A Unified Online Deep Learning Prediction Model for Small Signal and Transient Stability

Syafiq Kamarul Azman, Younes J. Isbeih, *Member, IEEE*, Mohamed Shawky El Moursi, *Senior Member, IEEE*, and Khaled Elbassioni

***Abstract*—This paper proposes a novel unified prediction approach for both small-signal and transient rotor angle stability as opposed to other studies that have only addressed transient rotor angle stability. Deep learning techniques are employed in this paper to train an online prediction model for rotor angle stability (RAS) using the voltage phasor measurements which are collected across the entire system. As a result, the trained model provides a fast yet accurate prediction of the transient stability status when a power system is subjected to a disturbance. Also, if the system is transiently stable, the prediction model updates the power system operator concerning the damping of low-frequency local and inter-area modes of oscillations. Therefore, the presented approach provides information concerning the transient stability and oscillatory dynamic response of the system such that proper control actions are taken. To achieve these objectives, advanced deep learning techniques are employed to train the online prediction model using a dataset which is generated through extensive time domain simulations for wide range of operating conditions. A convolutional neural network (CNN) transient stability classifier is trained to operate on the transient response of the phasor voltages across the entire system and provide a binary stability label. In tandem, a long-short term memory (LSTM) network is trained to learn the oscillatory response of a predicted stable system to capture the step-by-step dynamic evolution of the critical poorly damped low-frequency oscillations. The superior performance of the proposed model is tested using the New-England 10-machine, 39-bus, IEEE 16-machine, 68-bus, 5-area and IEEE 50-machine, 145-bus test systems and is verified with time domain simulation.**

***Index Terms*—Convolutional neural network (CNN), long short-term memory (LSTM), rotor angle stability prediction, synchronized phasor measurement units (PMUs).**

I. Introduction

Power system stability refers to its ability to return to an acceptable state of equilibrium operating condition after being subjected to a disturbance. The instability problem has been traditionally associated with the rotor angle instability which occurs due to the loss of synchronism [1]. Rotor angle stability can also be classified according to the size of disturbance into small-signal or large-signal (transient) stability. Consequently, small-signal and transient stability are defined as the ability of a power system to retain synchronism when it is subjected to small and large disturbances, respectively.

The dynamic response of a power system is governed by a set of highly nonlinear differential and algebraic equations (DAE) which describe the behavior of the synchronous generators and its associated control systems, loads, renewable power generation, flexible AC transmission devices (FACTs) in addition to the transmission network [2], [3]. When a power system is subjected to small changes, the DAE model can be linearized about the equilibrium point. Small-signal stability is ensured when the change in the electrical torque of a synchronous machine has simultaneously sufficient synchronizing and damping torque components. Therefore, the rotor angle of a synchronous generator can experience a periodic drift and oscillatory response due to the lack of synchronizing and damping torque, respectively. In practical power systems, the small-signal stability problem is largely associated with insufficient damping of oscillations. On the other hand, the transient response of a power system after being subjected to a severe disturbance involves large excursions of the generator rotor angles. As a result, the DAE model cannot be linearized around an operating point and needs to be numerically solved through time domain simulations for each contingency [1]. The transient instability can deteriorate the overall performance of a power system and is considered one of the major causes of power blackouts in the past [4].

Although time domain simulations result in an accurate transient stability assessment (TSA), it requires considerable time and computational efforts especially for large power systems with almost infinite number of contingencies and operating points. In contrast, energy functions are proposed in the literature to analytically assess the transient stability of a system [5], [6]. It is based on calculating the kinetic energy stored on the rotor during fault and the potential energy released during the post-fault period. The deployment of this approach to on-line TSA is constrained by the use of simplified models.

Manuscript received October 22, 2019; revised March 15, 2020; accepted May 25, 2020. Date of publication June 1, 2020; date of current version November 4, 2020. This work was supported by the Khalifa University of Science and Technology under Award No. [CIRA-2018-37] in collaboration with Abu Dhabi Transmission and Dispatch Company (TRANSCO) and Manitoba Hydro International. Paper no. TPWRS-01599-2019. *(Corresponding author: Dr. Mohamed Shawky El Moursi.)*

Syafiq Kamarul Azman, Younes J. Isbeih, and Khaled Elbassioni are with the Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi 127788, United Arab Emirates (e-mail: muhammad.azman@ku.ac.ae; younes.isbeih@ku.ac.ae; khaled.elbassioni@ku.ac.ae).

Mohamed Shawky El Moursi is with the Advanced Power and Energy Center (APEC), EECS Department, Khalifa University, Abu Dhabi 127788, UAE, on leave from the Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt (e-mail: mohamed.elmoursi@ku.ac.ae).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2020.2999102

The traditional system control and data acquisition (SCADA) systems which are installed in power systems generate one measurement every 2 to 10 seconds [7]. This hinders the engineers ability to analyze the dynamics of the system and clouds the view on the whole system. To overcome these issues, the synchrophasor technology is introduced to overcome the drawbacks of SCADA systems. The phasor measurement unit (PMU) can provide measurements in synchronism from different locations and at higher sampling rates in the order of 3–60 measurements per second [8]. As a result, several approaches have been reported in the literature for online TSA.

Burgeoning availability of data and computation have recently reignited interest towards artificial neural networks (ANN) and deep learning which are pivotal in several studies for TSA. Earlier works capitalized on ANN prediction with derived generator rotor angle information for TSA. The first of such studies model TSA through steady-state variables and featured one hidden layer ANN [9] or parallel ANN [10]. Due to the prohibitive nature of computation and scarcity of data at the time, transient dynamics were not studied until later. Some works prefer static variables around the clearing time [11], [12] which ignore the dynamics after clearing the fault. Other works prefer dynamic input features by incorporating PMU measurements such as voltage magnitude and voltage angle [13]–[20]. Furthermore, simulation dynamic information which are not directly measurable with PMUs are also prevalent in the literature; including rotor kinetic energy [21], [22], and active and reactive power [15], [16], [20], [23].

The literature has presented various ANN architectures and schemes to model transient stability. ANN ensembles were deployed to increase prediction performance on a small number of features of rotor angle dynamics at post-fault but depended on hand-engineered features [11]. Similar static features were used to pre-train autoencoders for a convolutional neural network (CNN) fine-tuning but needed separate training episodes for the autoencoder and CNN [12]. Separate training episodes were also found for other autoencoder-based models of TSA for dynamic voltage data [17]. An ensemble of fast-training ANNs was proposed to classify voltage collapse and predict the voltage severity index during the transient phase of a fault, but the prediction only occurs once rather than over the entire transient phase [13].

CNN classifiers are popular among stability classification models for its ability to construct sophisticated features. In [16], a twin convolutional network with a support vector machine classifier output was trained on handcrafted dynamic features and tested on small systems. A multi-size kernel CNN was trained in an ensemble on a large set of raw time domain signals to show that high reliability and security in the prediction is maintained [21]. A novel training framework involving cascaded CNN allows for time adaptivity in the prediction of transient stability after fault clearance [19]. However, the framework consisted of training multiple CNNs which increases training and executions costs.

Variants of recurrent neural networks (RNN) have also been evaluated as classifiers for its ability to capture time domain patterns in input data. Long short-term memory (LSTM) networks are employed to develop a time-adaptive scheme for TSA [14]. The proposed model achieved superior accuracy and was first to formalize a time-dependent metric for TSA. To similar effect, a gated recurrent unit model was developed for TSA assessment but featured a much more comprehensive input signal space [20]. Furthermore, two separate classifiers based on LSTM networks are presented to predict the stability status when a power system is subjected to a disturbance [18]. These classifiers deploy voltage measurements and rate-of-change-of-frequency (RoCoF) during the first five cycles of the post-fault period. A short time-scale LSTM was trained to continuously predict generator trajectory post-fault [22]. This work was the first RNN-based attempt at producing continuous transient predictions albeit for a small time scale. Although fast and accurate predictions are achieved by the proposed models, transient stability assessment is only considered and no attempts have been made to provide more information concerning other dynamic attributes of the system.

Meanwhile, few studies have been reported in the literature on the deployment of machine learning techniques to predict the small-signal stability of a power system [24]–[26]. The training models are constructed using a dataset of different static operating conditions to predict the frequency and damping ratios of low-frequency oscillatory modes. For example, a CNN model is built using the t-distributed stochastic neighbor embedding algorithm which is applied to the electrical distance of power plants for different scenarios [24]. However, the proposed approach does not take into consideration how the frequency and damping ratios of oscillatory modes evolve as time elapses.

To address the aforementioned issues, a unified online approach is proposed in this paper for predicting both small-signal and transient stability using deep learning techniques. The main contributions of the proposed work can be summarized as follows:

1) Unified model is proposed in this paper for predicting both transient and small-signal stability using real-time measurements of the system voltages. The novelty of the proposed model lies in providing an online prediction concerning the damping ratios and frequencies of local and inter-area modes of oscillations as system trajectories evolve with time. Thus, the proposed model can be utilized to synthesize an adaptive supplementary controller to enhance the damping of these oscillations.
2) The prediction model employs an advanced CNN architecture and LSTM network to predict transient and small-signal stability, respectively. A performant, ensemble-like CNN classifier is trained using the transient response of the phasor voltages across the entire system. Furthermore, a LSTM network is constructed to learn the oscillatory response of the stable system over time after classification.

II. Small-Signal and Transient Stability of a Multi-Machine Power System

In this paper, deep machine learning techniques are applied to construct prediction models for both the transient and small-signal stability problems in a multi-machine transmission network. This section describes the mathematical formulation for both problems and provides insight onto the required features to select and train the proper deep machine learning technique.

