Data-Driven Fast Transient Stability Assessment Using (Fault-on + 2) Generator Trajectories

Lipeng Zhu, David. J. Hill, *Life Fellow, IEEE,* and Chao Lu, *Senior Member, IEEE*

***Abstract*—For transient stability assessment (TSA) in modern power systems, the assessment results should be issued as soon as possible to leave enough time for pre-emptive control. To this end, this paper develops a fast online TSA scheme using fault-on trajectories and their two adjacent data-points in pre- and post-fault stages, i.e., (fault-on + 2) trajectories. First, (fault-on + 2) trajectories of voltage magnitudes, rotor angles, and frequency deviations are acquired from multiple generators via PMUs. With such transient trajectories, a novel anti-noise transient bitmap based descriptor is then strategically designed to comprehensively describe the system-wide transients in bitmap forms. Finally, a convolutional neural network based TSA model is constructed by deep learning of transient bitmaps. Test results on the IEEE 39-bus system demonstrate the effectiveness, adaptability, and robustness of the proposed TSA scheme.**

***Index Terms*--Bitmaps, convolutional neural networks, phasor measurement units, transient stability, time series, trajectories.**

I. Introduction

With an increasing amount of power delivered via slowly evolved transmission networks, modern power systems are often operated near their stability limits, being greatly threatened by various stability issues. Amongst them, transient stability, which mainly relies on synchronous generators' ability to maintain synchronism against large disturbances [1], has attracted great attention in the power engineering field. Once transient instability occurs, it may have severe outcomes, leading to large-scale outages and even blackouts. To tackle the transient stability problem, many efforts have been made in the research community, e.g., time-domain simulation based analysis, transient energy function (TEF) and related methods. However, due to the salient nonlinear and high-dimensional features of transient stability, these conventional approaches have either extremely heavy computational burdens or limited applicability when applied to large-scale systems, except for highly approximated implementations, see [2] for instance. In practice, it is still challenging and urgent to develop reliable and efficient tools for online transient stability assessment (TSA).

In recent years, owing to the wide deployment of wide area measurement systems (WAMS) and phasor measurement units (PMU), the above TSA challenge has been eased greatly. In particular, with huge volumes of synchronous PMU data available, data-driven machine learning methods have shown to be powerful for online TSA [3]-[7]. These methods generally learn the underlying relationships between system states/responses and stability status from the data-analytics perspective, and then directly apply them to online TSA with a high efficiency. In this way, they are hopeful to overcome the dilemmas encountered by conventional TSA methods when applied to realistic large-scale systems. Most of these efforts take post-fault responsive data as inputs [3]-[5]. Since they need to collect post-fault PMU data before performing TSA, the issuance of TSA results would be delayed by a certain amount of time after fault clearance. In emergency situations, if the TSA results come out too late, it would be too difficult for corrective actions to save the system from instability. Although system stability can be predicted using security assessment models trained with steady-state measurements regarding presumed events [6], it may be short of adaptability to unknown contingencies.

To improve the early response of online TSA, recently a TSA method focusing on learning fault-on PMU data has been reported in [7]. Supposing fault occurrence/clearance can be identified by well-developed fault detection techniques [8], it tactfully takes steady-state and fault-on PMU data as raw inputs. Since post-fault data is not needed, it can predict the onset of potential instability once a fault is cleared, leaving sufficient time for pre-emptive control. However, as it requires to identify time-varying Thevenin equivalent parameters in real time, the TSA results may be sensitive to severe measurement noises.

Taking the above concerns into account, this paper develops a data-driven online TSA scheme that enables fast TSA almost immediately after fault clearance. In particular, the fault-on time series (TS) trajectories plus their two closest data-points in pre- and post-fault stages [(fault-on + 2) trajectories] acquired from multiple generators are taken as the raw inputs. An anti-noise transient bitmap based descriptor is devised to capture the spatial-temporal correlations within such transient trajectories and thus characterize the system-wide dynamics. Deep bitmap learning is then performed with the convolutional neural network (CNN) algorithm, which eventually derives a highly reliable and robust TSA model for online monitoring.

