Real-Time Transient Stability Early Warning System using Graph Attention Networks

Arvid Rolander
Anton Ter Vehn
Lars Nordström
Electric Power and Energy Systems
KTH - Royal Institute of Technology
Stockholm, Sweden
{arvidro, antontv, larsno}@kth.se

Robert Eriksson
System Development
Svenska Kraftnät
Sundbyberg, Sweden
robert.eriksson@svk.se

***Abstract*—In this paper, a classifier based early warning system is designed, trained and tested based on time-series of Phasor Measurement Unit (PMU) measurements at all buses in a power system. The classifier is based on a novel combination of Graph Attention Networks and Long Short-Term memories, and is trained to label power system data in the form of captured windows of PMU measurements. These labels are then used to provide early warning for transient instability. The classifier is trained and tested data from simulations of the Nordic44 test system, and includes extensive topological variations under two different load levels. It is found that accurate early warnings can be provided, but the quality of prediction is highly dependent on specific power system characteristics, such as how quickly the power system responds to transient disturbances.**

***Index Terms*—Graph Attention Networks, phasor measurements, smart grid, transient stability, WAMS**

I. Introduction

The share of inverter based resources (IBRs), such as renewables like wind and solar energy, in the European grid has increased rapidly in recent years [1], and this development seems set to continue and even accelerate in the future [2]. One consequence of increasing amounts of IBRs in the system is a change of the fundamental power system dynamics from an electro-mechanical into a electromagnetic system, with faster dynamics. This, together with a general increase in complexity of the power systems due to increased market coupling and larger variability in generation, is creating challenges for traditional ways of operating the power system. Operators in control rooms need improved decision support tools and automation that can provide early warning and propose remedial actions. One such area which will require improved support is Transient Stability Assessment (TSA)

Several methods, including the Single Machine Equivalent (SIME) method [3], time domain simulations [4], and the Transient Energy Function (TEF) method [5] have been used for transient stability assessment (TSA). These methods share a reliance on high quality input data, and potentially heavy calculations which can become intractable with increasing system complexity. In addition, it is as of yet unclear how to model a system with a very high share of renewables [6]. A potential solution to these challenges is to make use of data driven methods, which enable placing the computational burden offline before deployment and are not reliant on accurate system modelling. Several such methods have been proposed, among them Support Vector Machine (SVM) based classifiers [7], Convolutional Neural Networks (CNNs) [8], Vision Transformers (ViTs) [9], and Graph Convolutional Networks (GCNs) [10]. In [8], fixed length sequences of generator bus voltage phasors are treated as a 3-channel RGB-image or heatmaps, where the channels correspond to the measured voltage magnitude, angle and frequency. The method in [9] differs from that of [8] in that it only considers states as stable or unstable, whereas [8] also identifies the start, duration and clearing of faults. In [10], both single and two component failures are considered in data generation, and the network is trained to distinguish between oscillatory and aperiodic instability. Both [11] and [12] make use of GAT networks, with the attention mechanism of [13], for transient stability prediction, but neither make use of time series of measurements, instead using individual snapshots of the system state.

Another promising data-driven approach to TSA is based on estimating Lyapunov Exponents (LE) from measurements of the post-fault system [14]. Ref. [15] proposes a method where the largest Lyapunov Exponent (LLE) is estimated based on time-series starting shortly after short clearing and covering up to 5s. This method was able to accurately predict system stability status under all tested scenarios. One drawback of this method, that is avoided in this work, is that the exact moment of fault clearing has to be known, which can potentially delay the stability assessment if used in real time. Additionally, the use of relatively long time-series could be an issue in the study of transient stability, as this manifests at very short time scales.

An interesting application of using LLEs for real-time TSA can be found in [16], in which PMU-data is used to estimate

This work was supported by the project System Integrity Protection Schemes for added transfer capacity, funded by Energimyndigheten under project number P2022-00679. Submitted to the 23rd Power Systems Computation Conference (PSCC 2024).

LEs. This method is not evaluated not extensively, but it is found that for the examined cases accurate TSA can be performed within 2.5s of fault clearing. Similarly to [15], the method is highly accurate, but suffers from needing knowledge of the moment of fault clearing for real-time application.

In summary, a large number of data-driven methods for transient stability detection are being investigated, in general showing good promise for the use of data-driven approaches.

A. Scope of the paper

This paper addresses the challenge of transient stability assessment in two stages. Firstly, a classifying algorithm, which provides labeling of power system data in the form of captured windows of PMU-measurements during transient conditions, is developed. Secondly, these labels are used to analyze a stream of measurements and to identify the most probable label for the data in the real-time stream, thereby providing an early warning for angular instability.

The classifier is based on Graph Attention Networks (GATs) and Long Short-Term Memories (LSTMs) [17], and uses node attributes, in the form of frequency, voltage magnitude and voltage angle measurements as well as the topology of the power system. The structure of the neural network is novel compared to earlier works, and uses different components such as the improved GAT module [18]. The data used for training and testing is gathered from simulations of the Nordic44 test system [19] in DigSILENT Powerfactory, and consists of synthetic PMU data of voltage phasor and frequency measurements with 50 Hz resolution taken at every bus in the system.