A. The Transient Stability Problem (TSP)

When a power system is subjected to a disturbance, its dynamic response is governed by a set of differential and algebraic equations which can be expressed in the compact form as

$$\dot{x} = h(x, y) \tag{1}$$

$$0 = g(x, y) \tag{2}$$

where x and y are the state and algebraic variables. In addition, h and g represent the vectors of differential and algebraic equations. Solving for (1) and (2) yields time-varying trajectories of the state variables x such as the rotor angles and frequencies, and the algebraic variables y such as bus voltages and active power injections. To this end, numerical techniques such as the trapezoidal method are employed to discretize the set of differential equations in (1). The resulting algebraic equations are then solved simultaneously with the rest of algebraic equations (2) using Newton's method at each time step. The transient stability is assessed by observing the dynamic trajectories over the simulation time window. This method results in accurate assessment of the transient stability for a specific contingency. For large power systems with tens of machines, hundreds of transmission lines and thousands of buses, the transient stability assessment of the infinite number of contingencies represents a tedious and time consuming task. Therefore, a more comprehensive approach is required to generalize the transient stability assessment without the need to solve the entire DAE model for each contingency.

B. The Small-signal Stability Problem (SSP)

The set of differential and algebraic equations presented in (1)–(2) can be linearized about an equilibrium point for small disturbances as shown in (3)–(4).

$$\Delta\dot{x} = A\Delta x + B\Delta y \tag{3}$$

$$0 = C\Delta x + D\Delta y \tag{4}$$

$$A = \frac{\partial h}{\partial x}, B = \frac{\partial h}{\partial y}, C = \frac{\partial g}{\partial x}, D = \frac{\partial g}{\partial y} \tag{5}$$

The linearized model expressed in (3)–(4) is used to study the small-signal or local stability of a power system about an equilibrium point when it is subjected to small perturbation. This objective is achieved using the Lyapunov's first method where the roots of the characteristic equation (eigenvalues) need to be calculated as follows [1]

$$\det(A_{sys} - \boldsymbol{\lambda} I) = 0 \tag{6}$$

where $A_{sys} = A - B(D^{-1})C$ and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]$. The computed eigenvalues correspond to either real or complex ones where they result in non-oscillatory and oscillatory response, respectively. In addition, complex eigenvalues occur in conjugate pairs where each pair represents an oscillatory mode [1]. The small-signal stability of a practical power system is largely related to insufficient damping of oscillations. In addition, the stability of local or machine-system and inter-area modes of oscillations specifically represents a crucial concern to power system operation. Local modes are observed when synchronous machines oscillate against each other in one area at a frequency within the range of 1-2 Hz. The inter-area mode of oscillations are triggered over a large geographical area where one or group of generators oscillate against a group of distinct generators at a frequency of 0.3-1 Hz. The presence of local and inter-area modes constrains the power transfer capability of a transmission network since their magnitude can grow if there is insufficient damping. Therefore, real-time monitoring of low-frequency power system oscillations is mandatory for secure operation of a power system. However, the traditional approach for assessing small-signal stability requires the complete mathematical DAE model which is difficult to develop for practical power systems. To resolve this issue, the synchrophasor measurements collected from PMUs contain useful information concerning these oscillations and can be employed to provide estimates for their overall damping. Consequently, data oriented techniques provide an effective and successful framework to construct learning models for predicting small-signal stability which will be thoroughly described in Section III.

III. The Online Prediction Model for Transient and Small-Signal Rotor Angle Stability

Transient stability assessment (TSA) is a high dimensional time-dependent problem and contemporary solutions to TSA almost exclusively involve data-driven methods and ANNs. Formally, a neural network, H_ψ, parameterized by weights ψ, learns from the database $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{|\mathcal{D}|}, \mathbf{y}_{|\mathcal{D}|})\}$, to map $H_\psi(\mathbf{x}_i) = \mathbf{y}_i$ by minimizing a loss function that characterizes the error of the mapping. Using gradient descent, neural networks iteratively improve on this mapping by updating its weight parameters. In TSA, $\mathbf{x}_i$ denotes a set of features of the power system and $\mathbf{y}_i$ denotes some output mapping of $\mathbf{x}_i$. TSP is a TSA problem that is framed as a classification task where the objective is to categorize a signal into classes ($\mathbf{y}_i \in \mathbb{N}$) while SSP is framed as a regression task whose solution is to approximate continuous values ($\mathbf{y}_i \in \mathbb{R}$). We briefly describe the mathematics and rationale behind each class of deep learning model in the following subsections.

A. CNN Classifiers for TSP

CNNs efficiently learn time- or space-independent features from data. Instead of a densely connected network, CNNs deploy kernel convolution which emulates the response of an individual neuron to local stimuli thus reducing the learnable parameters [27]. While CNNs typically have sparse interactions at each layer, the receptive field broadens with depth [28] meaning that signatures that appear later could be detected earlier. CNN operates upon the convolution operation which is a blending of an input vector $\mathbf{x}$ as it is strided over a variable kernel $\mathbf{w}$

$$\mathbf{h}_i^{(\phi)} = (\mathbf{w} \otimes \mathbf{x})_i = \sum_{j=1}^{n} \mathbf{x}_{i+j}\mathbf{w}_j + \mathbf{b} \quad \forall i \in \{1, 2, \ldots, m\}. \tag{7}$$

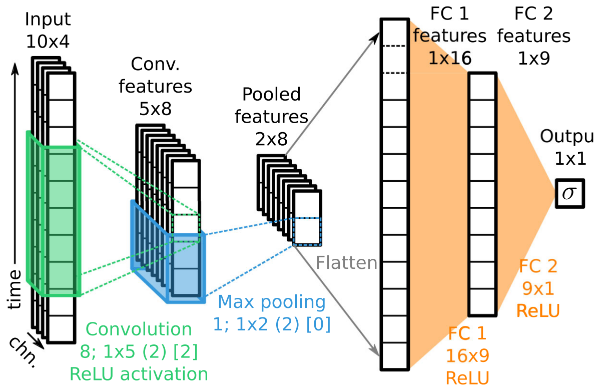


Fig. 1. An example 1-dimensional CNN classifier with an input of 10 time steps and 4 channels.

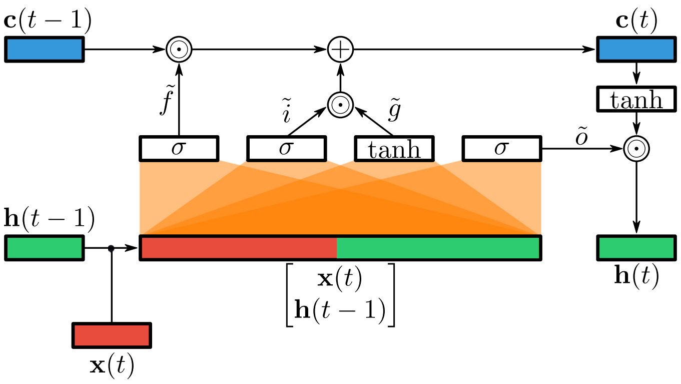


Fig. 2. Flow of input, hidden and cell data in the LSTM unit.

In practice, the convolution is substituted for the cross-correlation operation, after which a nonlinearity is applied, commonly the ReLU activation for its rapid convergence [29]. For multi-channeled signals, the convolution operates over all channels simultaneously thus combining the dynamics of many signals to create one new signal containing useful features. CNNs feature pooling layers in between convolutions to shrink the original input into an approximate representation allowing for small local invariance to be modeled [28]. After the convolutional layers, one or more fully connected (FC) layers are added to the network to combine the kernel features together. Binary CNN classifiers have an output layer containing a sigmoid activation function, σ, to threshold activations between 0 and 1. A visual representation of a CNN including the described operators can be seen in Fig. 1. The convolution and max pooling operations have user-definable hyperparameters such as the number of kernels, k, the kernel size, $1 \times |w|$, the stride size, (s), and the padding size, $\{p\}$. For example, a convolution with 8 units of 1×5 kernel, with stride of 2 and padding size of 2 is denoted as "8; 1×5 (2) {2}".

CNN classifiers are well-suited for TSP as convolution kernels learn from raw PMU data to produce prediction (stable or unstable) in the form of a probability. Formally, a binary CNN classifier learns by maximizing the likelihood of the correct label as a Bernoulli random variable, $\Pr[Y = y \mid H_\phi(\mathbf{x})]$, by updating the parameters, ϕ. The likelihood is maximized by minimizing the following loss function

$$L_{\mathrm{BC}} = -\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i), \tag{8}$$

where $\hat{y}_i = H_\phi(\mathbf{x}_i)$. L_{BC}, also known as the binary cross-entropy loss, is the negative log-likelihood averaged over all i contingencies.

B. LSTM Networks for SSP

LSTM networks are variants of RNNs which have the ability to recall past information in ordered time series data. The network learns by encoding step-wise time domain inputs into internal hidden states which are carried forward to the next time step. RNNs update their parameters via backpropagation through time which propagates gradients back along the sequence over which the RNN unfolds. However, RNNs suffer from the vanishing or exploding gradient problem during weight updates for highly protracted sequences; a problem which LSTM networks remedy [30].

