

Received 15 June 2023, accepted 4 July 2023, date of publication 12 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294472

RESEARCH ARTICLE

Online Identification of Cascading Events in Power Systems With Renewable Generation Using Measurement Data and Machine Learning

GEORGIOS A. NAKAS¹, (Graduate Student Member, IEEE), ALARA DIRIK²,
PANAGIOTIS N. PAPADOPOULOS¹, (Member, IEEE),
AMARSAGAR REDDY RAMAPURAM MATAVALAM³, (Member, IEEE),
OLIVER PAUL⁴, AND DIMITRIOS TZELEPIS¹, (Member, IEEE)

¹Department of Electronics and Electrical Engineering, University of Strathclyde, G1 1XQ Glasgow, U.K.

²Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey

³School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA

⁴CheckSolar, 40219 Düsseldorf, Germany

Corresponding author: Panagiotis N. Papadopoulos (panagiotis.papadopoulos@strath.ac.uk)

The work of Georgios A. Nakas was supported by the Engineering and Physical Sciences Research Council (EPSRC) Research Excellence Award. The work of Panagiotis N. Papadopoulos was supported in part by the SuperGen Energy Networks Hub under Grant EP/S00078X/1, and in part by the U.K. Research and Innovation (UKRI) Future Leaders Fellowship under Grant MR/S034420/1. The work of Dimitrios Tzelepis was supported in part by the SuperGen Energy Networks Hub under Grant EP/S00078X/1.

ABSTRACT This paper introduces a framework for online identification of cascading events in power systems with renewable generation, based on supervised machine learning techniques and measurement data. Cascading events are low-probability, high-impact events, the propagation of which can lead even to large-scale blackouts, with severe consequences to society. The proposed methodology is based on Long-short term memory networks, considering uncertainties associated with renewable generation, system loading and initial contingencies. By utilizing time-series measurement data, the proposed method can predict the appearance of cascading events, as defined by the discrete action of protection devices which can capture voltage, frequency or transient instability related dynamic phenomena. The proposed framework is applied on a modified version of the IEEE-39 bus model incorporating detailed dynamic renewable generation and protection devices implementations. Results highlight that the suggested method can successfully identify cases with cascading events with up to 95.6% accuracy and with an average inference time of 0.042s, taking into account practical considerations related to phasor measurement units, such as availability and noise in measurement data.

INDEX TERMS Cascading failures, dynamic simulation, machine learning, phasor measurement units, renewable generation.

I. INTRODUCTION

In modern power systems, the uncertainty that comes with the integration of renewable energy sources (RES) penetration, makes the online dynamic security problem a challenging task. The highly complex, non-linear behaviour of electrical power systems is not yet well understood, creating the need to re-establish stability definitions [1]. In some occasions, the unpredictable response of a system to a contingency can cause the appearance of cascading events, compromising its secure

The associate editor coordinating the review of this manuscript and approving it for publication was Sarasij Das¹.

operation. For this reason, intelligent approaches that are able to predict unstable behaviour by using real-time measurement data, coming from Phasor Measurement Units (PMUs) that are nowadays available, are being investigated to ensure the secure operation of modern power systems with increasing renewable penetration.

The accurate representation of protection devices is a key element in capturing the cascading events that might appear in a system following a contingency [2], [3]. In some cases, protection devices might activate before instability limits are reached. Their action can also cause subsequent events, leading to the appearance of cascading event sequences. A method

to predict this behaviour can provide valuable information about the online system state, enabling system operators to take corrective actions in order to prevent cascading events from spreading.

A platform designed for online static and dynamic security assessment is described in [4]. This method, incorporating machine learning techniques, takes into account load and renewable generation related uncertainty and can also provide possible control actions in order to avoid insecure operation. The offline phase of this platform is presented in [5], where a large amount of dynamic simulations, for various operating conditions and contingencies, is performed and decision trees are used to extract security rules.

Various machine learning techniques have been applied to address the problem of transient stability assessment. A method based on decision trees and hierarchical clustering is presented in [6]. The proposed methodology, considering the impact of RES penetration, identifies the unstable generator groups and the order in which these groups become unstable. This method, trained for specific network topologies, achieves a high performance. As it is concluded, the uncertainty that comes with the penetration of RES affects the network dynamic behaviour. In [7], the original network data is transformed into an abstract representation state using a deep belief network. During the learning process, the power system topology is considered and an index is used to tune the parameters of the deep belief network. The classification of unstable and stable cases is achieved using a linear model on the representation space. In [8] online transient stability is approached using long short-term memory (LSTM) networks. In this method the model can be informed from the temporal data dependency, which according to the results leads to increased accuracy. Also, in this paper a sensitivity study related to PMU measurements is carried out, using a sequential feature selection algorithm.

In [9] a real-time transient stability index as a measure of the distance to instability is introduced. Online measurements are used to define the parameters of the dynamic equivalent model, which includes an aggregated generator and relevant controllers of each area, reducing the model order and complexity. The method is applied on a large power system, demonstrating high performance for predicting early stages of unstable conditions. In [10], transient stability assessment is approached by an analytical method using a specific form of Lyapunov functions. The algorithm introduced in this study, chooses the best-suited function to specific contingency cases. Another analytical method for transient stability is presented in [11], by utilising stochastic continuous disturbances, which are brought to modern power systems due to sources of uncertain nature, such as converter-connected generators and electric vehicles.

In [12] the authors present an online transient stability assessment scheme using bitmaps, consisting of trajectories acquired from PMUs installed at generator terminals. These bitmaps are used to train a convolutional neural

network (CNN). The results showcase higher performance of CNNs compared to several conventional machine learning classifiers, such as support vector machines, decision trees and random forests. A method with a similar use of bitmaps and CNNs is introduced in [13], in this case the method is applied on addressing small-signal stability.