The rest of the paper is organized as follows. Section II introduces the transient bitmap based descriptor. In Section III, the CNN based TSA scheme is described in detail. Section IV tests the TSA scheme's overall performances on the IEEE 39-bus system. Finally, conclusions are summarized in Section V.

II. Transient Bitmap Based Descriptor

In a specific system, rotor angle swings are the direct and reliable indicator of transient stability. However, as generator voltages usually evolve much faster than generator rotor angles [3]-[5], they may be utilized to predict potential instability more rapidly. To achieve fast yet reliable online TSA, the fault-on trajectories of generator voltage magnitudes, rotor angles and frequency (rotor speed) deviations from nominal values, i.e., *V*,

This work was supported in part by the Research Grants Council of Hong Kong Special Administrative Region under the Theme-based Research Scheme through Project No. T23-701/14-N and in part by National Natural Science Foundation of China under Grant 51677097 and Grant U1766214.

L. Zhu and D. J. Hill are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: zhulpwhu@126.com).

C. Lu is with the Department of Electrical Engineering, Tsinghua University, Beijing, China (e-mail: luchao@tsinghua.edu.cn).

δ and Δf, are acquired by PMUs and taken as the main raw inputs. However, as PMUs cannot always precisely capture the time instants of fault occurrence and clearance, both the data-points on their left/right sides are sampled. This simple inclusion of the two adjacent data-points in pre- and post-fault stages makes the whole trajectories more informative for the TSA task. For brevity, the acquired TS trajectories are thus called (fault-on + 2) trajectories in this paper. With these trajectories, a transient bitmap based descriptor is developed to characterize the system's overall dynamics in the following.

A. Symbolic Characterization of Individual Trajectories

Before depicting a bitmap for system-wide transient description, individual (fault-on + 2) trajectories of $\{V, \delta, \Delta f\}$ are first compactly characterized via a technique called *Symbolic Aggregate approXimation* (SAX) [9]. It discretely encodes the numerical sampling points of the TS trajectories with a group of prescribed symbols. In this way, a TS is represented by a specific symbolic code or word, which preserves the TS's main evolution trends and simultaneously gains a strong resistance to noises. In particular, how a TS trajectory is characterized by a symbolic word is presented as follows.

Let a (fault-on + 2) trajectory be denoted as $\boldsymbol{TS} = \{x_1, x_2, \ldots, x_n\}$, where n is the total number of data-points in $\boldsymbol{TS}$. Assuming the cardinality of symbols is m, an alphabet for SAX is designated as $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. Given the cardinality m, the range of sampling values in $\boldsymbol{TS}$ can be split into m intervals with a specific breakpoint vector $\boldsymbol{\beta} = \{\beta_0, \beta_1, \ldots, \beta_m\}$. For the j-th interval ($1 \le j \le m$), all the data-points of $\boldsymbol{TS}$ falling into it are represented by the j-th symbol in $\boldsymbol{\alpha}$, i.e., α_j. Following this idea, $\boldsymbol{TS}$ is transformed into a symbolic word $\boldsymbol{C}$:

$$\boldsymbol{C} = \left\{ c_i \,\middle|\, c_i = \alpha_j, \text{if } \beta_{j-1} \le x_i < \beta_j \right\} (\text{for } 1 \le i \le n) \tag{1}$$

where c_i is the i-th symbol in $\boldsymbol{C}$, corresponding to x_i in $\boldsymbol{TS}$. Clearly, the parameters m and $\boldsymbol{\beta}$ have a crucial impact on the results of symbolic characterization. According to [9], [10], setting the cardinality to m = 4~8 generally results in satisfactory performances on diverse TS datasets. Considering the computational complexity of subsequent bitmap learning, the cardinality is chosen as m = 4. As for $\boldsymbol{\beta}$, its elements are statistically determined based on the mean value and standard deviation of all the TS data-points of the same quantity. Taking the voltage quantity for instance, by statistically collecting the voltage trajectories of each generator from all the learning cases, their mean value and standard deviation are obtained and denoted as μ_V and σ_V, respectively. $\boldsymbol{\beta}$ is determined in the sense that the area under the probability density curve of the ideal Gaussian distribution $N(\mu_V, \sigma_V^2)$ is separated into m equal parts by the (m+1) breakpoints in $\boldsymbol{\beta}$ [9], [11]. Specifically, $\boldsymbol{\beta}$ is formulated as