LSTM networks contain LSTM units which comprise of three "gates": input, forget and output, and a cell state in addition to a hidden state. The gates of an LSTM unit are four interacting FC ANNs which determine whether information in the hidden state and current inputs should be forgotten or remembered [30]. For an LSTM unit containing the weight parameters $\mathbf{W}^{(x*)}$, where $*$ denotes the symbol for one of the four ANNs of the LSTM units ($\tilde{f}$ for forget, $\tilde{i}, \tilde{g}$ for input, and $\tilde{o}$ for output), the LSTM unit updates its cell state $\mathbf{c}(t)$ and hidden state $\mathbf{h}^{(\theta)}(t)$ according to the following formulation:

$$\tilde{\mathbf{f}} = \sigma(\mathbf{W}^{(x\tilde{f})}\mathbf{x}(t) + \mathbf{b}^{(x\tilde{f})} + \mathbf{W}^{(h\tilde{f})}\mathbf{h}^{(\theta)}(t-1) + \mathbf{b}^{(h\tilde{f})}) \tag{9}$$

$$\tilde{\mathbf{i}} = \sigma(\mathbf{W}^{(x\tilde{i})}\mathbf{x}(t) + \mathbf{b}^{(x\tilde{i})} + \mathbf{W}^{(h\tilde{i})}\mathbf{h}^{(\theta)}(t-1) + \mathbf{b}^{(h\tilde{i})}) \tag{10}$$

$$\tilde{\mathbf{g}} = \tanh(\mathbf{W}^{(x\tilde{g})}\mathbf{x}(t) + \mathbf{b}^{(x\tilde{g})} + \mathbf{W}^{(h\tilde{g})}\mathbf{h}^{(\theta)}(t-1) + \mathbf{b}^{(h\tilde{g})}) \tag{11}$$

$$\tilde{\mathbf{o}} = \sigma(\mathbf{W}^{(x\tilde{o})}\mathbf{x}(t) + \mathbf{b}^{(x\tilde{o})} + \mathbf{W}^{(h\tilde{o})}\mathbf{h}^{(\theta)}(t-1) + \mathbf{b}^{(h\tilde{o})}) \tag{12}$$

$$\mathbf{c}(t) = \tilde{\mathbf{f}} \odot \mathbf{c}(t-1) + \tilde{\mathbf{i}} \odot \tilde{\mathbf{g}} \tag{13}$$

$$\mathbf{h}^{(\theta)}(t) = \tilde{\mathbf{o}} \odot \tanh(\mathbf{c}(t)) \tag{14}$$

where $\odot$ is the Hadamard (element-wise) product. The structure of an LSTM unit is visible in Fig. 2.

LSTM networks are ideal for converting sequential data in real-time which is fitting for SSP: LSTM networks can learn, using data from different operating contingencies, to transform patterns in observed time-domain signals into the oscillatory modes of the system at any time instance; formally, $\hat{\boldsymbol{\lambda}}(t) = H_\theta(\mathbf{x}(t))$. As past predictions affect future ones, the LSTM network must minimize the mean squared error (MSE) between $\hat{\boldsymbol{\lambda}}(t)_i$ and actual values, $\boldsymbol{\lambda}(t)_i$, post fault clearance until the end

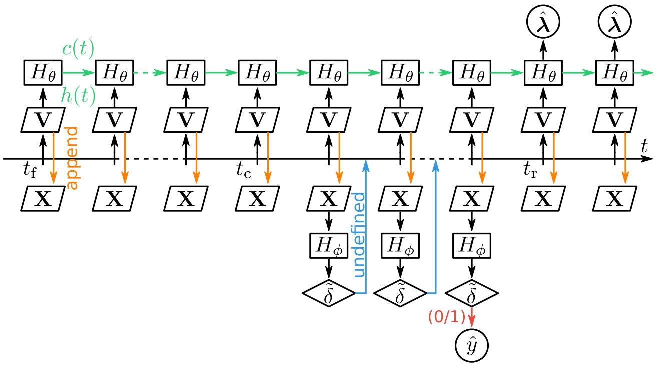


Fig. 3. The proposed TSA system showing the inputs, actions and outputs over a fault duration.

of simulation, U, and over all i contingencies:

$$L_{\mathrm{MSE}} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\sum_{t=1}^{U}(\boldsymbol{\lambda}(t)_i - \hat{\boldsymbol{\lambda}}(t)_i)^2. \tag{15}$$

The weight parameters of the LSTM network, θ, comprise the weights of all FC networks and the LSTM units. Furthermore, greater sophistication can be attained from LSTM networks by stacking LSTM units such that the hidden state of one LSTM unit feeds into the input of another LSTM unit. Commonly, FC layers are added to the output of LSTM units.

C. Proposed Combined Approach to TSP and SSP

This work proposes the combination of a CNN and a LSTM network in tandem to solve TSP and SSP in an online setting after offline training. We devise a process that feeds regularly sampled PMU data into neural networks for TSA (depicted in Fig. 3). At any time, t, the system collects phasor voltage and voltage angle data, $\mathbf{V}(t)$, from all available buses. To supply the appropriate input data for the CNN, a fixed size matrix $\mathbf{X}$ is maintained continually by appending $\mathbf{V}(t)$ into $\mathbf{X}$ in a first-in-first-out queue manner. When a fault occurs at t_f, the system triggers the execution of the LSTM network (H_θ) to continually "warm up" its internal states (green arrows in Fig. 3) by collecting data during the fault stage. Once the fault is cleared at t_c, the CNN (H_ϕ) is executed until a reliable stability prediction is achieved at t_r, or a deadline is reached at T. Reliable stability predictions are assessed based on a user-defined tolerance value, $0 < \Delta < 0.5$, adapted from a time-adaptive TSP study in the literature [14]. The reliability, $\tilde{\Delta}$, is calculated based on the output prediction $\hat{y}(t) = H_\phi(\mathbf{X}(t))$:

$$\tilde{\Delta}(t) = \begin{cases} 1 & \text{if } \hat{y}(t) > 1-\Delta \\ 0 & \text{if } \hat{y}(t) < \Delta \\ \text{undefined} & \text{otherwise} \end{cases} \tag{16}$$

In uncertainty, the CNN will continue to operate on the following time step (shown as blue arrows in Fig. 3). At T, Δ is automatically set to 0.5. Once reliable (at time t_r), if the CNN has determined that a stable signal will follow from the clearance ($\tilde{\Delta} = 1$), the CNN stops operating and the LSTM network will then proceed to operate. Note that in Fig. 3 that t_r appears after several steps of the CNN, but can in fact be as soon as one time step after $t_\mathrm{c}+1$. The LSTM network continually predict the eigenvalues, $\hat{\boldsymbol{\lambda}}(t) = H_\theta(\mathbf{V}(t))$, of the system until it is manually terminated.

IV. Implementation of CNN and LSTM Networks for Predicting Transient and Small-Signal Stability

A. Data Preparation for Offline Training of CNN

The CNN commences when the fault is cleared and operates on a window of data rather than single vectors. This means that the earliest data that must be captured for the CNN should be a matrix where the latest input vector is at $t_\mathrm{c}+1$. With that in mind, we describe a data augmentation step by extracting slices of data from a window of up to 20 cycles (0.2 s) centered around $t_\mathrm{c}+1$ to capture the voltage signals until the first 10 cycles after t_c for training. A typical extracted data matrix appears as

$$\mathbf{X} = [\mathbf{V}(t_c-8)^\intercal, \ldots, \mathbf{V}(t_c)^\intercal, \ldots, \mathbf{V}(t_c+10)^\intercal] \tag{17}$$

A sliding window of width 10 and stride length of 1 is used to extract 10 smaller submatrices of size $2B \times 10$ from the original $2B \times 20$ matrix for each contingency. To evaluate the time adaptivity, the testing data encompassed a slightly different time scale. We required that a test instance had at least 20 time steps following t_c to evaluate the CNN on untrained horizons as it was conducted in [14].

B. Data Preparation for Offline Training of LSTM Networks

The objective of SSP is to predict $\boldsymbol{\lambda}$ for contingencies that are deemed stable by the CNN. Therefore, the predictions should occur as soon as the CNN outputs a confident value of $\hat{y}$. As there can be no foresight of the time at which the CNN terminates, a conservative measure would be to commence prediction as early as the fault clearance. Thus, for each contingency, we extracted data for the last 850 time steps (equalling 8.5 s) as training data after accounting for initialization, clearing times and simulation lags. A training case for the LSTM network consists of the voltage and voltage angles and system oscillatory modes

$$\mathbf{V}(t) = \{V_b(t), \angle V_b(t) \mid b = 1, \ldots, B\} \tag{18}$$

$$\tilde{\boldsymbol{\lambda}}(t) = \{\lambda_l(t) \mid l = 1, \ldots, 2L\} \tag{19}$$

$$\mathbf{X} = [\mathbf{V}(t)^\intercal \mid t = t_\mathrm{c}, \ldots, T] \tag{20}$$

$$\boldsymbol{\Lambda} = [\tilde{\boldsymbol{\lambda}}(t)^\intercal \mid t = t_\mathrm{c}, \ldots, T] \tag{21}$$

where, at time t, $V_b(t)$ and $\angle V_b(t)$ are the voltage magnitude and angle for the bus b. For the output $\tilde{\boldsymbol{\lambda}}(t)$, we note that Eqn. (15) is ill-fitted as a loss function for complex numbers as they do not occupy Euclidean space. However, to the best of our knowledge, no complex counterpart of the MSE loss function is available in the literature. In this light, we propose a disjoint approach by separating the real and imaginary part of $\boldsymbol{\lambda}$ as $\tilde{\boldsymbol{\lambda}} = \{\Re(\lambda_1), \ldots, \Re(\lambda_L), \Im(\lambda_1), \ldots, \Im(\lambda_L)\}$, effectively substituting Eqn. (15) in place of a complex one.