A few methods have also been proposed in order to predict cases of voltage instability. The online short term voltage stability (SVS) assessment is addressed in [14] using an LSTM-based algorithm. Learning from both spatial and temporal information, the proposed framework showcases high accuracy and reliability. A random-forest-based method is presented in [15] for real-time voltage stability identification. The results suggest that this methodology can predict voltage instability cases fast enough, leaving time to take control actions and mitigating these events. In [16], a PMU-based method is proposed to predict SVS, combining support vector machine and online learning. The voltage stability status is identified in advance based on time-series prediction and can be applied on both symmetrical and asymmetrical fault conditions. While studies [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] do not focus directly on the prediction of cascading events, they provide valuable insights about the application of data-driven methods for the online identification of the power system state following a contingency.

The above mentioned methods, mainly aim at identifying specific stability or security related phenomena and do not investigate the online identification of cascading events, defined by the action of protection devices. However, in certain cases protection devices might activate before instability limits are reached. So far, methods predicting cascading events have been focusing on static simulations. A method for the prediction of the number of line failures and the amount of load shed given an initial operating point is introduced in [17], with a dataset consisting of DC power flow simulation results. In [18] an influence model based on a hybrid learning method is used for the prediction of cascading failures from simulations carried out on a DC/AC power flow model. A probabilistic approach for predicting cascading events using a support vector machine model and static simulations is presented in [19]. An early method for the prediction of cascading events is described in [20], using the creation of temporal trees to predict the appearance of events only related to voltage collapse phenomena. The method proposed in [21] utilizes a Graph Convolution Network (GCN) with spatio-temporal properties to predict cascading failures, in a network with renewable generation and protection devices. However, the method focuses on investigating topological aspects and practical implementation aspects in an online setting related to the availability of PMU measurements, are not discussed.

As results from [22] and [23] have shown, dynamic simulations can capture cascading events in more detail than static, as some of the events during the later stages of

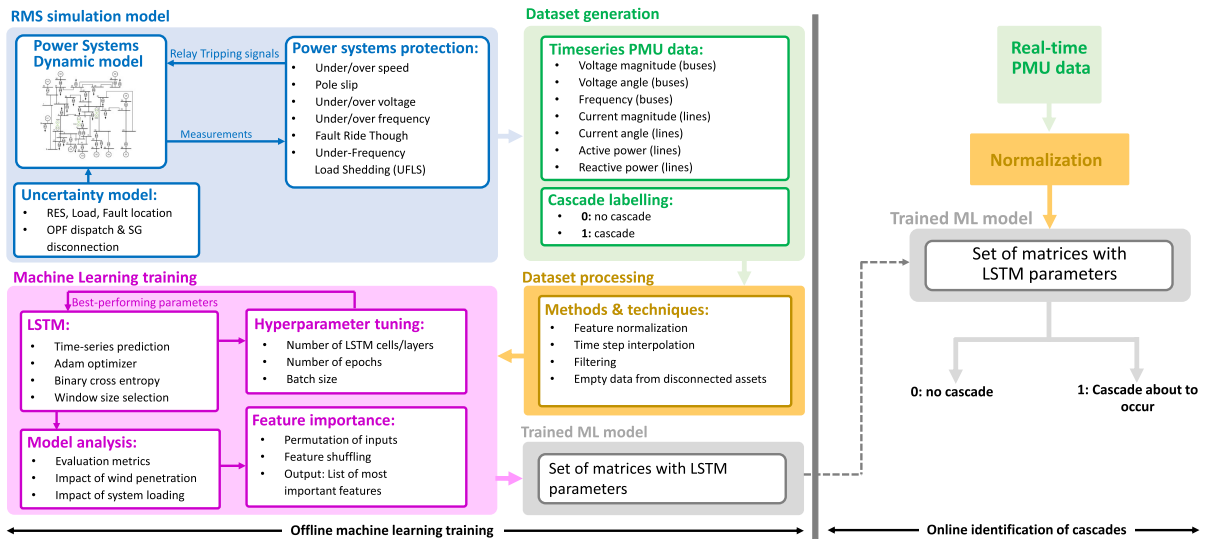


FIGURE 1. Flowchart illustrating the steps of the proposed framework.

cascading sequences are captured only by dynamic models. Using dynamic models to capture, and consequently predict cascading events can provide a more realistic real-life representation of power systems operation. Combined with the need for dynamic simulations, [2], [3] highlight the importance of representing protection devices in capturing the evolution of cascading events, an approach our proposed method is following. More importantly, [24], [25] highlight the fact that not including protection devices in dynamic studies might result in inaccurate assessment of system behaviour.

So far, existing methods have been focusing on online dynamic security, or individually on transient, small-signal or voltage security of power systems as defined by stability limits. The prediction of cascading events has only been examined using data from static simulations. To the authors knowledge, an approach for the online identification of cascading events defined by the activation of protection devices using time domain measurements has not yet been proposed.