$$\boldsymbol{\beta} = \left\{ \beta_i \,\middle|\, \int_{\beta_{i-1}}^{\beta_i} f(u)du = \frac{1}{m}, \text{for } 1 \le i \le m \right\} \tag{2}$$

where $f(u)$ is the probability density function of the hypothetical Gaussian distribution $N(\mu_V, \sigma_V^2)$. To cover the whole range of sampling values, β_0 and β_m are set to $-\infty$ and $+\infty$, respectively. Note that although the statistical voltage values may not necessarily comply with a standard Gaussian distribution, practical tests show that any possible violation has a negligible effect on the SAX-based characterization [9], [11]. Similarly, the breakpoints for δ and Δf can also be calculated with their mean values and standard deviations. Taking the learning cases in Section IV-A for instance, based on the statistical analysis of all the $\{V, \delta, \Delta f\}$ trajectories, their breakpoints are summarized in Table I.

TABLE I BREAKPOINT VECTORS FOR $\{V, \delta, \Delta f\}$ SAMPLING VALUES

Quantity	Mean value	Standard deviation	Breakpoint vector
V	0.825 pu	0.180 pu	$\{-\infty, 0.704, 0.825, 0.946, +\infty\}$
δ	57.65 deg	36.60 deg	$\{-\infty, 32.96, 57.65, 82.34, +\infty\}$
Δf	0.262 Hz	0.722 Hz	$\{-\infty, -0.225, 0.262, 0.748, +\infty\}$

Given the above parameters, how a (fault-on + 2) voltage trajectory is represented by a symbolic word is exemplified in Fig. 1. As can be seen, the voltage TS is discretely represented as $\boldsymbol{C}$ = {*daaaaaaaaaaaab*}.

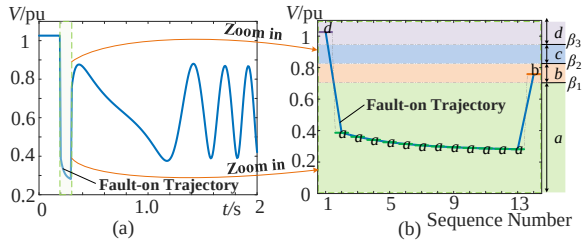


Fig. 1 Illustrative SAX based trajectory characterization. (a) transient voltage profile of a generator; (b) SAX representation of the (fault-on + 2) trajectory.

B. Pictorial Representation of System-Wide Transients

With all the $\{V, \delta, \Delta f\}$ trajectories transformed into symbolic words, they are integrated as a compact wordbook, from which the main transient characteristics of the system will be eventually described by a discerning bitmap. First, inspired by the idea that critical transient evolution trends may exist in fractional sequences instead of individual snapshots [12], a series of affixes are extracted from the full-length words to gain further insights into system dynamics. Supposing the length of affixes is L, (n–L+1) affixes can be extracted from $\boldsymbol{C}$, which yields

$$\boldsymbol{A} = \left\{ \boldsymbol{a}_i \,\middle|\, \boldsymbol{a}_i = \{c_i, c_{i+1}, ..., c_{i+L-1}\}, \text{for } 1 \le i \le n - L + 1 \right\} \tag{3}$$

where $\boldsymbol{A}$ is the collection of all the L-length affixes extracted from $\boldsymbol{C}$. In general, an affix with a larger length L may help to sufficiently describe the evolution trends of transient trajectories, especially for relatively slow rotor angle swings. However, with the increase of L, the cardinality of affixes, i.e., m^L, grows significantly, leading to a larger transient bitmap. As the bitmap size has a crucial impact on the computational burden of CNN learning in the sequel, L should not be too large. In fact, empirical tests show that setting L to 2~3 for $\{V, \delta, \Delta f\}$ would result in a satisfactory trade-off between the CNN classification performance and computational efficiency. Hence, L is uniformly set to L = 2 for all the trajectories of $\{V, \delta, \Delta f\}$.