The experimental setup is similar to that of [7]–[10]. An improvement in this work is that topological variations of the base system are considered in the training data. Additionally, this work differs in that a generator is considered out of step when its rotor angle has deviated more than 360° from it's starting position.

II. GAT-based Classifier

A. Graph Neural Networks

In the wake of the successes of convolutional neural networks (CNN) in various grid-structured domains such as image recognition and sequence encoding [20], attempts have been made to generalize convolutions to general, graph structured domains [21]–[23]. These are encountered in various applications, such as citation networks, molecular interactions and not least power systems. Generalization is not without challenges, as nodes in a graph can have varying degrees of connectivity, and, moreover, one might wish to use the same network for several completely different graphs.

Early attempts to use neural networks on arbitrarily structured graph data includes using recurrent neural networks (RNN) [24], [25]. More recent solutions can be divided into spectral and spatial approaches [23]. Spectral approaches focus on applying Fourier domain filters on the eigendecomposition of the Graph Laplacian. The eigenbasis is determined by the graph structure, so a model fitted on a specific structure cannot be directly applied to a new problem with a different structure. This is limiting in power system applications, as the topology of the system can and does change. Non-spectral approaches define convolutions directly on the graph through a message-passing framework, where messages are passed between nodes and information is aggregated to update node features. One challenge in this approach is to define a convolutional operator that works for different sizes of node-neighborhoods, while maintaining the weight-sharing property of CNNs.

GATs were introduced in [13], and leverage an attention mechanism similar to that used in the transformer models [26] which have garnered massive success in Natural Language Processing (NLP), being used for machine translation and more recently formed the basis for Large Language Models (LLM). The idea behind the attention mechanism is that for each node in the graph, the relative importance of the features of each of its neighboring nodes is considered in the update. This creates a more expressive operator than a standard graph convolution. The graph attention implementation of [13] suffered some issues which were resolved by [18], and this work therefore uses the implementation found in [18]. Given a node i, the node feature vector is denoted by $h_i \in \mathbb{R}^F$ and the one hop neighborhood of the node by $\mathcal{N}_i$. The updated feature vector $h_i' \in \mathbb{R}^{F'}$ after the graph attention layer is given by

$$h_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}h_j\right), \tag{1}$$

where σ is some non-linear function, $\alpha_{ij} \in \mathbb{R}$ is the attention score for node j with respect to node i, $\mathbf{W} \in \mathbb{R}^{F'} \times \mathbb{R}^F$ is a learnable weight matrix and h_j is the feature vector of node j. The attention score α_{ij} is calculated by

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}. \tag{2}$$

Further, the unnormalized attention score e_{ij} is given by

$$e_{ij} = \mathbf{a}^T \mathrm{LeakyReLU}([\mathbf{W}h_i \Big\| \mathbf{W}h_j]), \tag{3}$$

where $\mathbf{a} \in \mathbb{R}^{2F'}$ is a learned vector,$\|$ denotes vector concatenation and $\mathrm{LeakyReLU}(x) = \max(cx, x)$ for some $c \in [0, 1)$. In the case where K attention heads are used, similar to [26], updates from different heads are concatenated giving a final output

$$h_i' = \Big\|_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^K \mathbf{W}^k h_j\right). \tag{4}$$

B. Neural Network design

The problem of transient instability prediction can be framed as predicting the trajectory of the power system, followed by determining if the predicted trajectory will lead to an unstable state. The GAT-network can be used to extract feature vectors from graph-structured data, if the GAT-layers are followed by some sort of global pooling or readout operation.

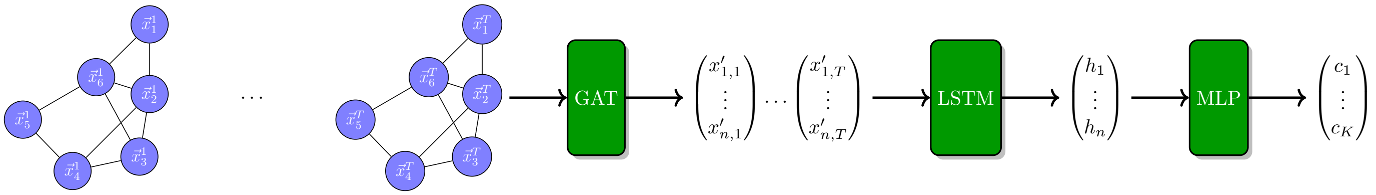
$\vec{x}_1^1$
$\vec{x}_2^1$
$\vec{x}_3^1$
$\vec{x}_4^1$
$\vec{x}_5^1$
$\vec{x}_6^1$
$\cdots$
$\vec{x}_1^T$
$\vec{x}_2^T$
$\vec{x}_3^T$
$\vec{x}_4^T$
$\vec{x}_5^T$
$\vec{x}_6^T$
GAT
$\begin{pmatrix} x'_{1,1} \\ \vdots \\ x'_{n,1} \end{pmatrix} \cdots \begin{pmatrix} x'_{1,T} \\ \vdots \\ x'_{n,T} \end{pmatrix}$
LSTM
$\begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix}$
MLP
$\begin{pmatrix} c_1 \\ \vdots \\ c_K \end{pmatrix}$

Fig. 1. Illustration of the neural network: A length T sequence of graph structured data is fed through a GAT network, producing a sequence of feature vectors. This sequence is fed through an LSTM, producing a single feature vector which can be classified.