The main contribution of this study is the use of a LSTM model for the prediction of the appearance of cascading events using time-series measurement data from dynamic RMS simulations. The realistic representation of cascading events is achieved by the accurate modelling of system dynamics (capturing phenomena related to voltage, frequency and transient instability), the implementation of the action of protection devices, and the consideration of RES penetration along with the uncertainty that comes with it. This approach for online identification of cascading events makes the proposed method distinct to stability/security assessment methods mentioned before, with the key reason being that the actual limit where an event might propagate is more accurately represented, as it includes the action of protection devices and system dynamics [3], [22]. As the prediction takes place in close to real-time, this information could be vital in taking corrective control actions in time and preventing cascading events from spreading. For this reason,

a fast prediction time is critical, especially during the fast propagation phase of cascading events that can lead to load shedding events or blackouts [26]. The proposed model-based method can be trained offline from time-series data produced by detailed dynamic simulations and provide online a prediction with a fast inference time, as compared to methods based on Monte Carlo simulations that can be time-consuming and focus on longer time-scales [3]. Other contributions include: i) the investigation of the impact that the time window length, used for the online prediction, has on the model performance. ii) How the performance of the prediction model differentiates for individual operating conditions. As it is concluded from the results, the model performance can vary for different system loading and wind penetration, which can offer useful information to system operators, related to the level of confidence when using the method. iii) A feature importance analysis is also performed to identify which of the features play a significant role in the prediction of the onset of blackouts. This can offer interesting information on the parameters affecting cascading events as well as identify the specific PMU measurements that have the highest impact on the model performance, which can inform measurement infrastructure decisions. iv) The model performance is evaluated considering limited availability of PMU measurements and noisy data, which can be found in practical applications.

II. METHODOLOGY

The proposed framework aims to the online identification of cascading events followed by an initial disturbance. A schematic illustrating the main steps of the framework online and offline stages, which are described in detail below, is presented in Fig. 1.

A. DETAILED PROCEDURE

The method presented in this paper consists of two main stages: i) the offline generation of the dataset and the training

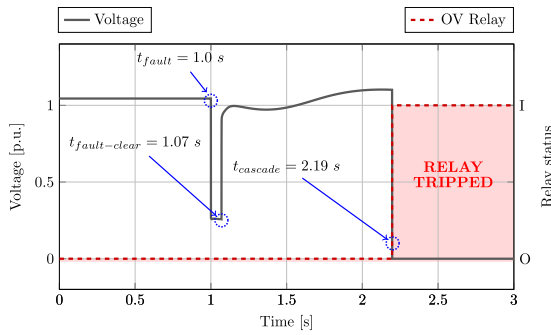


FIGURE 2. Example of a cascading event and the method application.

of the appropriate supervised machine learning model, and ii) the online binary classification using the pre-trained model to predict the appearance of cascading events. A cascading event sequence in this study is defined as the sequence of events that are caused by the intentional activation of protection devices following the initial disturbance and consequent disconnection of the line to clear fault. The appearance of cascading events due to hidden failures (e.g. related to equipment failure or human error), is out of the scope of this paper. During the offline stage, a number of dynamic RMS (Root Mean Square) simulations for various initial operating conditions and contingencies is performed, taking into consideration the increased uncertainty that comes with RES penetration and the reduced network inertia caused by SG disconnection as dictated by economic dispatch. The initial applied fault may cause the appearance of cascading events, as dictated by the discrete action of protection devices that have been implemented in the system. The time-series data obtained from the dynamic simulations are pre-processed to represent a typical PMU sampling rate and are subsequently used for training the model. For this method, Long-short term memory networks (LSTM) have been used, because of their ability to store information and to learn from time series dependencies. This approach is compared to the performance of a regular feed-forward neural network, as a baseline model, and to the performance of a simple recurrent neural network (RNN).

In practical applications, the time domain measurement data during the online phase can be obtained from PMU measurements [27] and used as input to the pre-trained machine learning model to predict the appearance of cascading events. It should be noted that in this study, measurement data from simulations have been used for training and testing the method. The analysis of the model performance is related to the window size and the performance for different loading and RES penetration levels. A feature importance analysis using the pre-trained model is then carried out to identify the most important features. Taking into account practical applications, the model performance is evaluated considering limited availability and noise of measurement data.

An example of a cascading event and the application of the proposed method is shown in Fig. 2. In this plot, the cause of a cascading event, the tripping of a wind generator (NSG2) due to over-voltage simulated in the test network used in this

study, and the signal of the protection relay are presented. After the fault clearance (1.07s), as the bus voltage recovers it causes the violation of the over-voltage protection limits (in this case 1.1 p.u. for over 0.15s) that leads to the activation of the over-voltage relay and the disconnection of the wind generator from the grid (2.19s). The tripping of this wind generator may cause the violation of other protection device limits, causing the tripping of more components and creating a sequence of cascading events. Predicting the possibility of voltage instability in this case might not capture the tripping of this element and any subsequent events. This highlights the importance of capturing, and subsequently being able to predict, the action of protection devices, and not just the instability mechanisms involved.

Time domain features, that can be obtained from typical PMU measurements as the voltage measurement presented in this plot, prior to the cascading event are used to train the machine learning model. The online application of the proposed method starts after the initial fault clearance, by utilizing the pre-trained model in order to predict the appearance of the cascading event before the actual time of the event. In this example, the method should be able to predict the appearance of the tripping of the wind generator before the positive signal of the protection relay at 2.19s. This is the time window during which the pre-trained model has to make the prediction. In general, a shorter time window is beneficial, as an earlier prediction could provide more time for any corrective actions that can be made before the appearance of the cascading event. On the other hand a longer time window that consists of more time steps, could provide more information about the evolution of the system response, after the initial applied contingency. Thus, the size of time window depends on the specific application and the ability of the model to learn from sequential data.

B. MODELING OF SYSTEM DYNAMICS

In order to effectively capture the appearance of cascading events, the accurate modelling of the various mechanisms related to voltage, frequency and transient stability as well as protection devices is of significant importance. While the modelling and simulation approach does not contain significant novelty on its own, the ability to capture complex dynamic phenomena using machine learning models for time-series through our proposed method is novel.

