suggested HCP technique for the latter classification. The performance of the proposed framework is tested on a variety of scenarios and under varying load conditions. Simulations verified that the proposed framework with HCP reveals a more accurate outcome compared to the traditional CNN models. The suggested model comes with a higher accuracy at the trade off of the computing time, yet computationally-efficient and applicable to applications in online setting.

Future work could be targetted at implementing the proposed framework on a large real-world power grid, and validating the results accuracy and computational effectiveness during real-time applications. The performance of the proposed model could be tested and compared by using a real-time digital simulator. Additionally, an interesting and important direction could be to use reinforcement learning for power systems stability surveillance which can train the deep learning models online unlike the proposed work where the model needs to be trained offline.

References

[1] B. Tan, J. Yang, X. Pan, J. Li, P. Xie, and C. Zeng, "Representational learning approach for power system transient stability assessment based on convolutional neural network," *The Journal of Engineering*, vol. 2017, no. 13, pp. 1847–1850, 2017.

[2] A. Hoballah and I. Erlich, "Transient stability assessment using ann considering power system topology changes," in *International Conference on Intelligent System Applications to Power Systems*, Nov 2009, pp. 1–6.

[3] M. H. Rezaeian Koochi, P. Dehghanian, S. Esmaeili, P. Dehghanian, and S. Wang, "A synchrophasor-based decision tree approach for identification of most coherent generating units," in *Annual Conference of the IEEE Industrial Electronics Society*, Oct 2018, pp. 71–76.

[4] S. Wang, P. Dehghanian, M. Alhazmi, J. Su, and B. Shinde, "Resilience-assured protective control of dc/ac inverters under unbalanced and fault scenarios," in *2019 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Feb 2019, pp. 1–5.

[5] S. Wang, P. Dehghanian, M. Alhazmi, and M. Nazemi, "Advanced control solutions for enhanced resilience of modern power-electronic-interfaced distribution systems," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 4, pp. 716–730, Jul 2019. [Online]. Available: https://doi.org/10.1007/s40565-019-0559-9

[6] P. Kundur, "Power system stability and control. mcgrawhill, london, 1994. pikeresearch. worldwide revenue from microgridswill reach $17.3 billion by 2017," Technical report, PikeResearcch, 2012. URL http://www. pikeresearch. com . . . , Tech. Rep.

[7] M. Pai, *Energy function analysis for power system stability*. Springer Science & Business Media, 2012.

[8] S. E. Stanton, "Transient stability monitoring for electric power systems using a partial energy function," *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1389–1396, Nov 1989.

[9] Y. Xue, "Fast analysis of stability using eeac and simulation technologies," in *1998 International Conference on Power System Technology. Proceedings (Cat. No.98EX151)*, vol. 1, Aug 1998, pp. 12–16 vol.1.

[10] D. Ruiz-Vega and M. Pavella, "A comprehensive approach to transient stability control. i. near optimal preventive control," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1446–1453, Nov 2003.

[11] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Systems*, vol. 26, no. 3, pp. 1474–1483, Aug 2011.

[12] Zhehan Yi and A. H. Etemadi, "A novel detection algorithm for line-to-line faults in photovoltaic (pv) arrays based on support vector machine (svm)," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, July 2016, pp. 1–4.

[13] Z. Yi and A. H. Etemadi, "Line-to-line fault detection for photovoltaic arrays based on multiresolution signal decomposition and two-stage support vector machine," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8546–8556, Nov 2017.

[14] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for online transient stability assessment of power systems," *IEEE Trans. Power Systems*, vol. 4, no. 2, pp. 789–800, 1989.

[15] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn, "Decision tree based transient stability method a case study," *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 459–469, Feb 1994.

[16] C. Zheng, V. Malbasa, and M. Kezunovic, "Regression tree for stability margin prediction using synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1978–1987, May 2013.

[17] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li, "Intelligent time-adaptive transient stability assessment system," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1049–1058, Jan 2018.

[18] S. Wang, P. Dehghanian, L. Li, and B. Wang, "A machine learning approach to detection of geomagnetically induced currents in power grids," in *IEEE Industry Applications Society Annual Meeting (IAS)*, 2019, pp. 1–7.

[19] A. Gupta, G. Gurrala, and P. S. Sastry, "An online power system stability monitoring system using convolutional neural networks," *IEEE Trans. Power Systems*, vol. 34, no. 2, pp. 864–872, March 2019.

[20] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, Sep. 2016.

[21] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[22] S. Wang, L. Li, and P. Dehghanian, "Power grid online surveillance through PMU-embedded convolutional neural networks," in *IEEE Industry Applications Society Annual Meeting (IAS)*, 2019, pp. 1–8.

[23] L. Li, M. Doroslovački, and M. H. Loew, "Discriminant analysis deep neural networks," in *53rd Annual Conference on Information Sciences and Systems*, March 2019, pp. 1–6.

[24] ——, "Loss functions forcing cluster separations for multi-class classification using deep neural networks," in *2019 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, November 2019, pp. 1–5.

[25] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[26] S. Wei, M. Yang, J. Qi, J. Wang, S. Ma, and X. Han, "Model-free mle estimation for online rotor angle stability assessment with pmu data," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2463–2476, 2017.

[27] M. Jha, S. Chakrabarti, and E. Kyriakides, "Estimation of the rotor angle of a synchronous generator by using pmu measurements," in *2015 IEEE Eindhoven PowerTech*, June 2015, pp. 1–6.

Authorized licensed use limited to: The George Washington University. Downloaded on August 06,2020 at 18:13:46 UTC from IEEE Xplore. Restrictions apply.