The neural network in this work employs a global max-pooling operation to extract a graph embedding following the GAT-layers. The output of the GAT-network is therefore a sequence of graph-embeddings. One common issue in Graph Neural Networks (GNNs) is the phenomenon of over-smoothing [27], where node features tend to become too similar after passing through many GNN-layers. To alleviate this problem and enable higher depth, initial connections, where the input of the network is added to the output of every layer, are added between all layers of the network. To make the network input the correct size, a learnable linear transformation is applied before the first GAT-layer. In addition to this, graph normalization layers [28] are added between every GAT-layer after the initial connection. This is done to prevent problems with exploding or vanishing gradients.

To capture the time-dependent information of the input data, this sequence is then fed through an LSTM network, the output of which is a single feature vector which should capture both the spatial and temporal information of the input data. This feature vector is passed through a final classification layer consisting of a multilayer perceptron (MLP), using the ReLU [29] activation function between hidden layers and a softmax function for the output layer. Dropout [30] is used between every layer of the network. The number of GAT, LSTM and MLP layers, GAT attention heads, hidden nodes at each layer, the amount of dropout and the length of input sequences were determined using the Optuna [31] framework. The general structure of the network is illustrated in Fig. 1.

C. Neural Network Inputs and Outputs

The inputs of the classifier consist of length T time series of voltage phasors and frequency measurements, i.e. synthetic PMU-signals, measured at every bus in the network, together with an edge list which describes the topology of the power system at every time step.

The output of the neural network is a vector with scores for the two labels defined in Sec. III-C. By applying a softmax function to the output vector, these scores can be interpreted as a probability for the sample belonging to either of the classes. Formally, the ouput y for input x is given by

$$y = f_\theta(x) \begin{pmatrix} \hat{p}(x \in C_0) \\ \hat{p}(x \in C_1) \end{pmatrix}, \tag{5}$$

where $\hat{p}(x \in C_k)$ is the estimated probability that x belongs to class i. Theoretically, since the task is binary classification, a single output could be used. However, since an extension of the method could be to correctly identify disturbances, a multi-class formulation was chosen instead.

III. Experimental Setup

In this section the experimental setup, including how data was generated, how training and testing was performed, as well as performance metrics used to evaluate the proposed approach, are presented.

A. Data Generation

The data used to train, test and validate was generated through simulations of the Nordic44 system using DigSILENT PowerFactory. Solid three-phase short circuits with clearing times randomly selected between 160 ms and 420 ms were simulated at every bus and at every line at 25%, 50%, and 75% of the line length. The fault durations were chosen to give a reasonable distribution of stable and unstable cases, with the maximum clearing time chosen within the time delay range for Zone-2 of typical distance relays [32]. The simulations were performed with nominal load and nominal load plus 5%, at different system topologies resulting from disconnecting each line and transformer in the system one at a time. Each simulation was run for 500 cycles, corresponding to 10 s, and the fault was applied after 100 cycles or 2 s every time. This gave a total of 40320 distinct cases. If the rotor angle of any generating unit deviated from its initial value by more than 360° at any point during a simulation, the corresponding case was labeled as unstable, yielding 1812 unstable cases.

B. Balancing

Since a vast majority of the simulated cases showed no instability, the dataset was balanced by randomly selecting 1812 stable cases and discarding the rest. This was done so that the classifier would not be biased in favor of the dominant label. The new, pruned dataset was split into a training set containing 90%, or 3262, of the cases and a test set containing 362 cases.

C. Class Labels

As mentioned, the input data for the neural network consists of length T sequences of graphs. Let $\{\mathcal{G}_T\} = \{\mathcal{G}_{t_1}, \mathcal{G}_{t_2}, \ldots, \mathcal{G}_{t_T}\}$ be one such sequence. Furthermore, let t_f be the time of fault onset. Finally, define the stability status η_{case} by

$$\eta_{\text{case}} = \begin{cases} 0: & \text{no generator out of step} \\ 1: & \text{at least one generator out of step} \end{cases} \tag{6}$$

where a generator is considered out of step if its rotor angle has deviated more than 360° from its initial value. The label $y_{\{\mathcal{G}_T\}}$ for $\{\mathcal{G}_T\}$ is then given by

$$y_{\{\mathcal{G}_T\}} = \begin{cases} 0 \text{ (Stable)}: & t_T < t_f \text{ or } t_T > t_c \ \& \ \eta = 0 \\ 1 \text{ (Unstable)}: & t_T > t_f \ \& \ \eta = 1 \end{cases} \tag{7}$$

D. Training

In this section, hyperparameter optimization (HPO) and training steps are explained in detail. Weighted cross-entropy loss was used for both training and HPO, and both steps used the SGDR optimizer [33] with Nesterov momentum [34] and L_2-decay. In addition to the network parameters, the amount of L_2-decay were also tuned.