The synchronous generators (SGs) are represented by full detail four winding models (6th-order), equipped with Automatic Voltage Regulator (AVR), Power System Stabilizer (PSS), and Governor (GOV). The renewable generation of the system is represented by International Electrotechnical Commission (IEC) Type 4A wind turbines.

The following protection devices are implemented in the network to capture the above mentioned dynamic phenomena. The SGs are equipped with an under-/over-speed protection relay, an under-voltage protection relay and pole-slip protection. The wind generators are protected with an under-/over-voltage protection relay with fault-ride through (FRT)

and an under-/over-frequency protection relay. An Under-Frequency Load Shedding (UFLS) scheme with four stages is also implemented for the disconnection of a percentage of demand at low frequency to restore the active power balance and try to avoid frequency collapse. More details about the protection devices settings can be found in [28].

The action of Load Tap Changers (LTCs) and Over-excitation Limiters (OELs) has been also implemented within the model, in order to capture longer phenomena related to voltage instability [29]. The duration of the RMS simulations has been set to 120s to capture both fast and slower evolving dynamic phenomena. In large power systems there is a wide range of uncertainties including load variation and RES, that can affect the dynamic behaviour of the system. In this study, the sampling of possible initial operating conditions is based on the discretization of the operating range of the variables of interest, towards creating a large data-set of operating scenarios and events. The parameters considered for this purpose are RES output generation (for each RES unit), system loading and line fault location.

C. DATASET GENERATION

After the sampling of wind generation output and system loading values, which are discretized within a certain step, an AC OPF problem is solved to determine the dispatch of the SGs. Each SG is allocated a cost curve, which can be found in [30], establishing a merit order between them. The objective of the OPF problem is the minimisation of the total synchronous generation cost, while respecting constraints set by the active and reactive power limits of the generators, the maximum loading of the lines and the bus voltage limits. For the initial operating conditions considered in this study the OPF problem can successfully converge in all cases. An amount of conventional SG is also disconnected to represent the reduction of inertia caused by penetration of RES. To achieve this, it is considered that each generator consists of 4 identical machines. According to the operating point of each SG as it is calculated by the OPF solution, the number of machines for each generator that are needed to be connected is determined.

Next, the line on which the fault happens is selected. Three phase faults on lines are considered as an initial contingency. The fault occurs at $t=1$ s and gets cleared by disconnecting the faulted line after 70ms. If the network conditions following the contingency cause the activation of the protection devices implemented in the system, the protection devices trip the system component and a cascading event occurs. This component disconnection might potentially lead to consecutive cascading events. More details on the dataset generation are provided in V-G.

The time series data of each simulation is obtained, with a total of 178 features describing the states in various power system locations over time. These features represent the measurements that can be obtained from PMU devices found in real power systems and include the voltage and frequency of every bus element, and the current, active and reactive power

of every line of the network. At the end of each simulation it is determined whether the system remains secure or if any cascading events occur, and each simulation is labelled as 0 or 1 respectively.

D. PREPROCESSING DATA

In order to convert the dataset into the input format expected by the selected machine learning model, a number of pre-processing steps are performed, consisting of feature normalisation, time step interpolation and windowing. All features are normalized to enable more efficient and high performance model training. Normalization is a widely used pre-processing technique for data smoothing that aims to retain information related to within feature variance, while ensuring that all features are on the same scale. In this study, the scaling value for all quantities in per unit (p.u.) has been set to 1, and for all the other quantities it has been set to 100. After application all the measurements values are in the range of [-10,10]. Without this normalization step the model training resulted in 9.31% lower accuracy.

After normalizing all features, we perform interpolation to ensure evenly sampled time steps across all simulations. We use first order spline interpolation and set the time interval δ to 0.01 seconds, a typical PMU sampling rate. The interpolation step both ensures a smoother cost function by avoiding drastic changes in feature values across two time steps and prevents performance drops in production, where model inference takes place at fixed intervals.

III. NEURAL NETWORK MODELS AND TRAINING

A. RECURRENT NEURAL NETWORKS (RNN) AND LONG-SHORT TERM MEMORY NETWORKS (LSTM)

Recurrent neural networks, also known as RNNs, are a class of deep neural networks designed to model data with temporal qualities where the order of data points is important (sensor data, natural language, speech, etc.) In RNNs each data point is fed into units called cells and gets transformed. Furthermore, the output of each cell is fed into the next cell along with the next data point, essentially creating a long-term memory structure where each layer takes all previous values into account. The transition function is given by:

$$h_t^l = f \left(Ux_t^{l-1} + Wh_{t-1}^l \right) \quad (1)$$

where h_t , h_{t-1} are the hidden states at time step t and $t-1$ respectively, W and U are weight matrices, l is the layer number, x_t is the input at time step t and f denotes the activation function.

However, RNNs suffer from a significant problem: vanishing or exploding gradients. To train a neural network, the total loss is back-propagated through all layers and gradient descent is performed in order to minimize the contribution of each parameter to the loss by updating their weights. Hence, weight updates that are too small or too large can cause the gradients of the parameters in earlier layers to vanish or explode.

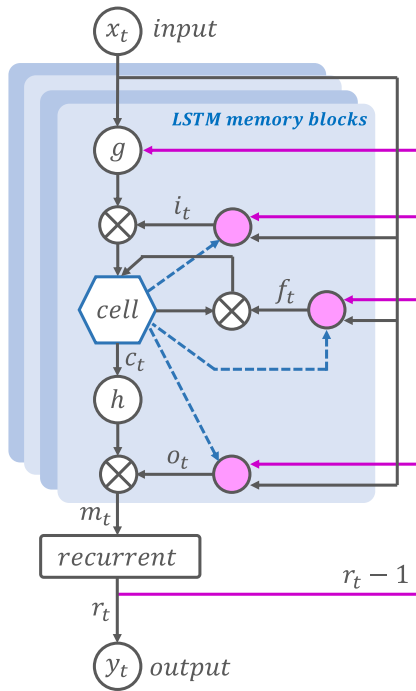


FIGURE 3. A schematic diagram of a LSTM memory cell.