TABLE I
THE PROPOSED CNN ARCHITECTURES IN THIS WORK

	Model	
Layer	GCNN	GINN
Input	Input signal $(2B \times 10)$	
Conv 1	$K^{(1)}$; (1×5) (2) {2}	64 kernels Inception
Conv 2	$K^{(2)}$; (1×5) (2) {2}	128 kernels Inception
	Flatten	
FC 1	256 units	128 units
Output	1 unit Sigmoid	

TABLE II
OPTIMIZATION PARAMETERS FOR TRAINING THE NN MODELS

Model	Opt.	η	ϵ	γ	E
CNN	SGDN	0.01	10^{-4}	0.1	180
LSTM	Adam	0.003	10^{-1}	0.3	240

C. CNN Architectures: GCNN and GINN

The CNN architecture presented in this work aims to minimize computation such that the execution time does not exceed the data sampling rate. To this end, we propose a classical CNN network architecture (GCNN) with two convolutional layers and one FC layer. The number of kernels in the first convolutional layer is equal to double the number of input channels for a system ($K^{(1)} = 2 \times 2B$). The second convolution layer has double the number of kernels in the first convolution layer with a cap of 512 kernels ($K^{(2)} = \max(2K^{(1)}, 512)$) to minimize trainable parameters. Our proposed CNN deploys padded and strided convolutions, as opposed to pooling layers, to quicken the convolution operation step while maintaining performance [31].

In search of a performant architecture, we pursued a deeper and more varied feature mapping architecture. Specifically, we substituted the 1-dimensional version of the Inception block that is introduced in [32] instead of the traditional convolutions in GCNN. The rationale of the Inception block is akin to packaging an ensemble of CNNs with different kernel sizes into one layer and are trained together rather than in separate episodes. For the filter sizes, we propose 1×3, 1×5 and 1×7 kernels to capture salient features of the input signal, following a 48-kernel 1×1 bottleneck convolution. This CNN variant, called GINN, contains four layers of convolutions divided over two Inception blocks. ReLU is selected as the activation function for all layers with the exception of the output layer.

Dropout is applied throughout the CNN to alleviate overfitting during training, with a rate of 0.2 between convolutions and 0.5 between the FC and output layers [33]. The training parameters are displayed in Table II with the following keys: initial learning rate η, weight decay ϵ, learning rate decay factor γ, and number of epochs E. Nesterov stochastic gradient descent (SGDN) is used as the optimizer with momentum parameter $\nu = 0.9$ and learning rate is decayed after 80, 120, 160 epochs. We tabulate the architecture of GCNN and GINN in Table I.

D. LSTM Implementation

We opted for a stacked LSTM network architecture with two LSTM layers to model the highly nonlinear dynamics in the system. The first LSTM unit captures the input signal from raw data and feeds its hidden state into the second LSTM unit before connecting to the output layer with L units. We experimented with 64, 128 and 256 hidden units to evaluate the performance of different hidden sizes on overfitting. We aptly named these LSTM networks as SSLSTM, noting the purpose of the LSTM networks as SSP solutions. Our preliminary trials showed poor performance when dropout is introduced and thus we have opted for a strong weight decay to regularize training. Due to the instability of SGDN in LSTM training, the Adam optimizer [34] is deployed instead with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$). The learning rate is decayed after 120, 200, 220 epochs.

E. Evaluating GCNN and GINN

The binary predictions of the CNN models are assessed based on the accuracy and, additionally, as we considered the time adaptivity of the CNN, we adopted the average response time (ART) presented in similar works [14], [35]. Recall that training was conducted on the sliced matrices as described in Section IV-A; however, we evaluate the CNN performance based on the scheme presented in Section III-C and we set $T = 20$ cycles (0.2 s) to identify models' performance outside its training window. The ART is defined as the average number of reliable predictions of test instances, $\mathrm{R}(t)$, made at time step t from t_c until T; formally

$$\mathrm{ART} = \frac{\sum_{t=1}^{T-t_\mathrm{c}} t \times \mathrm{R}(t)}{\sum_{t=1}^{T-t_\mathrm{c}} \mathrm{R}(t)}. \tag{22}$$

F. Evaluating SSLSTM

The performance of SSLSTM is measured using the root mean squared error (RMSE) metric and mean arctangent absolute percentage error (MAAPE) metric [36]. While the literature [22] has opted for mean absolute percentage error (MAPE) to quantify LSTM network prediction errors, due to the near-zero values of $\Re(\boldsymbol{\lambda})$ the MAAPE is a preferable measure while maintaining the same semantic value [36]. These metrics are applied on to the damping ratio $\boldsymbol{\xi}$ which is reconstructed from $\boldsymbol{\lambda}$. The average errors are reported over all cases in the test set. $\boldsymbol{\xi}$ is reconstructed and evaluated based on the expressions

$$\xi_j(t) = \frac{-\Re(\lambda_j(t))}{\sqrt{\Re(\lambda_j(t))^2 + \Im(\lambda_j(t))^2}} \tag{23}$$

$$\mathrm{RMSE}(\xi_j) = \sqrt{\frac{1}{U} \sum_{t}^{U} (\hat{\xi}_j(t) - \xi_j(t))^2} \tag{24}$$

$$\mathrm{MAAPE}(\xi_j) = \frac{100\%}{U} \sum_{t}^{U} \arctan\left(\frac{|\hat{\xi}_j(t) - \xi_j(t)|}{\xi_j(t)}\right) \tag{25}$$

where $\hat{\boldsymbol{\xi}}$ correspond to the damping ratio reconstructed from $\hat{\boldsymbol{\lambda}}$ and U is the timespan of the testing window which is equivalent to the training window (8.5 seconds). Well-fitted models should provide RMSE and MAAPE values that are close to zero.

V. Case Studies

In this section, the effectiveness of the proposed approach is tested using the New England 10-machine, 39-bus system [5], 16-machine, 68-bus, 5-area system [37] and 50-machine, 145-bus system [38]. For these systems, time-domain simulations are carried out using the power system analysis toolbox (PSAT) [39]. The generator rotor angles, voltage magnitudes and angles at all buses are recorded in addition to the modes of oscillations. In addition, these simulations are run for 10 s and at a time step of 0.01 s. The voltage magnitudes and angles which are provided by the PMU units are employed in the training and testing phases. The rotor angles of the synchronous generators are employed to provide stability labels for the CNN transient prediction model. The transient stability status of a contingency is determined by evaluating the relative rotor angles of generators against 360° threshold which has been widely used in the literature [14], [21], [35], [40]. The edge cases are corrected with active machine learning. In addition, the dynamic trajectories of the oscillatory modes are utilized to train the LSTM network. The generated cases are divided into training and testing sets in the ratio of $3:1$, concurring with other similar works [14], [35].

A. The New England 10-Machine, 39-Bus Test System

The 4th order model is used to describe the dynamics of all synchronous generators except generator 1 for which the 3rd order model is used [39]. In addition, type II exciters were implemented for all generators except generator 10 for which manual excitation is assumed [39].

Time-domain simulation of post-contingency power system dynamics is used to generate the training and testing cases. Three-phase to ground faults are applied at different locations since they represent the most severe disturbances among other types of faults such as single line or double line to ground ones. The loading level of the system is varied from 60% up to 100% at an increment of 5%. The maximum loading condition at 100% represents 122.19% of the base case loading presented in [5]. For each loading level, a three-phase fault is applied at each bus and transmission line where the faults were located: at $20\%, 40\%, 60\%$ and 80% of the whole line. In addition, the fault clearing time is varied from 100 ms up to 400 ms [14], [43]. Consequently, 8,316 $(4 \times 9 \times (39 + 4 \times 48))$ TSA contingency cases were generated with the previously described conditions.

1) CNN Evaluation: From the generated cases, 3,318 stable cases and 2,920 unstable cases were present in the training set. Out the 2,920 unstable cases, 2,910 cases contained at least 1 cycle of data following t_c. Overall, the augmentation process generated 61,871 data points for training. In the testing set, 1,984 contingencies with at least 20 cycles of data after t_f are collected for evaluation. Fig. 4 displays the $\mathrm{R}(t)$ and accuracy of GCNN and GINN.

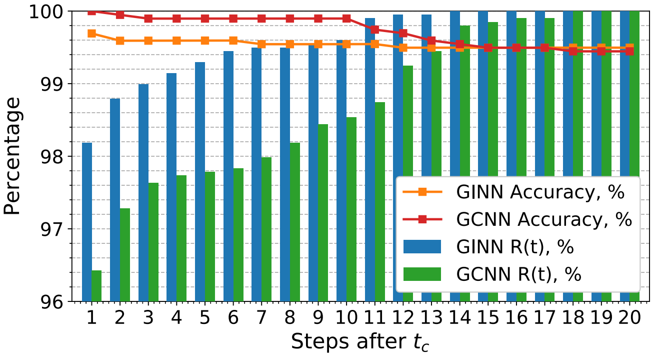


Fig. 4. Comparing GCNN and GINN performance on $\mathrm{R}(t)$ and accuracy at $\Delta = 0.08$.

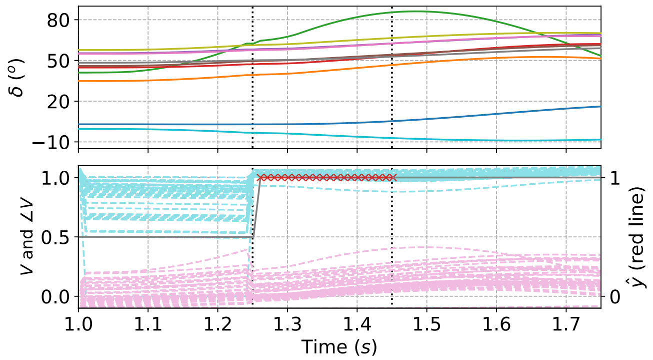


Fig. 5. The rotor angle values time-aligned with the PMU data inputs and GINN predictions for a New England 39-Bus system test case.