1) HPO: To find good hyperparameters for the network, the Optuna [31] framework was used. For each part of the neural network, the number of nodes per layer and the number of layers were determined. For the GAT, the number of attention heads was determined and, finally, the amount of global dropout and the sequence length T were tuned. HPO amounts to training many networks with different hyperparameters, to see which combination maximizes or minimizes some evaluation metric. The networks were each trained for one epoch, i.e. they saw each training sample once. This was performed with half of the available data, since this provided a significant speedup, and minimization of the validation loss was chosen as the evaluation metric. The best hyperparameters found are listed in Tab. I.

TABLE I
NETWORK HYPERPARAMETERS

Parameter	Value
GAT Layers	6
GAT Nodes	32
Attention Heads	8
LSTM Layers	2
LSTM Nodes	32
MLP nodes	32
MLP Layers	3
Dropout	64%
L_2-decay	0.0004
Sequence length	20
Momentum	0.575

2) Training : After hyperparameters were established, regular training was performed. During this step, 20% of the training cases were randomly reserved for validation. The validation set was selected and separated from the training data in this way instead of randomly selecting samples to prevent information leakage between training and validation sets, which could skew the results. The same optimizer and learning rate scheduler were used in this step as during HPO. The decay rate of the learning rate scheduler was kept constant, with two epochs corresponding to a full learning rate cycle. Training was performed for a total of 10 epochs. Checkpoints were taken every time a new minimum validation loss was achieved, and this checkpoint was later used as the final model.

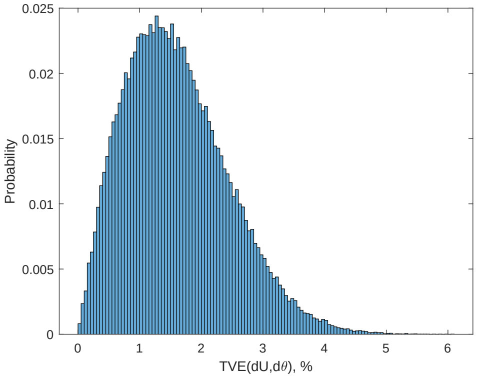

Fig. 2. Normalized histogram showing the TVE of 100,000 samples

The samples were augmented with zero-mean Gaussian noise with standard deviations given in Tab. II. It is worth mentioning that the assumption of Gaussian PMU-measurement noise does not hold in general [35]. However, as observed in [36], for a large number of samples it is reasonable to assume a zero-mean Gaussian distribution. The total number of measurements in the dataset used in this work is in the order of tens of millions per bus, which can be compared to the 60,000-350,000 samples used in [36]. This suggests that the assumption should give a reasonable approximation over the whole dataset. Fig. 2 shows a normalized histogram showing the *total vector error* (TVE) [37] in percent for 100,000 samples. It should be noted that the distribution of Fig. 2 shows errors significantly higher than the 1% TVE allowed for steady state by [37], but should not be unreasonably high for dynamic conditions. For frequency measurements, a signal to noise ratio (SNR) of 40 dB, corresponding to 1%, was assumed, which is higher than that found in [36].

The noise model does not account for different error sources, such as incorrect installation and calibration, time synchronization between units, analog-digital conversion etc. However, based on [36], it should give a reasonable approximation when taken over the whole dataset.

TABLE II
STANDARD DEVIATIONS FOR NOISE USED DURING TRAINING

Measurement	σ
U	1.25%
θ	0.75°
f	1%

E. Performance Metrics

Three metrics are used for evaluating the quality of the labler and early warning system, these being the false alarm FA and missed detection MD rates, as well as the average warning time ψ given by the system measured in cycles. The

FA and MD rates are given by

$$FA = \frac{f_p}{f_p + t_p} \tag{8}$$

$$MD = \frac{f_n}{t_p + f_n} \tag{9}$$

where f_p is the number of false positives, t_p the number of true positives and f_n false negatives. The average warning time is given by the difference in number of reporting cycles from when instability is declared by the system to the first generator going out of step, as defined in Sec. III-C. The average warning time is an important metric as it ultimately determines what kind of automatic remedial actions are suitable in a dedicated System Integrity Protection Scheme to prevent further system degradation. Since the purpose of the classification is to function as an early warning system, the performance per sample is not of interest and the metrics are instead calculated per case.

F. Classification Threshold & Reporting Delay

To better balance the ψ, FA and MD rates, a classification threshold τ is introduced for unstable cases. This means that a sample is labeled as unstable only if $\hat{p}(x \in C_1) > \tau$. Additionally, a reporting delay N is introduced, meaning that in order for the warning system to issue an alert, N consecutive samples have to be labeled as unstable. Both τ and N work to decrease the FA rate, but they do so at the expense of the MD rate and ψ.