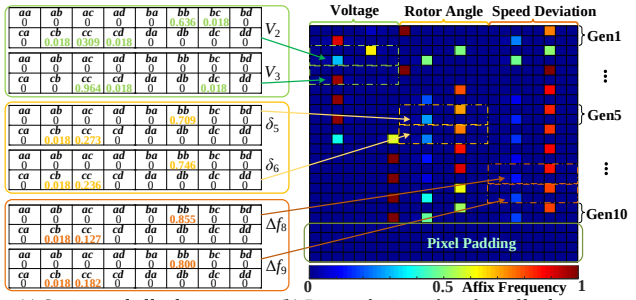
Given a two-symbol affix $\{\alpha_i, \alpha_j\}$ ($1 \le i, j \le m$), its normalized frequency of occurrence in $\boldsymbol{C}$ is calculated as

$$f_{\alpha_i \alpha_j} = \frac{1}{n-2+1} \sum_{k=1}^{n-2+1} \chi\left(\boldsymbol{a}_k, \alpha_i \alpha_j\right) \tag{4}$$

$$\chi\left(\boldsymbol{a}_k, \alpha_i \alpha_j\right) = \begin{cases} 1 & \text{for } \boldsymbol{a}_k = \{\alpha_i, \alpha_j\} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\chi(*)$ is a characteristic function judging whether the k-th affix in $\boldsymbol{A}$ is equal to $\{\alpha_i, \alpha_j\}$. The normalization in (4) enables the processing of transient trajectories with different lengths, which are induced by different fault durations under different scenarios. By computing the occurrence frequencies of all the possible affixes in $\boldsymbol{C}$, an $m*m$ frequency matrix characterizing

the statistical features of local evolution trends is constructed. As illustrated in Fig. 2(a) (for convenience, each frequency matrix is reshaped as a 2*8 matrix), frequency matrices of {V, δ, Δf} variables collected from different generators are lined up together, being ready for bitmap depiction.



(a) Statistics of affix frequencies (b) Bitmap depiction based on affix frequencies

Fig. 2 Illustration of bitmap based representation.

In essence, a bitmap is rendered by mapping affix frequency values to a specific colormap [10]. As shown in Fig. 2(b), each 2∗8 frequency matrix is represented by a 2∗8 pixel matrix. As for the construction of the whole bitmap, the pixel blocks of all the generators with the same type of quantities are grouped and aligned from top to bottom (in the same columns). Given a system with 10 generators, the pixel matrices of all the 10 generator voltages are aligned in the 1st~8th columns, occupying a 20∗8 block. The pixel matrices of the other two types of quantities {δ, Δf } are depicted in the same way. All of these matrices form a 20*24 pixel block. As it is generally more convenient for the classical CNN algorithm to handle a squared picture, the 20∗24 pixel block is further padded with a 4∗20 blank patch (each pixel with the 0 value). In this way, a squared 24∗24 bitmap is eventually depicted to characterize the system-wide dynamics for a (fault-on + 2) transient process.

Remark: Pixel matrices of topologically adjacent generators can be positioned nearby in the bitmap, so as to take into account possible spatial correlations of transient behaviors between generators (similar to the bus re-ordering strategy in [13]).

III. Proposed TSA Scheme

With the above transient bitmap based descriptor, all the transient cases are represented by transient bitmaps. By learning features from such transient bitmaps, a fast online TSA scheme can be constructed, as shown in Fig. 3. The whole scheme consists of three stages, which are described in the following.

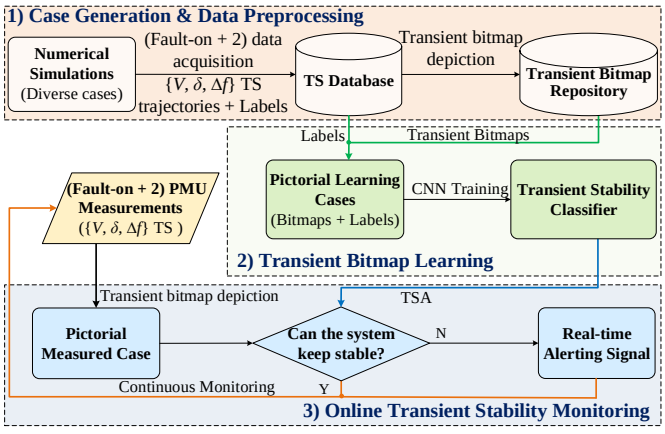


Fig. 3 Proposed TSA scheme.