At the best setting, $\Delta = 0.08$, GINN reliably predicts 1,948 (98.18%) contingencies with 99.69% accuracy in the first cycle before completing all remaining cases in 13 cycles at a final accuracy of 99.55%. As for GCNN, 1,913 contingencies are reliably labelled at 100% accuracy in the first cycle but significantly depreciate over 17 cycles to 99.45%. The conservative accuracy of GINN is attributed to its deeper features sets. Note that GCNN suffers a sizeable decrease in accuracy after 10 time steps, showing its weakness outside its training window. Conversely, GINN maintains its accuracy by having more complex feature sets to operate upon. Fig. 5 shows the dynamic response of the relative rotor angles (δ) of the synchronous generators at 60% load when a three-phase to ground fault is applied at Bus 32 at time $t = 1$ s and is cleared after 200 ms. GINN successfully predicts the transient stability status for this contingency immediately at $t = 1.26$ s considering a time lag of 0.06 s in the simulation. We remark that the GCNN model comprises a much larger parameter space at 385,053 parameters while GINN has only 227,393 parameters despite having a deeper architecture which is also true for the GCNN deployed in the larger systems.

The performance of the proposed models is compared against that of other models which are reported in the literature for the New England 39-Bus system as shown in Table III (extended from [14]). From the results, it can be concluded that our proposed models, in comparison to the state-of-the-art, are competitive in terms of accuracy of prediction and prevailed in ART. Although our method is slightly slower in ART in comparison to the stacked GRU model of [20], that work incorporated a larger input feature space compared to the voltage and voltage angles presented in this work and in [14].

TABLE III
PERFORMANCE INDEX COMPARISON WITH OTHER TSP WORKS INCORPORATING ART

	New England 39-Bus		IEEE 68-Bus		IEEE 50-Machines	
Method	ART	Accuracy	ART	Accuracy	ART	Accuracy
Our method, GCNN	1.253 cycles	99.45%	1.411 cycles	97.22%	1.442 cycles	98.31%
Our method, GINN	**1.106 cycles**	99.55%	1.339 cycles	98.40%	**1.312 cycles**	98.46%
CNN-based model [19]	1.111 cycles	99.60%				
CNN-based model [21]	0.05 seconds	98.63%				
GRU-based model [20]	*1.08 cycles*	*100%*				
LSTM-based model [14]	1.448 cycles	**100%**			2.047 cycles	**99.98%**
ELM-based model [35]	1.4 cycles	99.10%			2.8 cycles	99.70%
SVM-based model [40]	4 cycles	100%				
Template matching [41]	6 cycles	100%				
Fuzzy decision tree [42]	1-2 seconds	95%				

TABLE IV
RECONSTRUCTED $\boldsymbol{\xi}$ RMSE FOR DIFFERENT HIDDEN UNITS.

	RMSE								
	New England 39-Bus			IEEE 68-Bus			IEEE 50-Machine		
	64	*128*	*256*	*64*	*128*	*256*	*64*	*128*	*256*
ξ_1	0.0027	0.0027	0.0032	0.0053	0.0052	0.0041	0.0006	0.0005	0.0004
ξ_2	0.0034	0.0034	0.0035	0.0028	0.0026	0.0023	0.0002	0.0002	0.0002
ξ_3	0.0033	0.0036	0.0028	0.0054	0.0047	0.0045	0.0005	0.0005	0.0004
ξ_4	0.0024	0.0020	0.0024	0.0007	0.0007	0.0005	0.0016	0.0014	0.0014
ξ_5	0.0075	0.0051	0.0065	0.0023	0.0022	0.0020	0.0017	0.0014	0.0014
ξ_6	0.0072	0.0057	0.0067				0.0002	0.0001	0.0001
ξ_7	0.0039	0.0031	0.0030						
ξ_8	0.0183	0.0170	0.0160						
ξ_9	0.0624	0.0600	0.0630						

TABLE V
FALSE POSITIVE AND FALSE NEGATIVE CASES PREDICTED BY GINN IN THE TESTING SET

System	FPR	False positives	False negatives
39-Bus	0.59%	5 (0.25%)	6 (0.30%)
68-Bus	2.33%	15 (0.89%)	25 (1.47%)
50-Machine	0.83%	3 (0.14%)	25 (1.39%)

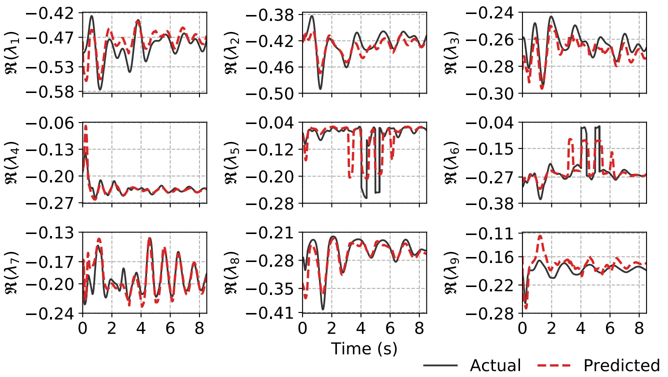


Fig. 6. Actual and SSLSTM-128 predictions of $\Re(\lambda_j)$ for an arbitrary test case in the New England 39-Bus System.

2) LSTM Evaluation: The LSTM network is only trained on stable contingencies and therefore the training database contained 3,318 training samples. For testing, the data is similarly generated and yielded 1,130 instances. As summarized in Tables IV andVI, SSLSTM-128 result in the lowest errors of reconstructed $\boldsymbol{\xi}$. It is likely that SSLSTM-64 is underfitting the data as the errors are higher than SSLSTM-128 and, conversely, SSLSTM-256 has slightly overfitted the training data. Additionally, SSLSTM-128 is almost 2% better than the other models on average. To the best of our knowledge, our method is the first to produce continuous prediction of small-signal stability and over a significantly longer time scale compared to the current literature [22].

The real-time predictions of the low-frequency power system oscillations which are obtained using the SSLSTM-128 model are shown along with their estimated damping ratios in Figs. 6, 7 and 8. It is found that the test system exhibits nine modes of oscillations where the eigenvalues λ_{1-7} and λ_{8-9} correspond to local and inter-area modes, respectively. In general, the LSTM predictions (red dashed lines) match closely the actual simulation values in terms of oscillation patterns. The $\Im(\lambda_j)$ values are usually underestimated which, in turn, overestimates the ξ_j values. Large overestimation are ill-received as it misinforms the damping characteristics of the system. Sudden dips are observed in $\Re(\lambda_5)$ and $\Re(\lambda_6)$ around 3.5 s and 6 s for this particular test case. This behavior is attributed to the lack of data during training and hence the LSTM network responds to the dynamics of the input voltage values in this way. However, SSLSTM is also able to capture the fluctuations that appear in the actual simulation around 4 s and 5 s.

TABLE VI
RECONSTRUCTED $\boldsymbol{\xi}$ MAAPE FOR DIFFERENT HIDDEN UNITS

	MAAPE, %								
	New England 39-Bus			IEEE 68-Bus			IEEE 50-Machine		
	64	*128*	*256*	*64*	*128*	*256*	*64*	*128*	*256*
ξ_1	4.30	4.36	5.61	3.52	3.16	2.50	0.41	0.34	0.31
ξ_2	5.25	5.01	5.59	2.33	2.17	1.85	0.26	0.19	0.19
ξ_3	6.98	8.37	5.93	5.44	4.91	4.57	0.75	0.68	0.62
ξ_4	5.96	4.62	5.58	1.22	1.11	0.91	2.03	1.69	1.61
ξ_5	28.9	17.5	25.7	5.23	4.88	4.43	1.98	1.71	1.64
ξ_6	19.2	17.2	19.2				0.16	0.12	0.10
ξ_7	6.44	4.90	4.65						
ξ_8	8.18	5.81	6.84						
ξ_9	45.9	40.6	46.5						

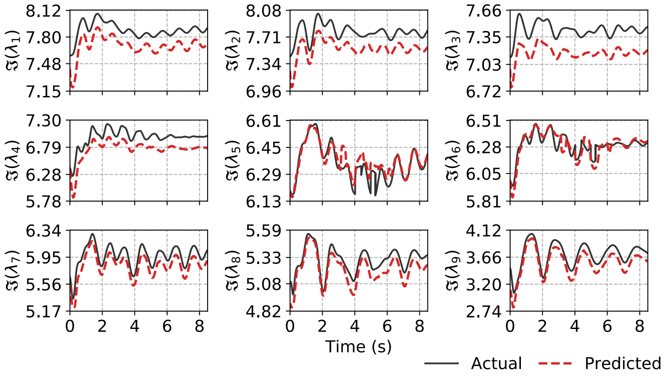


Fig. 7. Actual and SSLSTM-128 predictions of $\Im(\lambda_j)$ for an arbitrary test case in the New England 39-Bus System.

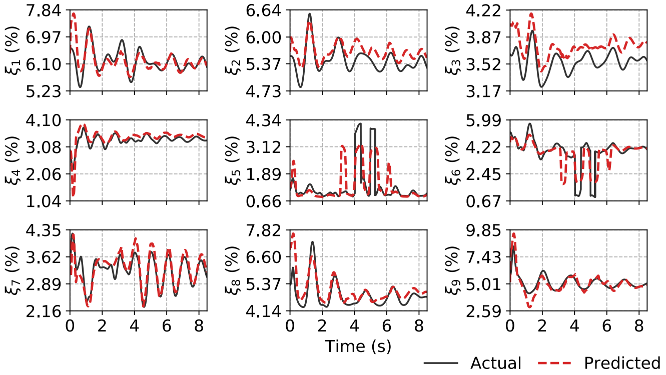


Fig. 8. Actual and SSLSTM-128 reconstructed values of ξ_j for an arbitrary test case in the New England 39-Bus System.