IV. Results & Discussion

The results presented in this section are calculated on the reserved test set described in Sec. III-B, both without noise and with added Gaussian noise with standard deviations as given in Tab. II. FA and MD rates as well as ψ were calculated for values of the reporting delay N between 1 and 30, and for 30 equally spaced values of the classification threshold τ between 0.7 and 0.99. Fig. 3 shows two scatterplots of the FA and MD rates for all combinations of τ and N under noise-free (a) and noisy (b) conditions. To find good settings for τ and N, some combinations clearly need to be eliminated. This can be achieved by finding the *Pareto Front* of the set of points in the FA-MD plane [38]. Doing this yields the candidate points shown in Fig. 4. These candidate points are best possible combinations of τ and N, in the sense that for any one of the points, it is impossible to lower either the FA or MD rate by changing τ or N without increasing the other. However, these solutions do not take ψ account. Fig. 5 shows ψ as a function of N for all values of τ for both the noisy and noise-free cases. In both Fig. 5 (a) and (b), it is clear that high values of N result in values of ψ smaller than zero cycles, i.e. the system does not signal an instability until after it has already happened. Excluding settings that give negative values of ψ yields the new candidate points shown in Fig. 6. Tab. III shows numerical values for the points found in Fig. 4 a) and 6 a), and Tab. IV the points found in Fig. 4 b) and 6 b). Interestingly, noise at the level introduced here barely affects performance. This is not unreasonable, as requirements for PMUs in regards to noise levels are understandably strict. Assuming an operator would want both the FA and MD rates as low as possible, with no clear preference for one over the other, Tab. IV shows that reasonable settings could be e.g. $\tau = 0.95$ and $N = 16$ under noisy conditions, yielding both FA and MD rates of 2.21% if ψ is not taken into account. If ψ is taken into account, Tab. IV shows that it is still possible to achieve MD rates of 2.21% and lower, but at a cost of increasing the FA rate above 11%. A reasonable setting here might be $\tau = 0.99$ and $N = 6$, which yields the maximum possible value of $\psi = 9.61$ cycles, with FA and MD rates of 8.20% and 7.18% respectively. It is worth mentioning here that the average time from fault onset until instability in this work is 40.81 ± 10.71 cycles, which means that the time window for making accurate predictions is fairly small. Although the time from fault onset to instability is not directly mentioned in [8], mean warning times of 30.12 and 120.59 cycles are reported for the IEEE 118 bus and IEEE 145 bus systems respectively. From this, it can be inferred that at least compared to those systems, the Nordic44 system reacts fast to transient disturbances. This illustrates that system characteristics are the deciding factor in what performance is achievable for the early warning system. The FA and MD rates found here are higher than those in [8], but the above argument demonstrates that to make a stringent comparison between this work and [8], the same test system should be used. Ref. [9] does not measure the warning time, FA or MD rates, instead focusing on the classification accuracy, which was between 96.78% and 98.92%. The classification of [9] is based on the rotor angles as well as active and reactive power outputs of the generators themselves. This means that the method presented here has to perform one extra step, namely rotor angle estimation based on bus voltages, to get the same quality of inputs. Thus, a direct comparison between the results of this work and those of [9] is not straightforward, but superficially it appears that the method found is comparable in accuracy.

TABLE III
Optimal points including negative warning times for noise free conditions.

τ	N	ψ [cycles]	MD %	FA %
0.76	29	-34.48	0	11.27
0.8	26	-34.53	0.55	8.16
0.91	22	-40.48	2.21	2.75
0.92	20	-36.39	1.66	3.26
0.93	19	-36.31	1.10	3.76
0.93	21	-40.53	2.76	2.22
0.96	15	-25.52	3.31	1.68
0.97	15	-24.99	5.52	0.58
0.98	14	-25.27	6.63	0
0.93	11	0.43	0	16.20
0.95	10	2.14	1.10	13.94
0.96	8	7.51	0.55	15.89
0.98	8	1.33	2.76	9.74
0.98	7	7.39	1.66	11
0.99	7	6.87	7.73	6.18
0.99	6	9.27	5.52	8.56

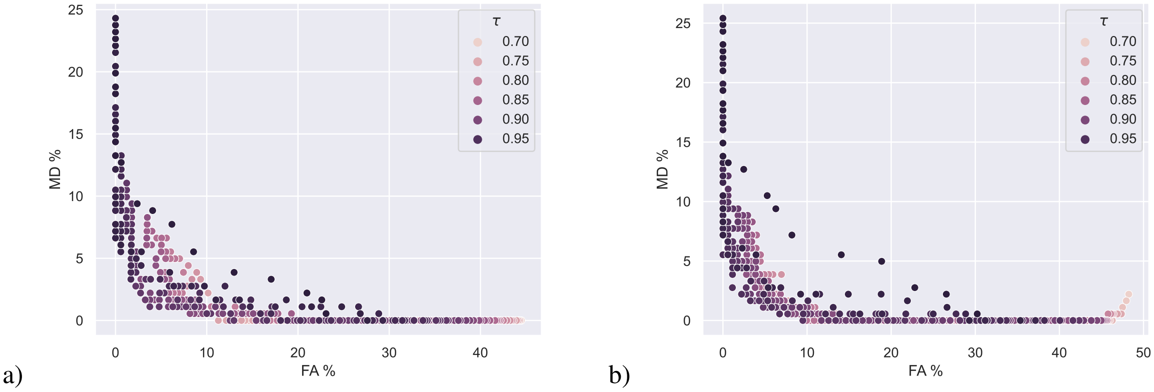

Fig. 3. FA vs MD for different choices of τ and N for noise free conditions (a) and with noise as given in Tab. II (b)