The LSTM networks [31], [32] are a kind of recurrent neural network that aims to resolve the vanishing / exploding gradient problem. LSTMs use memory cells with input, output and forget gates to maintain information for longer periods and regulate the flow of information. LSTMs can decide to overwrite the memory cell, retrieve it, or keep it for the next time step, hence maintaining both long and short term memory depending on the task and context. Moreover, the long-term memory is stored in a vector of memory cells $c_t^l \in \mathbb{R}^n$. A schematic diagram of a memory cell is shown in Fig. 3.

RNNs, including LSTMs, can map one to many, many to many or many to one. For example, given an input sequence $x = (x_1, \dots, x_T)$ and target output sequence $y = (y_1, \dots, y_T)$, the LSTM network unit activations can be calculated iteratively from $t = 1$ to T with the following equations [33]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (2)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (4)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (5)$$

$$m_t = o_t \odot h(c_t) \quad (6)$$

$$y_t = \phi(W_{ym}m_t + b_y) \quad (7)$$

where the W denote weight matrices, b denote bias vectors, σ denotes the sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the cell output activation vector m , \odot is the element-wise product of the vectors, g and h are the cell input and cell output activation

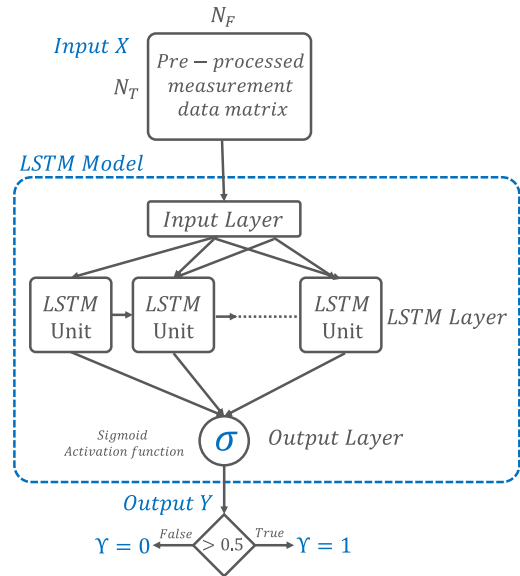


FIGURE 4. Structure of the LSTM model.

functions, \tanh and ϕ are the hyperbolic tangent and softmax activation functions respectively.

B. USING LSTMS TO PREDICT CASCADING EVENTS

Due to the particular properties of LSTMs explained previously, they offer a good fit for the problem of predicting cascading events. LSTMs can handle time-series data and their memory properties also fit well with the need to capture the evolution and inter-dependencies of the system variables as they evolve in time, an important aspect affecting cascading events. In order to predict the occurrence of cascading events, an LSTM model is trained using the pre-processed data, described in Sections II-C and II-D, as input. The input X of the model is a $N_F \times N_T$ matrix, where N_F is the number of features and N_T is the number of time steps included in the selected time-window (input size). The time-window length is investigated in Section V-A.

To pose the occurrence of a cascading failure as a binary classification problem, the final layer consists of a single neuron that is fed into a sigmoid activation function to output a value between 0 and 1, that represents the probability of a cascading event occurring or not. The threshold is set to 0.5, if the output probability is higher than the threshold then Y is set to 1 (a cascading event will occur), otherwise Y is set to 0 (no cascading event). We use the cross entropy between the model predictions and real values (1 for failure cases, 0 for non-failures) as our loss function, compute each parameter's contribution to the total loss via back propagation and perform batch gradient descent to optimize the weights of the model parameters, as explained in more detail below. The structure of the proposed model is presented in Fig. 4.

C. MODEL TRAINING

To train the LSTM models, we use the pre-processed dataset as outlined in Section V-G and perform a stratified split using

a ratio of 80-10-10 % to create training, validation and test sets. We use a single layer LSTM, where the number of hidden units/neurons is set to 150. The size of the hidden units is chosen based on model performance after performing a grid search for the following values: {50, 100, 150, 200, 250}. We use the Adam optimizer and binary cross entropy as our loss function, a common choice for binary classification problems. To compare the performance of LSTM, we train additionally a feedforward Multilayer Perceptron (MLP) and a simple RNN network. As a baseline approach, the MLP consists of an input layer with the number of neurons set equal to the number of input data points (number of features \times time steps) and a single hidden layer with 300 neurons, as set following a grid search. The rectifier linear unit (ReLU) activation function is chosen for the MLP, to capture the nonlinear behaviour. The number of hidden units for the RNN is set after performing a similar grid search as for the LSTM model.

Unlike vanilla gradient descent where the model parameters are updated at each data sample, batch gradient descent is used to perform back propagation and parameter updates over batches of input data. Using batch gradient descent helps overcome memory constraints and increases computational efficiency. At each optimization iteration, the model parameters are shifted in the opposite direction of their respective gradients (with respect to loss) by a configurable step size, known as the learning rate. Moreover, once all the batches are iterated, the dataset is shuffled and reiterated to prevent getting stuck in local minimas and help the weights of the model parameters to converge. Each complete iteration of the training dataset is called an epoch. Based on the size of the dataset, the batch size is set to 64. Furthermore, we use the default learning rate value of 0.001 and train the models for 10 epochs on a single GPU with early stopping enabled (based on validation loss) to avoid over-fitting.