A. Case Generation and Data Preprocessing

Given a specific system, considering a variety of possible operating conditions and contingencies, numerous transient stability cases are generated via numerical simulations. The (fault-on + 2) TS trajectories of the {V, δ, Δf} variables at all the generator buses are acquired from each case. Meanwhile, the transient stability status of each case, denoted as y, is labeled via a rotor angle based index η [14]:

$$y=\begin{cases}1 & \text{if } \eta>0 \quad \text{(stable)}\\ 0 & \text{otherwise (unstable)}\end{cases}, \text{ for } \eta=\frac{360^{\circ}-\left|\Delta\delta\right|_{\max}}{360^{\circ}+\left|\Delta\delta\right|_{\max}} \tag{6}$$

where $|\Delta\delta|_{\max}$ is the maximum absolute value of the rotor angle deviation between any two generators during the 0~10 s transient simulation period. The TS trajectories and the labels of all the cases are gathered to form a labeled TS database. With the help of the transient bitmap based descriptor, all the transient cases are further characterized by a series of bitmaps, which will be fed into the 2nd stage for classification learning.

B. Transient Bitmap Learning

Taking the bitmaps and labels of all the cases as the inputs and outputs, respectively, the proposed TSA scheme employs a CNN classifier to learn the hidden relationship between the transient bitmaps and system stability status. As this paper mainly focuses on presenting the application of CNN to the TSA task, for more details about CNN, interested readers may refer to [15].

Although a multitude of complex and powerful CNN models have been proposed in recent years, one of the simplest CNN structures is adopted here to minimize the offline computational burden of the TSA task. As shown in Fig. 4, the adopted CNN is a variation of the classical LeNet-5 model [16]. It has a bitmap based input layer, two convolution layers, two max-pooling layers, two fully connected layers and an output layer. All the convolutional computations are performed with a kernel size of is 5∗5 and a stride size of 1. All the max-pooling manipulations are implemented with a subsampling size of 2∗2 and a stride size of 2. The output of each convolution layer is activated by a rectified linear unit (ReLU) [15]:

$$f(x)=x^{+}=\max(0,x) \tag{7}$$

For the eventual output layer, the activation and loss functions are designed as

$$\begin{cases}\hat{y}=\dfrac{1}{\left(1+e^{-x}\right)}\\ \ell(y,\hat{y})=\left(y-\hat{y}\right)^{2}\end{cases} \tag{8}$$

where $\hat{y}$ and y are the predicted and actual outputs of a certain case, respectively. By feeding all the bitmaps to the CNN model, iterative learning is performed, which eventually builds a binary transient stability classifier for online TSA. With the predicted value $\hat{y}$, the binary stability status y_{pre} is predicted as

$$y_{\text{pre}}=\begin{cases}1 & \text{if } \hat{y}\geq\varepsilon \quad \text{(stable)}\\ 0 & \text{otherwise (unstable)}\end{cases} \tag{9}$$

where ε ($0<\varepsilon<1$) is the decision threshold acting as the boundary to separate stable cases from unstable ones. It is generally set to ε = 0.5 for unbiased online TSA.

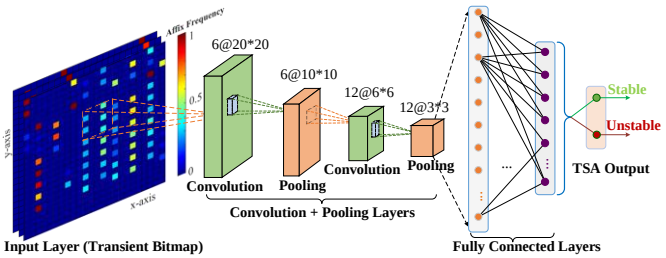


Fig. 4 CNN based transient bitmap learning.

C. Online Transient Stability Monitoring

During online monitoring, when the system encounters a transient fault, PMUs located at generator buses acquire the (fault-on + 2) TS data for all the $\{V, \delta, \Delta f\}$ quantities. With these measured TS trajectories, a pictorial measured case is generated by bitmap depiction and then fed to the transient stability classifier to quickly predict whether the system can keep stable. If not, alerting signals will be issued in real time to warn that corrective countermeasures be taken as soon as possible. Otherwise, continuous online monitoring will be performed, and TSA will be executed again for the next contingency.