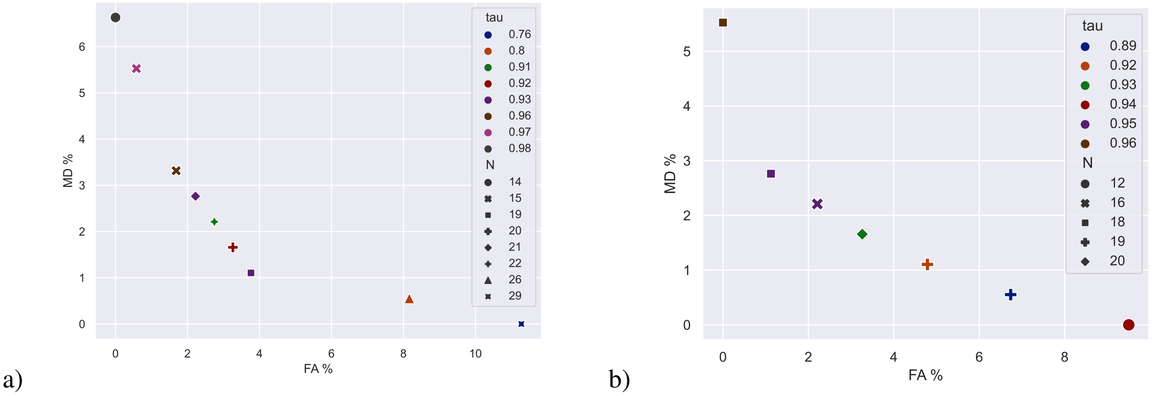

Fig. 4. Pareto efficient points w.r.t. FA and MD rates for noise free conditions (a) and with noise as given in Tab. II (b)

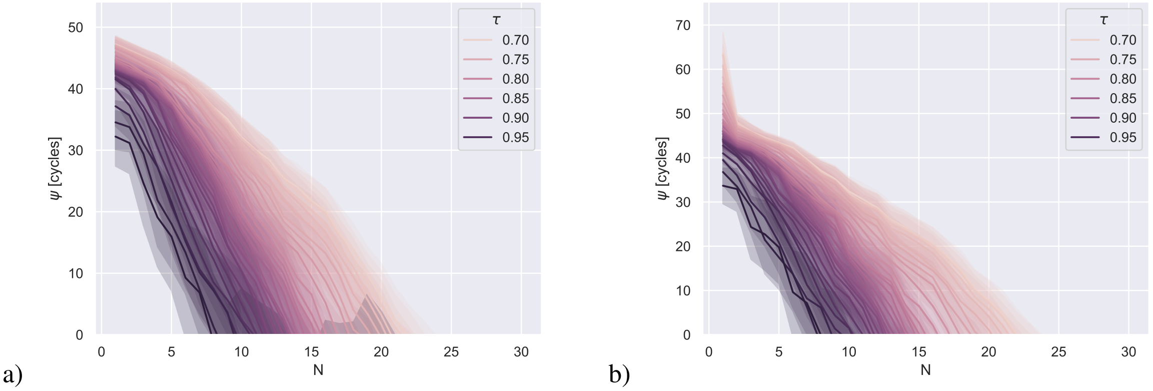

Fig. 5. Average warning time vs N for different values of τ for noise free conditions (a)) and with noise as given in Tab. II (b))

A. Future Work

Since power systems are critical infrastructure, any decision support tool deployed in system operations needs to be tested for robustness, which points out directions for future work.

The methods developed here are, as described in Sec. III-A, trained and tested only under three-phase short circuits with load variations of 5%. An important topic for future work is to curate a more diverse dataset, with larger variations in load and type of disturbances, such as single-phase or line to line short circuits. Another aspect that is important to consider is the effect of the electricity market, and different patterns of import, export and intermittent generation. ENTSO-E provides market