The errors show that ξ_9 was the oscillatory mode that was most difficult to predict as it has the largest error. In other cases, we observed large fluctuations in rapid succession in ξ_9 which the SSLSTM best approximated by generating smooth curves through the fluctuation. The smooth approximation is associated to the tanh and sigmoid functions as part of the LSTM architecture [30]. While ξ_5 and ξ_6 had low RMSE, the MAAPE was much larger meaning it was more challenging to predict than others. This is due to the large sudden shifts in the real part of $\boldsymbol{\lambda}$ that are difficult to fit and is notoriously apparent in ξ_5 and ξ_6 (in Fig. 6) but also sometimes existing in ξ_4. Upon closer inspection, for some cases, the predictions for ξ_8 displayed large erroneous predictions within the first 1 s of the signal but maintained small and stable predictions afterward. We suspect this is the culprit of the large average RMSE but much lower MAAPE compared to ξ_5 and ξ_6.

B. The IEEE 16-Machine, 68-Bus Test System

The sub-transient model with four equivalent rotor coils are used to describe the dynamic performance of all generators in the system. In addition, generators G_{1-8} and G_{10-12} are equipped with IEEE standard DC exciter (DC4B); G_9 is equipped with fast static excitation (ST1A), while the rest of the generators G_{13-16} have manual excitation. Furthermore, the DC and fast static excitation systems are equipped with conventional power system stabilizers which comprise of two lead-lag compensation and washout filter blocks. The test data for this system is directly taken from [37].

The training and testing dataset is prepared by conducting several time-domain simulations for different loading conditions, fault locations and clearing times. The active and reactive power of all loads have been changed randomly and independently within 80%–120% of the basic loading level. In addition, contingencies are created by applying three-phase to ground faults at any bus and at different randomly selected locations within the transmission lines. These faults are cleared after a random time within 0.06 to 0.5 seconds.

1) CNN Evaluation: From the generated cases, 3,129 stable cases and 4,371 unstable cases were present in the training set. Out the 4,371 unstable cases, 2,007 cases contained at least 1 cycle of data following t_c. Overall, the augmentation process generated 51,360 data points for training. In the testing set, 1,692 contingencies with at least 20 cycles of data after t_f are collected for evaluation.

At the best setting, $\Delta = 0.05$, GINN reliably predicts 1,513 (98.87%) contingencies with 98.87% accuracy in the first cycle before completing most of the remaining cases in 20 cycles leaving 25 cases to be predicted at $T_{\max}$ producing a final accuracy of 98.40%. As for GCNN, 1,536 contingencies are reliably labelled at 98.76% accuracy in the first cycle but significantly depreciate over 20 cycles to 97.22% leaving 20 cases to be

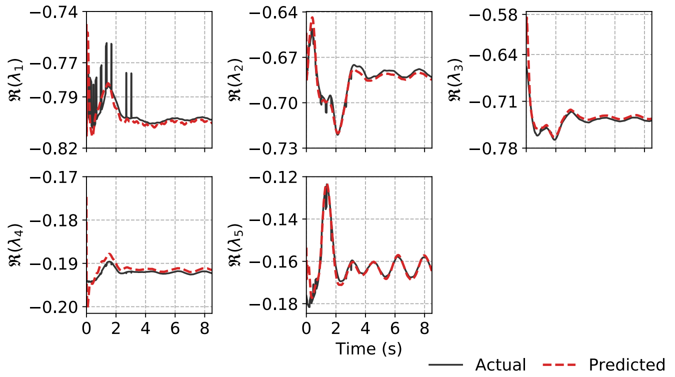


Fig. 9. Actual and SSLSTM-256 predictions of $\Re(\lambda_j)$ for an arbitrary test case in the IEEE 68-Bus System.

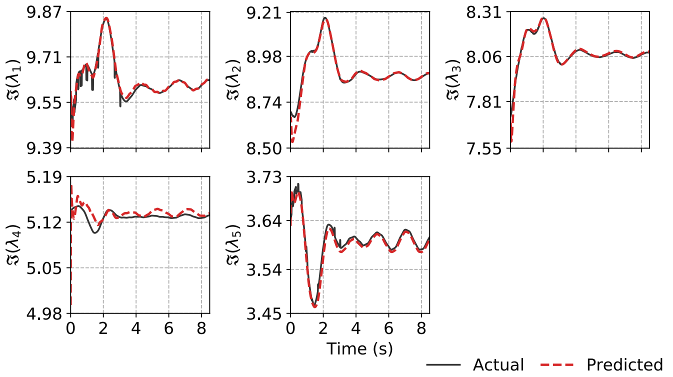


Fig. 10. Actual and SSLSTM-256 predictions of $\Im(\lambda_j)$ for an arbitrary test case in the IEEE 68-Bus System.

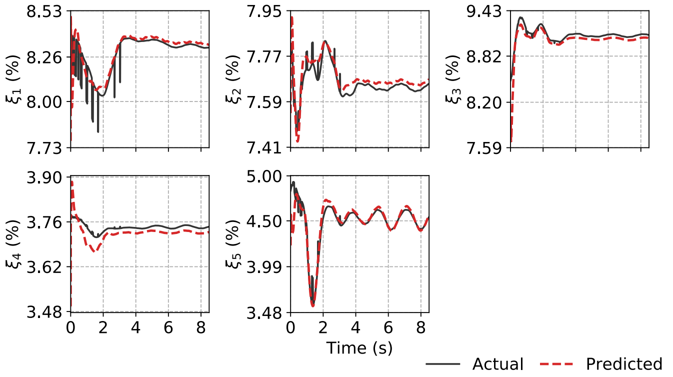


Fig. 11. Actual and SSLSTM-256 reconstructed values of ξ_j for an arbitrary test case in the IEEE 68-Bus System.

predicted at $T_{\max}$. This was similarly seen in the New England 39-Bus system during testing. However, due to the more strict threshold, the model failed to predict all cases within 20 time steps. We further inspect the misclassified cases and found that contingencies which go to instability 4 seconds or later in the simulation appear to be the majority of them.

The lower accuracy prompted an exploration into the conservativeness of GINN by examining the false positive rate (FPR) of the CNN models. The FPR describes the cases incorrectly predicted to be stable by the CNN but are actually unstable. Hence, a model which minimizes the FPR is characterized as more conservative as it is less likely to predict a stable case over an unstable one. FPR is calculated as

$$\mathrm{FPR} = \frac{FP}{FP + TN} \times 100\%. \tag{26}$$

where TN is the number of correctly predicted unstable cases in the testing set. For the 68-Bus system, the FPR is 2.33% meaning that proportion of the unstable cases are predicted incorrectly. In context of the entire testing set, 0.89% of the predictions are false positives while 1.47% of predictions are false negatives. This shows that the model is much less likely to predict unstable cases as stable hence making the model more conservative. The rates for other test systems are provided in Table V which shows that the model remains conservative when tested on different systems.

2) LSTM Evaluation: For the IEEE 68-Bus system, 3,020 training and 1,015 testing samples were extracted with valid $\boldsymbol{\lambda}$ values. As summarized in Tables IV and VI, SSLSTM-256 result in the lowest errors of reconstructed $\boldsymbol{\xi}$. Apparently, the larger input size benefitted from the increased hidden units of the LSTM as both the RMSE and MAAPE decrease with the larger models.

Figs. 9, 10 and 11 depict the real-time prediction of SSLSTM-256 on the voltage dynamics. This system exhibits five modes of oscillations where the eigenvalues λ_{1-3} and λ_{4-5} correspond to local and inter-area modes, respectively. It can be seen that the overall damping of these modes is below 10%. For this system, the LSTM predictions produced tight predictions that follow the dynamics of the simulated oscillatory modes. Over-estimations are visible in ξ_2 but only by very small amounts. While the simulation presented large spikes in λ_1 and less pronounced ones in λ_2 and λ_5 between 0 s and 3 s in Fig. 9, SSLSTM-256 is able to follow the stronger trend of the dynamics and produce accurate predictions.

Table VI shows that ξ_3 and ξ_5 are the more difficult modes to predict accurately. Further investigation ascertained that simulation values of λ_1, λ_3 and λ_5 contained rapid fluctuations similar to that seen in the New England 39-Bus system with the more prominent spikes appearing in the imaginary component of λ_3 and λ_5. Plateaus of λ_3 exacerbate this fit especially during the early period of transience after fault. However, considering that small signal stability is more important much later in the phenomena, the early predictions should be taken as very coarse responses and later values should be more heavily weighted to be accurate.

C. The IEEE 50-Machine, 145-Bus Test System

The sub-transient model is used to describe the dynamic performance of the synchronous generators G_{1-6} and G_{23}. These generators are equipped with fast static exciters (ST1A) and two lead-lag PSS. In addition, classical model is used to describe the dynamic performance of the rest of the generators [38]. Two types of faults are applied in order to generate the training and test cases for this system. The first type corresponds to a three-phase to ground fault which is applied at any bus and is naturally cleared. The second type represents three-phase to ground faults which are applied at any line with randomly selected locations and is cleared by tripping it from both ends. In addition, the fault clearing time is randomly selected within 0.06–0.5 seconds.

1) CNN Evaluation: From the generated cases, 4,996 stable cases and 2,504 unstable cases were present in the training set. Out the 2,504 unstable cases, 1,144 cases contained at least 1 cycle of data following t_c. Overall, the augmentation process generated 61,400 data points for training. In the testing set, 2,015 contingencies with at least 20 cycles of data after t_f are collected for evaluation.

At the best setting, $\Delta = 0.08$, GINN reliably predicts 1,902 (94.39%) contingencies with 98.89% accuracy in the first cycle before completing most cases in 20 cycles at a final accuracy of 98.46% leaving 12 cases to be predicted at $T_{\max}$. As for GCNN, 1,842 contingencies are reliably labelled at 99.34% accuracy in the first cycle but significantly depreciate over 20 cycles to 98.31% leaving 22 cases to be predicted at $T_{\max}$. Yet again, the resilience of GINN is shown in the empirical data. In comparison to other works, the accuracy of GINN has little desirable properties but it does prevail to its competitors [14], [35] in terms of ART, as shown in Table III. Training on a larger window may be of more importance to GINN as this point to compete further in accuracy.