IV. Simulation Results

A. Simulation Setting and Assumption

The whole TSA scheme was tested on the IEEE 39-bus system, as depicted in Fig. 5. The 10 generator buses 30~39 were assumed to be deployed with PMUs for TSA. Considering spatial correlations, the generator buses were sorted as {30, 37, 39, 31, 32, 33, 34, 35, 36, 38}. The PMUs' sampling rate for (fault-on + 2) trajectory acquisition was set to 120 Hz. Numerous conditions and contingencies were simulated for transient case generation. To cover a wide variety of operating conditions, different levels of load demands and typical topological variations were considered. Three-phase short-circuit faults with time durations 0.1~0.5 s were separately imposed on different transmission lines. Based on these settings, 9000 cases were generated by numerical simulations. All the cases were randomly assigned to two groups, with 7200 ones constituting a training set, while the remaining 1800 ones forming a testing set.

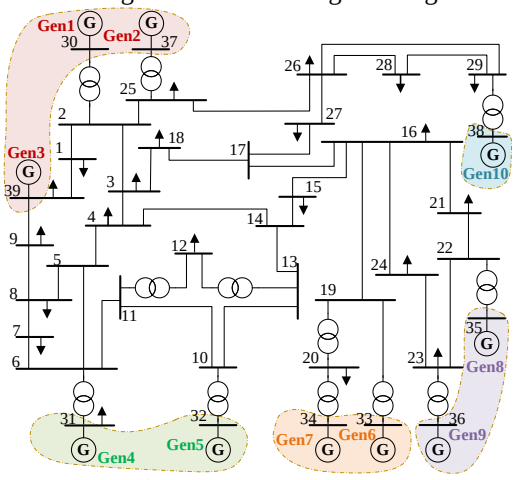
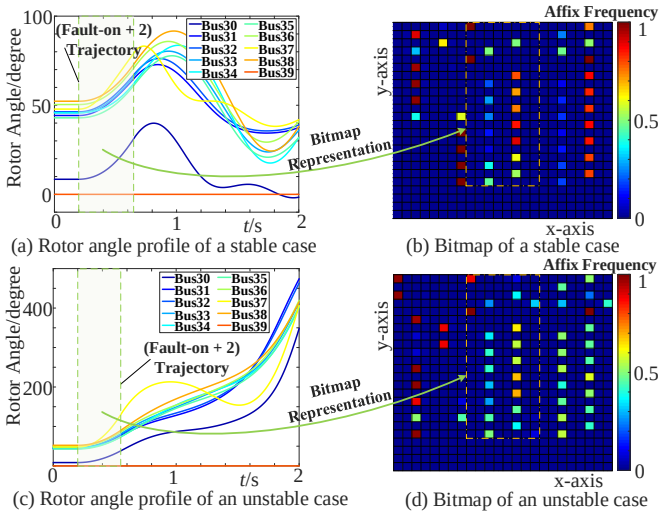


Fig. 5 One-line diagram of the IEEE 39-bus system.

B. Illustration of Transient Bitmap Description

Without loss of generality, two transient cases were randomly chosen from the training set to illustrate the effect of transient bitmap description, as shown in Fig. 6. Clearly, the two bitmaps exhibit significant differences. In particular, the bitmap of the stable case has more red pixels, while the bitmap of the unstable case has more diverse pixels. This is because the evolution of system trajectories in the unstable case is more drastic, resulting in multiple abundant TS segments and affixes for bitmap depiction. For the stable case, however, as its trajectories evolve moderately, the TS segments are more likely to concentrate on a certain affix. Such discriminative bitmaps would greatly contribute to the high performance of the TSA scheme, which will be shown below.



(a) Rotor angle profile of a stable case (b) Bitmap of a stable case

(c) Rotor angle profile of an unstable case (d) Bitmap of an unstable case

Fig. 6 Illustration of bitmap based transient case representation. (a)~(b): stable case; (c)~(d): unstable case.