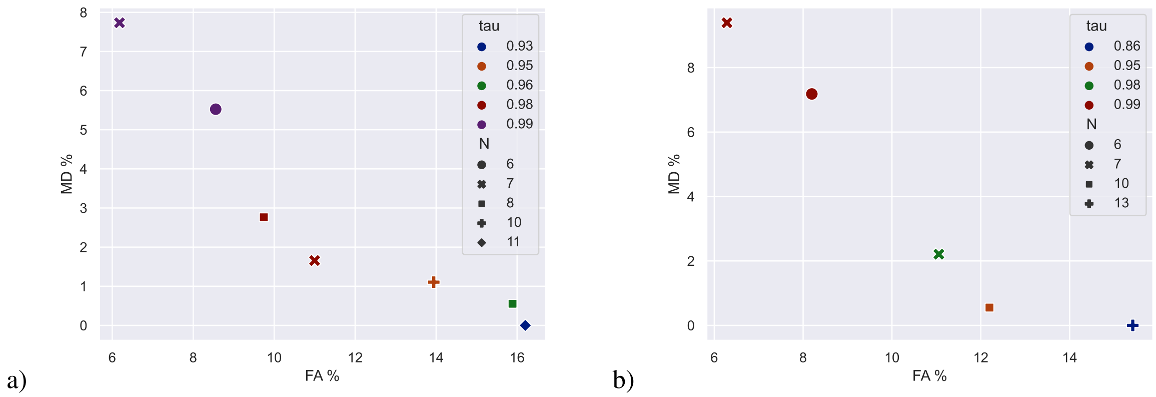

Fig. 6. Pareto efficient points w.r.t. FA and MD rates with $\psi > 0$ for noise free conditions (a) and with noise as given in Tab. II (b)

TABLE IV
OPTIMAL POINTS INCLUDING NEGATIVE WARNING TIMES FOR NOISY CONDITIONS.

τ	N	ψ [cycles]	MD %	FA %
0.89	19	-33.11	0.55	6.74
0.92	19	-40.48	1.10	4.79
0.93	20	-44.90	1.66	3.26
0.94	12	-11.13	0	9.50
0.95	16	-35.38	2.21	2.21
0.95	18	-42.83	2.76	1.23
0.96	18	-41.11	5.52	0
0.86	13	1.19	0	15.42
0.95	10	1.52	0.55	12.20
0.98	7	6.26	2.21	11.06
0.99	6	9.61	7.18	8.20
0.99	7	7.14	9.39	6.29

data spanning several years, which could be used to replicate these patterns, in addition to providing the current share of CIG. When considering the share of CIG, the impact of lower system inertia on the warning times should be of particular concern, but the effect on transient stability in general as well as model performance should also be considered. It would also be relevant to study and explicitly model protection equipment under these conditions as an increasing share of CIG can lead to reduced short-circuit currents, which could in turn affect the function of protection relays [6].

Another important topic for future work is that of availability and quality of real data. Luckily, large disturbances are comparatively rare in real power systems. However, this means that collecting enough real data, under sufficiently varied operating conditions, for training a model would be challenging. A potential solution that is worth investigating is to validate simulated data against real measurements using, for example, the Grid Event Signature Library [39]. In the same vein, validating simulated data against real measurements also provides an opportunity to investigate the impact of real measurement noise. As mentioned in Sec. III-D2, assuming a Gaussian noise distribution is not, in general, valid, and future work should investigate the impact of the noise model on the final results. Additionally, other error sources, such as synchronization errors between PMU-measurements, imperfect topological data gathered from the SCADA-system, and missing data should be taken into account. With this in mind, future work could also consider the impact of combining the method presented here with real-time state-estimation for more accurate input data. However, this could cause further delays which might be detrimental to performance. If a method like the one described in this work was to be implemented in practice, it would be crucial to leverage TSO expertise and experience with operating their system, as well as available proprietary measurement data, in order to represent potential scenarios as accurately as possible.

A final direction for further research is to compare the method developed here against other methods, including those described in Sec. I. There are several evaluation metrics to base such a comparison around, including those presented in Sec. III-E. A major concern when implementing a learning based algorithm is the availability of data. Thus, another important metric to account for would be the accuracy and generalization capability that is possible to achieve when limiting the dataset size, and to provide some indication of how much data is needed. One must also consider what data is available; one benefit of GNNs is that different types of data and covariates can be incorporated by simple concatenation. This is something other methods might struggle with, but to leverage this ability a sensitivity analysis would have to be performed in order to get a fair comparison and optimize results.

V. Conclusion

In this article, a classifier based on a combined GAT and LSTM network was designed and trained to perform power system state labeling based on voltage phasors and frequency measurements at all buses in a power grid. The input to the network consist of a list of edges describing which buses have direct connections via transformers or power lines, together with time series of phasor and frequency measurements at each bus. The labels were then used to analyze a stream of measurements to provide early warnings for angular instability.

It was found that the method was capable of identifying unstable conditions with missed detection and false alarm rates of 2.21% under realistic noise levels, with the ability to tune the system to decrease either the false alarm or missed detection rate at the expense of the other. The maximum achieved average early warning time while simultaneously keeping the false alarm and missed detection rates as low as possible was 9.61 50 Hz cycles. This was achieved at missed detection and false alarm rates of 7.18% and 8.20% respectively. However, the possible average warning time is heavily dependent on how quickly the power system goes unstable following a critical disturbance, and therefore longer average warning times with fewer missed detections and false alarms would be possible in a slower reacting system than the one studied here.

References

[1] Kotzeva Mariana, Timmermans Frans, and Thyssen Marianne, "Sustainable Development in the European Union — Monitoring Report on Progress towards the SDGs in an EU Context — 2018 Edition - Products Statistical Books - Eurostat.," pp. 1–356, 2018.