Because of the stochastic nature of neural network algorithms, the same network trained on the same data can produce different results. To ensure reproducibility, we set the model seed to 17 during the model training process. Our feature importance and time window length experiments in Section V show that our models perform well regardless of the data split.

As observed in Fig. 5, where the evolution of the training and validation loss is presented across the epochs, no over-fitting is observed and the model has converged towards the end of training. Moreover, we observed that models tended to overfit after 10 epochs (training loss decreases while validation loss increases). Once the model is trained, we perform inference on the test set and compare the predicted against the true labels.

D. EVALUATION METRICS

To evaluate the performance of the proposed LSTM binary classifier, the metrics presented in (8)-(11) are used. Accuracy, Precision, Recall and F1 score are typical measures used in machine learning that capture different aspects of the

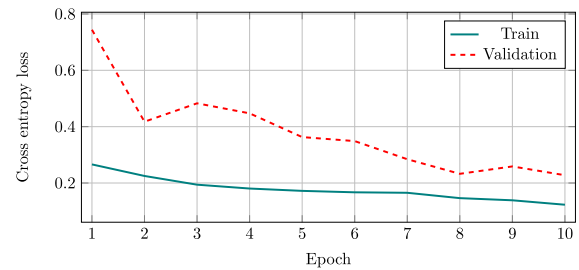


FIGURE 5. Train and Validation learning curves of the proposed LSTM model.

performance of a binary classifier [34].

$$Accuracy (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN}} \quad (8)$$

$$Precision (\%) = \frac{n_{TP}}{n_{TP} + n_{FP}} \quad (9)$$

$$Recall (\%) = \frac{n_{TP}}{n_{TP} + n_{FN}} \quad (10)$$

$$F1 \text{ Score } (\%) = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (11)$$

where n_{TP} , n_{FP} , n_{TN} and n_{FN} is the number of true positive, false positive, true negative and false negative predictions respectively.

In this case, true positives are the correct predictions of cases with cascading events and false positives are the cases with cascading events that are falsely predicted as safe cases. True negatives are the safe cases (no cascading events) that are correctly predicted, and false negatives are the safe cases that are incorrectly predicted as cases with cascading events. The confusion matrix that presents these values in a table format, is also examined.

These metrics can provide valuable information about the task of classifying whether or not a cascading event will occur: *Accuracy* describes the percentage of correct predictions. *Precision* describes the percentage of the cases predicted to include cascading events that is actually correct and *Recall* the percentage of actual cases with cascading events that is predicted correctly. *F1 Score* is a metric that combines *Precision* and *Recall*, and it is defined as the harmonic mean of these two metrics. We note that there is almost always a trade-off between recall and precision with datasets of limited size. Models with high recall-low precision and low recall-high precision performances can be interpreted as overfitting and underfitting respectively. In this particular application, a false negative is more critical than a false positive as missing a real failure event might lead to subsequent cascading events or even a widespread blackout. Thus, a high *Recall* is more important in our case.

In some cases, the first failure of the cascading event occurs too early and this makes it impossible to make a prediction within the selected time window. We define these cases as missed cases. In order to identify the time window that leads to the best performing model, a new accuracy metric, *Accuracy'*, is defined. This metric describes the percentage

of correct predictions that accounts for the missed cases:

$$Accuracy' (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN} + n_{MC}} \quad (12)$$

where n_{MC} is the number of missed cases.

E. PERMUTATION FEATURE IMPORTANCE

A feature importance analysis is performed to investigate the effect of each feature on model performance, with the goal of identifying the most important features and consequently system variables corresponding to them. These features represent time domain measurements that describe the measured electrical variables of the system and can be acquired by PMUs in practical applications. As in large real-life power systems a certain number of PMUs is installed and in certain locations, it is of great importance to investigate in which way and to what extent these measurements affect the prediction of cascading events. This can also contribute to better understanding the mechanisms involved behind the appearance of cascading events by identifying important system variables that might affect the evolution of cascades.

The concept of permutation feature importance is permuting each time a feature and calculating the model performance. More specifically given a sequence of n timesteps, the time order of all features except the feature to be permuted remains the same while the selected feature column is shuffled, breaking the time-order. Since LSTMs are recurrent neural networks that expect ordered time-series as input, a permutation of an important feature would cause a drop in accuracy, as described in [35]. For feature importance analysis, we permute each input feature one by one and compare the performance of the model on the test dataset to the performance of the original model. As this method is applied on the pre-trained model, it does not require training the model again, thus being computationally efficient.

IV. TEST SYSTEM

A. POWER SYSTEMS DYNAMIC MODEL

In this study, a modified version of the IEEE-39 bus model (Fig. 6) is used, implemented using dynamic RMS simulations in DIgSILENT PowerFactory. The original network model has been modified with the addition of three wind farms that are connected to three different network buses. The total installed capacity of the three wind farms is considered to be equal to 20% of the total installed conventional generation of the original IEEE-39 bus system case.