The over-conservative nature of the model as shown in Table V could be due to the highly unbalanced number of unstable cases in the training set. As there are much fewer unstable cases, the learning is highly focused on predicting the unstable cases more correctly. This mirrors the findings in [21] where the CNN ensemble is able to be more secure despite the imbalanced training set and the GINN. However, as this work does not use a strict ensemble but rather a packaged ensemble, there are reduced training costs associated with this work.

2) LSTM Evaluation: For the IEEE 50-Machines system, 4,973 training and 1,648 testing stable samples were extracted with valid $\boldsymbol{\lambda}$ values. As summarized in Tables IV and VI, and similar to the IEEE 68-Bus system, SSLSTM-256 result in the lowest errors of reconstructed $\boldsymbol{\xi}$. It should be noted that the difference in RMSE and MAAPE is small and hence there are diminishing returns in training larger LSTM networks for this system. As LSTM units contain fully connected networks, the number of parameters increase polynomially with the number of layers for a set number of hidden units. If the number of hidden units is H, and there are K layers, then the number of parameters grows in $\mathcal{O}(H^K)$ and care must be taken to minimize model execution time.

Figs. 12, 13 and 14 depict the prediction of SSLSTM-256 on the voltage dynamics. This system exhibits six modes of inter-area oscillations as shown by the eigenvalues λ_{1-6}. The $\boldsymbol{\lambda}$ values for this system are rather smooth for most contingencies in the training and testing set as opposed to the New England 39-bus and IEEE 68-Bus system. In addition, SSLSTM-256 exhibits a high-frequency prediction pattern despite the smoother training data.

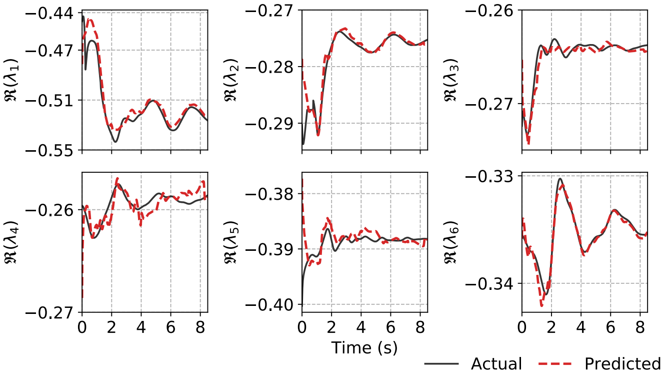


Fig. 12. Actual and SSLSTM-256 predictions of $\Re(\lambda_j)$ for an arbitrary test case in the IEEE 50-Machines System.

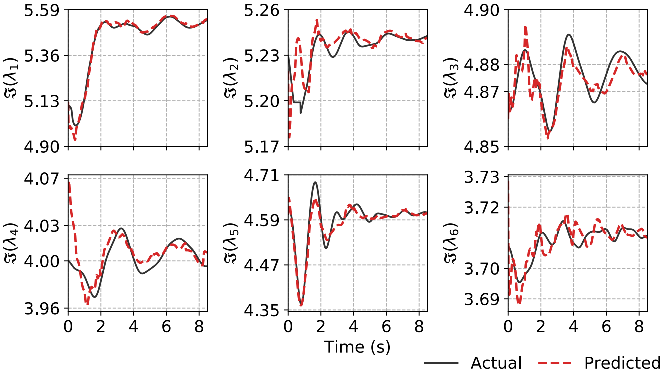


Fig. 13. Actual and SSLSTM-256 predictions of $\Im(\lambda_j)$ for an arbitrary test case in the IEEE 50-Machines System.

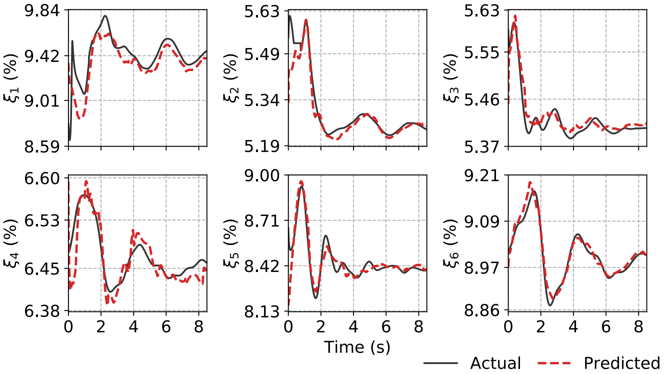


Fig. 14. Actual and SSLSTM-256 reconstructed values of ξ_j for an arbitrary test case in the IEEE 50-Machines System.

Occasionally, plateaus are visible in the real and imaginary components of λ_{1-2} as is evident in Fig. 13 for λ_2. More interestingly, the MAAPE values for ξ_{4-5} is much higher in Table VI. Through inspection, it was found that λ_{4-5} had a considerably higher variance compared to other λ values. Some contingencies appeared as almost straight lines with very small oscillations in λ_{4-5} translating to very small changes in values. Due to the minute differences, the NN is instead increasingly focused on

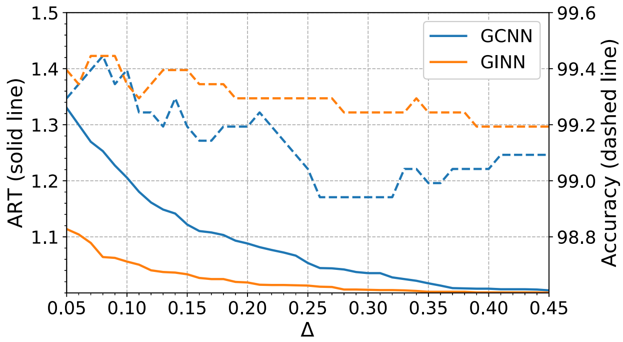


Fig. 15. GCNN and GINN sensitivity over different values of Δ.

fitting the signals with larger differences as the gradients are more extreme, thereby reasoning for the higher prediction error. A remedy to this would be longer training but risks the LSTM model overfitting on a small dataset.

D. Time Adaptivity in CNN

As the CNN execution can be parameterized by Δ, the sensitivity of the models have been investigated over different Δ settings. In Fig. 15, a clear separation between GCNN and GINN is visible in terms of ART for the New England 39-bus system. The IEEE 68-Bus and 50-machines system presented similar trends of ART and accuracy drop over increasing Δ. GINN is on average 6% faster than GCNN throughout all Δ values. As for accuracy, both models performed equally when Δ is close to 0.05 but GINN maintains its high accuracy compared to GCNN which instead fluctuated in accuracy. It can be inferred that having more strict threshold of confidence increases accuracy of models but causes the model to require more prediction time, thus responding more slowly. However, GINN is more resilient to the confidence threshold and we attribute this ability to the bottleneck convolution layer as a compression technique that maintains crucial kernels [32]. In relation to [14], the resilience of GINN allows for more freedom in tuning the Δ parameter to minimize ART as much as possible. Although our method displayed a lower accuracy score compared to that of [14], our model has a lower ART for both the larger and smaller systems. To make a fair comparison to the method in the literature, we have selected the training window to be 10 time steps. We expect our method to learn more complex dynamics and be able to predict with higher accuracy with a larger training window.

VI. Conclusion and Future Work

An online model is proposed in this paper for predicting rotor angle stability using voltage phasor measurements and deep learning techniques. The main objective of the presented models are to develop a unified approach for predicting both small-signal and transient stability when the system is subjected to a disturbance. This is achieved by first training a CNN model using a set of transient responses which are exhibited by the phasor voltages across the entire system and for wide-range of operating conditions. Once the system is found stable, the small-signal stability is predicted to provide information concerning the overall damping of low-frequency power system oscillations. The prediction model for small-signal stability is trained using an LSTM network and the phasor voltages which are provided by the PMUs. The presented results, tested on the New England 39-Bus, IEEE 68-Bus, 16-Machines, and IEEE 145-Bus, 50-Machines systems, show the capability of the proposed model to provide a comparatively accurate and much faster transient stability classification. Our unified method is also able to provide the small-signal stability response over a large time scale when the system is subjected to a disturbance.

The question of transferability of deep learning models comes to mind when discussing training efficiency across different power systems. Deep learning models learn high order representations of the transient signals to enable prediction and ideally should be able to transfer knowledge assimilated from one system to other larger systems. However, the process of conducting transfer learning between power systems has several issues. The major issue stems around the input data: the CNN's input channel size will change depending on the topology of the power system. Unlike images which consistently have 2 spatial dimensions and 3 color channels (4 if the transparency channel was included), the 1-dimensional signals in this work will vary in the number of channels for different systems, especially when the number of PMUs increase for larger systems. Furthermore, the ordering of the signals in a CNN's input layer is crucial to its execution as each kernel has learned the dynamics of a specific PMU and its corresponding data type making knowledge transfer non-trivial between power systems. Continual learning could be beneficial as a form of transfer learning but requires extensive effort to verify [44]. There is some evidence that learned convolution kernels can be fine-tuned to accelerate convergence during training but as the number of input kernels have changed, it is expected that other training issues will arise. This presents an avenue of research into system-agnostic NN architectures that could alleviate *ab initio* training of CNN and LSTM models in transient stability problems.

References

[1] P. Kundur, N. J. Balu, and M. G. Lauby, *Power System Stability and Control*, vol. 7, New York, NY, USA: McGraw-hill, 1994.