[2] ENTSO-E and ENTSOG, "TYNDP 2022 Scenario Report | Version. April 2022," tech. rep., 2022.

[3] M. Pavella, D. Ernst, and D. Ruiz-Vega, "TRANSIENT STABILITY OF POWER SYSTEMS A Unified Approach to Assessment and Control,"

[4] M. M. Begovic, "System protection," in *Power System Stability and Control, Third Edition*, pp. 4–1, 6 2017.

[5] M. A. Pai, *Energy Function Analysis for Power System Stability*. Boston, MA: Springer US, 1989.

[6] F. Milano, F. Dorfler, G. Hug, D. J. Hill, and G. Verbič, "Foundations and challenges of low-inertia systems (Invited Paper)," in *20th Power Systems Computation Conference, PSCC 2018*, 2018.

[7] 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 Transactions on Power Systems*, vol. 26, pp. 1474–1483, 8 2011.

[8] A. Gupta, G. Gurrala, and P. S. Sastry, "An Online Power System Stability Monitoring System Using Convolutional Neural Networks," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 864–872, 2019.

[9] J. Fang, C. Liu, L. Zheng, and C. Su, "A data-driven method for online transient stability monitoring with vision-transformer networks," *International Journal of Electrical Power and Energy Systems*, vol. 149, p. 109020, 7 2023.

[10] J. Huang, L. Guan, Y. Su, H. Yao, M. Guo, and Z. Zhong, "Recurrent graph convolutional network-based multi-task transient stability assessment framework in power system," *IEEE Access*, vol. 8, pp. 93283–93296, 2020.

[11] J. Huang, L. Guan, Y. Su, H. Yao, M. Guo, and Z. Zhong, "A topology adaptive high-speed transient stability assessment scheme based on multi-graph attention network with residual structure," *International Journal of Electrical Power and Energy Systems*, vol. 130, no. February, p. 106948, 2021.

[12] S. Gu, J. Qiao, Z. Zhao, Q. Zhu, and F. Han, "Power System Transient Stability Assessment Based on Graph Neural Network with Interpretable Attribution Analysis," *2022 4th International Conference on Smart Power and Internet Energy Systems, SPIES 2022*, pp. 1374–1379, 2022.

[13] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[14] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, pp. 285–317, 7 1985.

[15] D. P. Wadduwage, C. Q. Wu, and U. D. Annakkage, "Power system transient stability analysis via the concept of Lyapunov Exponents," *Electric Power Systems Research*, vol. 104, pp. 183–192, 2013.

[16] S. Dasgupta, M. Paramasivam, U. Vaidya, and V. Ajjarapu, "PMU-Based Model-Free Approach for Real-Time Rotor Angle Monitoring," *IEEE Transactions on Power Systems*, vol. 30, pp. 2818–2819, 9 2015.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[18] S. Brody, U. Alon, and E. Yahav, "HOW ATTENTIVE ARE GRAPH ATTENTION NETWORKS?," in *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.

[19] L. Vanfretti, S. H. Olsen, V. S. Arava, G. Laera, A. Bidadfar, T. Rabuzin, S. H. Jakobsen, J. Lavenius, M. Baudette, and F. J. Gómez-López, "An open data repository and a data processing software toolset of an equivalent Nordic grid model matched to historical electricity market data," *Data in Brief*, vol. 11, pp. 349–357, 4 2017.

[20] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 5 2015.

[21] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, pp. 18–42, 7 2017.

[22] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[23] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4–24, 1 2021.

[24] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.

[25] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, pp. 61–80, 1 2009.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, Neural information processing systems foundation, 6 2017.

[27] T. K. Rusch, E. Zurich, M. M. Bronstein, and S. Mishra, "A Survey on Oversmoothing in Graph Neural Networks," 3 2023.

[28] T. Cai, S. Luo, K. Xu, D. He, T.-Y. Liu, and L. Wang, "GraphNorm: A Principled Approach to Accelerating Graph Neural Network Training," 2021.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks,"

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[31] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, Association for Computing Machinery, 7 2019.

[32] G. Ziegler, *Numerical distance protection: principles and applications*. John Wiley & Sons, Ltd, 2011.

[33] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[34] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," 2013.

[35] S. Wang, J. Zhao, Z. Huang, and R. Diao, "Assessing Gaussian assumption of PMU measurement error using field data," *IEEE Transactions on Power Delivery*, vol. 33, no. 6, 2018.

[36] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," *IEEE Power and Energy Society General Meeting*, vol. 2016-Novem, 11 2016.

[37] "IEEE Standard for Synchrophasor Measurements for Power Systems.," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, 2011.

[38] S. Nayak, "Multiobjective optimization," *Fundamentals of Optimization Techniques with Algorithms*, pp. 253–270, 1 2020.

[39] S. Biswas, J. Follum, P. Etingov, X. Fan, and T. Yin, "An Open-Source Library of Phasor Measurement Unit Data Capturing Real Bulk Power Systems Behavior," *IEEE Access*, vol. 11, pp. 108852–108863, 2023.