The loads are modelled as balanced three-phase constant impedance loads and are connected to distribution voltage rated buses via step-down transformers, that have been added to the original system case. These transformers are equipped with LTCs, which adjust the transformer ratios keeping the distribution voltage within the deadband [0.99-1.01] p.u. The LTCs settings have been adopted from [29]. The protection devices, as described in Section II-B, have been added to the system and have been implemented using standard models found in the DIgSILENT Powerfactory library. The relays

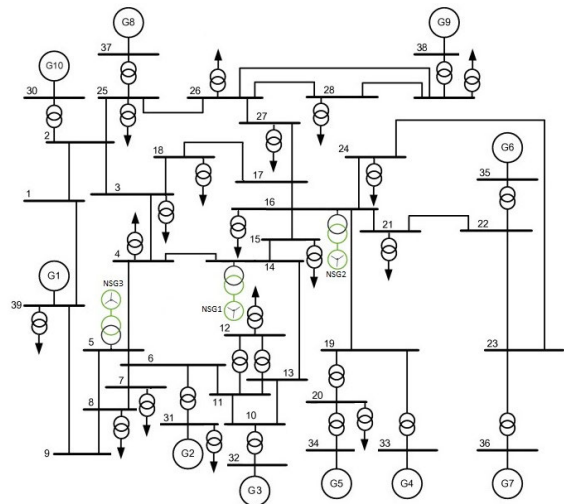


FIGURE 6. Modified version of the IEEE-39 bus model.

settings comply with the transmission system limits from the UK grid code [36]. More information on the modelling procedure can also be found in [37].

B. CASE STUDIES

In this study the system loading is assumed to range from 70% to 120% of the total network demand (as calculated in the base case) and the output of each of the three wind generators in the network is assumed to range from 0 to 100% (of the nominal capacity of each wind generator). In order to define the case studies, a deterministic approach is followed, by discretizing the search space as defined by system loading and the output of wind generators within certain steps. A 10% step is used for sampling system loading values and a 20% step for sampling the output of each wind generator. This sampling approach ensures that equally divided areas of the search space are taken into consideration. Three phase faults in the middle (50% length) of each line are considered as initiating events, considering every network line (34 total lines). That gives 34 different cases for each given network operating condition, multiplied by 8 different loading scenarios and by 6 different RES output scenarios. In total, 44064 cases have been simulated in this study, with cascading events appearing in 7131 cases (16.2% of simulated cases).

The percentage of cases with cascading events is higher compared to practical applications, as the lines that are disconnected as initial contingencies in reality could be comprised of double circuits. So, each initial contingency represents potentially the disconnection of two parallel circuits at a time. Consequently, in some cases the disconnection of a line causes an area of the system to become islanded, which leads to the appearance of cascading events. The reason for stressing the power system operation is to be able to observe more cases of cascading events and include these conditions in the training of the model for the following binary classification.

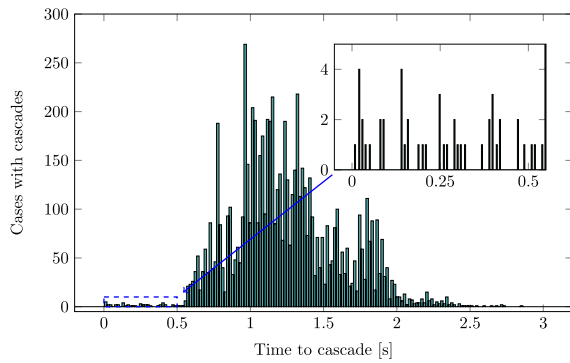


FIGURE 7. Time elapsed until first failure after fault clearance.

The dataset as resulted from these cased studies is imbalanced as cases with cascading events appear less commonly than safe cases. An imbalanced dataset can result in binary classification models that have poor predictive performance, specifically for the minority class. For this reason, a balanced dataset has been created, consisting of 7131 safe cases and 7131 cases with cascading events. The dataset is split in 12262 cases for training, 1000 cases for validation and 1000 cases for testing of the model.

V. RESULTS

A. TIME WINDOW SELECTION

In this study, a fixed length observation window approach is utilised, by training and testing the proposed model for various prediction times. In order to define this time constant, the time of the first cascading event needs to be investigated, as this defines the time window in which the prediction of whether a cascading event appears or not has to be made, i.e. before the first cascading event actually happens. Fig. 7 shows the time elapsed until the first cascading event occurs after the applied fault is cleared. After investigation, the first cascading event takes place at 0.5s-2.5s after the fault clearance in 98.8% of all cases. We observed that increasing the time window length to 0.6s leads to a significantly higher number of missed cases (98 cases - 6.34% of total cases). Hence, we exclude using time windows longer than 0.5s from our model experiments.

To investigate the impact of time window length selection on $Accuracy'$, a single layer LSTM model with 150 hidden units has been trained for 10 epochs for different time window lengths and the results are presented in Fig. 8. Also, the number of the cases with cascades for which the first cascading event appears inside this time window (referred to as missed cases) is presented in Fig. 8.c). It should be noted that these missed cases have been excluded from the training and testing dataset, as the training of the model should not include measurement data during or after cascading failures. According to the results, the time window of 0.1s leads to the highest $Accuracy'$, which is mainly driven by the lower number of missed cases (13), compared to longer time windows. For time windows longer than 0.2s, we can observe that the number of false positives increase and the number

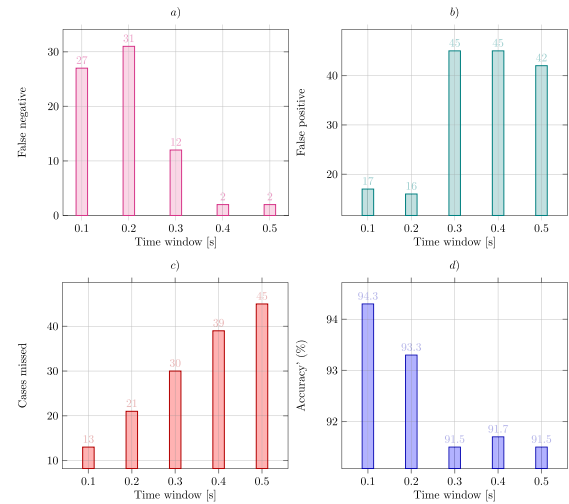


FIGURE 8. Impact of time window on online prediction.

of false negatives decrease, which results in lower precision score. Hence, the model exhibits a tendency to overfit when trained on data with window lengths of over 0.2s. It can be concluded that the time window of 0.1s allows the model to learn short-term trends and dependencies, which are more important for the predictive performance in the context of cascading events.