C. Performance on Online TSA

1) CNN Training and Classification: With the 7200 training cases described by bitmaps, a CNN model was trained for online TSA. In the meantime, several conventional classifiers, including support vector machine (SVM), decision tree (DT), and random forest (RF) [17], were trained on the same case base for comparative study. As these classifiers cannot directly handle 2-D images, all the 24∗24 bitmaps were flattened as 480-dimentional vectors (with the padded pixels removed). 10-fold cross validation was employed to evaluate the classifiers' performances on the training set. The 1800 testing cases not present in the learning stage were fed to all the classifiers to mimic online TSA. All their performances are summarized in Table II.

TABLE II Performance Comparison of Different Classifiers

Accuracy/%	CNN	SVM	DT	RF
Cross validation	99.40	95.19	95.56	96.04
Online test	98.11	94.83	94.17	95.67

Obviously, the CNN model outperforms the other 4 learners, with its accuracy being above 98%. In fact, its high performance mainly benefits from transient bitmap description, which not only statistically characterizes the temporal evolution trends of transient trajectories, but also considers possible spatial correlations. For the other 4 classifiers, only temporal characteristics are involved in the flat vector. Besides, as their learning structures are much simpler than CNN, it is difficult for them to fully deduce the underlying relationship between short TS trajectories (0.1~0.5 s) and the system stability status. Hence, the CNN model would be preferred for online TSA in practice.

2) Computational Efficiency: To evaluate the overall computational efficiency, the computation times cost by each stage

of the TSA scheme were recorded in Table III. As can be seen, most of the computation times are consumed by the CNN training procedure (with 2000 epochs). It is due to the deep learning structure of the CNN model which involves hundreds of thousands of parameters to be trained. However, this does not impact online TSA efficiency. Actually, online TSA results can be issued almost right after fault clearance, with a negligible online processing time. Note that all the data processing and calculations were conducted in MATLAB on a PC configured with an Intel Core i7 CPU (3.60 GHz∗8) and 8.0 GB RAM. If more advanced deep learning frameworks such as Google TensorFlow and GPU based computing platforms are deployed, the offline learning procedure can be significantly speeded up.

TABLE III STATISTICS OF COMPUTATIONAL TIME

Offline bitmap generation time (7200 cases)	Offline CNN training time (7200 cases)	Online bitmap processing time (on average)
8.79 s	7741.52 s	1.2 ms

3) Adaptability to Unexpected Conditions: As the operating conditions in practice generally vary from time to time, two brand-new scenarios not included in the initial database in Section IV-A were simulated to test the scheme's adaptability to unforeseen conditions: 1) Three transmission lines (bus 25-bus 26, bus 26-bus 28, bus 28-bus 29) were respectively disconnected to simulate unscheduled line tripping. 2) Load demands at load buses were separately randomized from 80% to 120% of their base levels, so as to mimic stochastic and unknown load patterns. Similar to Section IV-A, by setting different fault locations and fault clearing times, 1800 and 2700 new transient cases were generated for the two new scenarios via numerical simulations. TSA was implemented by directly applying the CNN model trained previously. As shown in Table IV, with such totally unknown scenarios, the scheme still achieves an excellent TSA accuracy (> 97.3%). This implies that the TSA scheme adapts well to unforeseen conditions in the system.

TABLE IV TSA PERFORMANCES IN UNEXPECTED SCENARIOS

Scenario	No. of cases	Accuracy/%	Misdetection/%	False Alarm/%
1	1800	97.39	1.22	1.39
2	2700	97.48	1.11	1.41

Remark: Misdetection rate is the ratio of falsely dismissed unstable cases, while false alarm rate stands for the percentage of stable cases that are falsely alarmed [18].

4) Robustness to PMU Measurement Errors: Considering practical measurement contexts, the TSA scheme's performance was further tested in the presence of PMU measurement errors. Specifically, PMU measurement errors were simulated by superimposing Gaussian noises following the distribution $N(0, \sigma^2)$ on transient $\{V, \delta, \Delta f\}$ trajectories of the 1800 testing cases in Section IV-A. Different levels of measurement errors were considered by setting σ to $\sigma = 0$, 1%, 2% and 3%, respectively. With the CNN model in Table II, the classification performances at different error levels are summarized in Table V.