[2] Y. J. Isbeih, M. S. El Moursi, W. Xiao, and E. El-Saadany, "H mixed-sensitivity robust control design for damping low-frequency oscillations with dfig wind power generation," *IET Gener., Transmiss. Distribution*, vol. 13, 2019.

[3] Y. J. Isbeih, M. S. El Moursi, W. Xiao, and E. El-Saadany, "Generator-based threshold for transient stability assessment," *IET Smart Grid*, vol. 2, 2019.

[4] H. Nguyen-Duc, H. Cao-Duc, C. Nguyen-Dinh, and V. Nguyen-Xuan-Hoang, "Simulation of a power grid blackout event in vietnam," in *Proc. IEEE Power Energy Soc. General Meeting*, 2015, pp. 1–5.

[5] M. Pai, *Energy Function Analysis for Power System Stability*. Berlin, Germany: Springer Science & Business Media, 2012.

[6] A.-A. Fouad and V. Vittal, *Power System Transient Stability Analysis Using the Transient Energy Function Method*. London, U.K.: Pearson Education, 1991.

[7] E. Hossain, Z. Han, and H. V. Poor, *Smart Grid Communications and Networking*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[8] J. Rasmussen and P. Jorgensen, "Synchronized phasor measurements of a power system event in eastern denmark," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 278–284, Feb. 2006.

[9] F. Aboytes and R. Ramirez, "Transient stability assessment in longitudinal power systems using artificial neural networks," *IEEE Trans. Power Syst.*, vol. 11, no. 4, pp. 2003–2010, Nov. 1996.

[10] N. Amjady and M. Ehsan, "Transient stability assessment of power systems by a new estimating neural network," *Canadian J. Elect. Comput. Eng.*, vol. 22, no. 3, pp. 131–137, 1997.

[11] N. Amjady and S. F. Majedi, "Transient stability prediction by a hybrid intelligent system," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1275–1283, Aug. 2007.

[12] 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," *J Eng.*, vol. 2017, no. 13, pp. 1847–1850, 2017.

[13] Y. Xu *et al.*, "Assessing short-term voltage stability of electric power systems by a hierarchical intelligent system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1686–1696, Aug. 2016.

[14] J. James, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, "Intelligent time-adaptive transient stability assessment system," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018.

[15] L. Zheng, W. Hu, K. Hou, X. Xu, and G. Shao, "Real-time transient stability assessment based on deep recurrent neural network," in *Proc. IEEE Innovative Smart Grid Technol.-Asia*, 2017, pp. 1–5.

[16] A. B. Mosavi, A. Amiri, and H. Hosseini, "A learning framework for size and type independent transient stability prediction of power system using twin convolutional support vector machine," *IEEE Access*, vol. 6, pp. 69 937–69 947, 2018.

[17] Q. Zhu *et al.*, "A deep end-to-end model for transient stability assessment with pmu data," *IEEE Access*, vol. 6, pp. 65 474–65 487, 2018.

[18] C. B. Saner, M. Kesici, M. Mahdi, Y. Yaslan, and V. I. Genc, "Wide area measurement-based transient stability prediction using long short-term memory networks," in *Proc. 7th Int. Istanbul Smart Grids Cities Congr. Fair*, 2019, pp. 159–163.

[19] R. Yan, G. Geng, Q. Jiang, and Y. Li, "Fast transient stability batch assessment using cascaded convolutional neural networks," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2802–2813, Jul. 2019.

[20] F. Pan *et al.*, "Stacked-gru based power system transient stability assessment method," *Algorithms*, vol. 11, no. 8, pp. 121–130, 2018.

[21] Y. Zhou, Q. Guo, H. Sun, Z. Yu, J. Wu, and L. Hao, "A novel data-driven approach for transient stability prediction of power systems considering the operational variability," *Int. J. Elect. Power Energy Syst.*, vol. 107, pp. 379–394, 2019.

[22] L. Liu, Y. Li, Y. Cao, F. Liu, W. Wang, and J. Zuo, "Transient rotor angle stability prediction based on deep belief network and long short-term memory network," *IFAC-PapersOnLine*, vol. 52, no. 4, pp. 176–181, 2019.

[23] B. Tan, J. Yang, Y. Tang, S. Jiang, P. Xie, and W. Yuan, "A deep imbalanced learning framework for transient stability assessment of power system," *IEEE Access*, vol. 7, pp. 81 759–81 769, 2019.

[24] D. Shi *et al.*, "Study on quick judgement of small signal stability using cnn," *J Eng.*, vol. 2019, no. 16, pp. 826–829, 2019.

[25] S. Teeuwsen, I. Erlich, and M. El-Sharkawi, "Small-signal stability assessment for large power systems using computational intelligence," in *Proc. IEEE Power Eng. Soc. General Meeting*, 2005, pp. 2661–2668.

[26] S. Teeuwsen, I. Erlich, A. Fischer, and M. El-Sharkawi, "Assessment of the small signal stability of the european interconnected electric power system using neural networks," in *Proc. Large Eng. Syst. Conf. Power Eng. Conf. Proc. Theme: Powering Beyond*, 2001, pp. 158–161.

[27] C. M. Bishop *et al.*, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.

[32] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[35] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong, "Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system," *IET Gener., Transmiss. Distribution*, vol. 9, no. 3, pp. 296–305, 2015.

[36] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *Int. J. Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.

[37] A. K. Singh *et al.*, "Report on the 68-bus, 16-machine, 5-area system," IEEE PES Task Force on Benchmark Systems for Stability Controls. Ver, vol. 3, 2013. [Online]. Available: http://www.sel.eesc.usp.br/ieee/

[38] V. Vittal *et al.*, "Transient stability test systems for direct stability methods," *IEEE Trans. Power Syst.*, vol. 7, no. 1, pp. 37–43, Feb. 1992.

[39] F. Milano, "Psat, Matlab-based power system analysis toolbox," 2002. [Online]. Available: http://faraday1.ucd.ie/psat.html

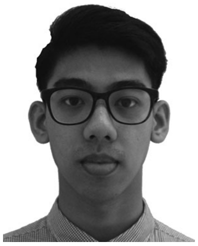
[40] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1474–1483, Aug. 2011.

[41] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley, and V. V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 947–956, May 2010.

[42] I. Kamwa, S. R. Samantaray, and G. Joós, "Development of rule-based classifiers for rapid stability assessment of wide-area post-disturbance records," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 258–270, Feb. 2009.

[43] L. Zhu, C. Lu, and Y. Sun, "Time series shapelet classification based online short-term voltage stability assessment," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1430–1439, Mar. 2016.

[44] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, 2019.



Syafiq Kamarul Azman received the B.Sc. degree in software engineering degree from the University of Nottingham, Nottingham, United Kingdom in 2014 and the M.Sc. degree in computing and information sciences degree from the Masdar Institute of Science and Technology, Abu Dhabi, UAE in 2017. Currently, he is a Research Engineer in the Department of Electrical and Computer Science in Khalifa University, Abu Dhabi, UAE and has been since 2018. His research interests include deep learning, applications of artificial intelligence, and bioinformatics.

Younes J. Isbeih (Member, IEEE) received the B.Sc degree from the German Jordanian University, Jordan in 2011 and the M.Sc and the P.hD. degrees from the Khalifa University of Science and Technology - Masdar Campus, UAE in 2015 and 2019, respectively.

He is currently working as a Postdoctoral Fellow at the Electrical Engineering and Computer Science department at the Khalifa University of Science and Technology. His research interests are diverse and include power system modeling and stability, robust control design in addition to renewable energy integration.

Mohamed Shawky El Moursi (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Mansoura University, Mansoura, Egypt, in 1997 and 2002, respectively, and the Ph.D. degree from the University of New Brunswick (UNB), Fredericton, NB, Canada, in 2005, all in electrical engineering.

He was a Research and Teaching Assistant in the Department of Electrical and Computer Engineering, UNB, from 2002 to 2005. He joined McGill University as a Postdoctoral Fellow with the Power Electronics Group. He joined Vestas Wind Systems, Arhus, Denmark, in the Technology $R\&amp;D$ with the Wind Power Plant Group. He was with TRANSCO, UAE, as a Senior Study and Planning Engineer. He is currently a Professor in the Electrical and Computer Engineering Department at Khalifa University of Science and Technology- Masdar Campus and seconded to a Professor Position in the Faculty of Engineering, Mansoura University, Mansoura, Egypt and currently on leave. He was a Visiting Professor at the Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. Dr. Shawky is currently an Editor of IEEE TRANSACTIONS ON POWER DELIVERY, IEEE TRANSACTIONS ON POWER SYSTEMS, Associate Editor of IEEE TRANSACTIONS ON POWER ELECTRONICS, Guest Editor of IEEE TRANSACTIONS ON ENERGY CONVERSION, Guest Editor-in-Chief for special section between TPWRD and TPWRS, Editor for IEEE POWER ENGINEERING LETTERS, Regional Editor for IET RENEWABLE POWER GENERATION and Associate Editor for IET POWER ELECTRONICS JOURNALS. His research interests include power system, power electronics, FACTS technologies, VSC-HVDC systems, microgrid operation and control, renewable energy systems (Wind and PV) integration and interconnections.

Khaled Elbassioni received the B.S. and M.S. degrees in computer science from Alexandria University, Egypt, and the Ph.D. degree in computer science from Rutgers University, USA. From 2006 to 2012, he was a Senior Researcher at Max-Planck Institute for Informatics, Saarbruecken, Germany. He is currently a Professor in the Electrical Engineering and Computer Science department at the Khalifa University of Science and Technology. His main research interests are in the design and analysis of efficient algorithms, with focus on discrete and continuous optimization, approximation algorithms, game theory and their applications in smart grid and power systems.