TABLE V TSA PERFORMANCES IN NOISY CONTEXTS (1800 TESTING CASES)

Noise level	Accuracy/%	Misdetection/%	False Alarm/%
$\sigma = 0$	98.11	0.89	1.00
$\sigma = 1\%$	98.06	0.83	1.11
$\sigma = 2\%$	97.72	1.00	1.28
$\sigma = 3\%$	97.06	1.33	1.61

Obviously, the TSA scheme remains resistant to PMU measurement noises, with a limited reduction of accuracy. Even with the error level as high as 3%, its TSA accuracy stays above 97%. Such a desirable anti-noise merit results from the SAX based characterization of TS trajectories, which is able to capture critical TS evolution trends by strategic discretization.

V. CONCLUSION

This paper develops a fast online TSA scheme by deeply learning transient trajectories from multiple generators. Taking (fault-on + 2) TS data of $\{V, \delta, \Delta f\}$ quantities as the raw inputs, it tactfully designs a transient bitmap based descriptor to characterize system-wide transient behaviors. With each case described by a 2-D bitmap, it leverages the CNN algorithm to train a transient stability classifier for online TSA. Test results on the IEEE 39-bus system verify that the proposed scheme achieves superior performances, with online TSA results reliably issued almost immediately after fault clearance. Preliminary tests of its adaptability to unforeseen conditions and robustness to measurement errors have further demonstrated its applicability. Yet more systematic studies on how the scheme performs in more imperfect practical contexts and how to cope with them remain to be carried out in relevant future work. In particular, special attention may be paid to how the malfunctions of the PMUs and WAMS will affect the scheme's online TSA performances.

VI. REFERENCES

[1] P. Kundur, *Power System Stability and Control*. NY: McGraw-Hill, 1994.

[2] Y. Xue, T. Huang, and F. Xue, "Effective and robust case screening for transient stability assessment," in *Proc. IREP Symposium-Bulk Power System Dynamics and Control – IX (IREP)*, pp. 1-8, 2013.

[3] A. D. Rajapakse, F. Gomez, K. Nanayakkara, *et al.*, "Rotor angle instability prediction using post-disturbance voltage trajectory." *IEEE Trans. Power Syst.*, vol. 25, no.2, pp.947-956, 2010.

[4] F. R. Gomez, *et al.*, "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, 2011.

[5] R. Zhang, Y. Xu, Z. Y. Dong, *et al.*, "Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system," *IET Gener. Transmiss. Distrib.*, vol. 19, no. 5, pp. 296–305, 2015.

[6] M. He, J. Zhang, V. Vittal, "Robust Online Dynamic Security Assessment Using Adaptive Ensemble Decision-Tree Learning", *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4089-4098, 2013.

[7] S. M. Mazhari, N. Safari, *et al.*, "A hybrid fault cluster and Thévenin equivalent based framework for rotor angle stability prediction," *IEEE Trans. Power Systems*, vol. 33, no. 5, pp. 5594–5603, 2018.

[8] Y. Ge, A.J. Flueck, *et al.*, "Power system real-time event detection and associated data archival reduction based on synchrophasors," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2088–2097, 2015.

[9] J. Lin, E. Keogh, L. Wei, *et al.*, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[10] N. Kumar, V.N. Lolla, E.J. Keogh, *et al.*, "Time-Series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases," in *Proc. SIAM Int'l Conf. Data Mining (SDM)*, 2005.

[11] L. Zhu, C. Lu, Y. Liu, *et al.*, "Wordbook-based light-duty time series learning machine for short-term voltage stability assessment," *IET Gener. Transmiss. Distrib.*, vol. 11, no. 2, pp. 4492-4499, 2017.

[12] L. Zhu, C. Lu, 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.

[13] S. Li, and V. Ajjarapu. "Real-time monitoring of long-term voltage stability via convolutional neural network," in *IEEE Power & Energy Society General Meeting*, pp. 1-5, 2017.

[14] *"User Manual-Transient Security Assessment Tool,"* Powertech, 2009.

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

[16] Y. LeCun, *et al.*, "Gradient-Based Learning Applied to Document Recognition", *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[17] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[18] L. Zhu, C. Lu, Z. Y. Dong, *et al.*, "Imbalance learning machine-based power system short-term voltage stability assessment," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2533–2543, 2017.