B. PERFORMANCE OF ONLINE PREDICTION

Following the previous analysis related to the time window selection, we use a time window of 0.1s to perform an online prediction analysis. The LSTM model performance is compared to the performance of a feed-forward MLP, as a baseline method, and a simple RNN model, which is another type of recurrent network configuration. The performance metrics of these models are presented Table 1. The metrics of the three models are calculated using the same test set of 1000 cases, which is pre-processed as described in Section II-D before being used as input to the trained models. The results show that the LSTM model exhibits the highest Accuracy Recall and F1 score. The MLP model shows a higher Precision than the LSTM model, but with a low Recall. As it is concluded, the LSTM shows overall the highest performance, and the following analysis is conducted for this model.

The confusion matrix (excluding missed cases) shown in Table 2 reveals that the trained LSTM model yields very low numbers of False Positives (17) and False Negatives (27) out of 1000 unseen data samples. While the model precision is slightly higher than recall, we find that the LSTM model has both high precision and accuracy (over 95 %) with negligible error rates.

To further investigate the cause of false predictions, the boxplots of the Y output value of the model are presented in Fig. 9. This value represents the probability of whether a case includes cascading events or not that the LSTM model provides as output. For the false positive predictions, it is observed that the values are in the range of [1, 0.5] and the

TABLE 1. Trained models and result metrics.

| | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|------|--------------|---------------|------------|--------------|
| MLP | 92.6 | 97.3 | 87.6 | 92.2 |
| RNN | 93.8 | 94.3 | 93.2 | 93.8 |
| LSTM | 95.6 | 96.5 | 94.6 | 95.5 |

TABLE 2. Confusion Matrix for the LSTM model.

| | Actually Positive (1) | Actually Negative (0) |
|------------------------|-----------------------|-----------------------|
| Predicted Positive (1) | 473 | 17 |
| Predicted Negative (0) | 27 | 483 |

TABLE 3. Cases with cascading events.

| | Number of cases | Average trips | Average Load loss (%) | SG trips | NSG trips |
|----------------|-----------------|---------------|-----------------------|----------|-----------|
| True Positive | 473 | 3.16 | 0.74 | 239 | 452 |
| False Negative | 27 | 1 | 0 | 0 | 27 |

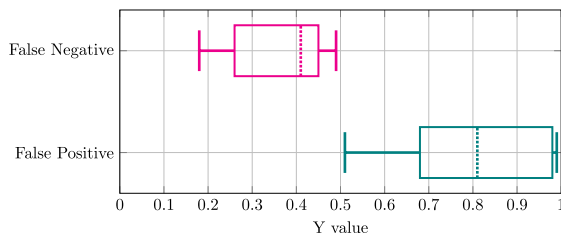


FIGURE 9. Boxplots of the model output Y for false predictions.

median value is close to 0.8. For the false positive predictions, the range of Y values and the median value (0.4) are closer to the threshold (0.5). This indicates that the model predicts falsely a safe case as a case with cascading events with higher confidence than a case with cascading events as a safe case.

The tripping of a system element may cause the appearance of subsequent events, creating cascading event sequences of varying length. A summary of the cases with cascading events is presented in Table 3 in order to identify what is the impact of the correct or incorrect predictions on system security. The cases with cascading events that the model predicts correctly (true positive), have a mean value of 3.16 trips per sequence, and a mean value of 0.74% load loss. This percentage is calculated as the amount of load that gets disconnected because of the UFLS scheme to the total amount of system load at this case. In these cases, 239 SG units trip in total. These metrics showcase that the model is able to accurately predict cases with cascading events that have a high impact on system operation. All of the actual cases with cascading events that are falsely predicted as safe cases (false negative), include only one cascading event, the tripping of wind generator NSG2 due to over-voltage. So, as in this cases only one cascading event appeared and no amount of load is shed, the false prediction does not have a high impact on system operation. However, a false model prediction about the appearance of cascading events might still provide incorrect information to system operators.

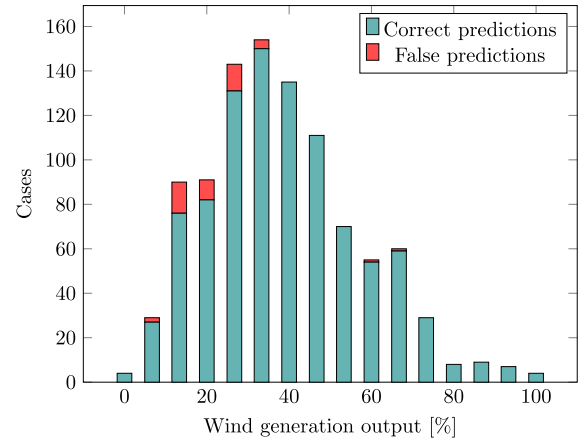


FIGURE 10. Wind generation impact on performance.

C. IMPACT OF SYSTEM LOADING AND WIND GENERATION ON PERFORMANCE

The way that initial operating conditions affect the model performance can provide useful information about machine learning applications on power systems. In Table 4 the number of cases as correct or false predictions is presented for each system loading state appearing in the test dataset. Moreover, the accuracy that the model achieved at this system loading is also shown. In the test dataset there have been no case at 90% loading, represented by XX values in the Table. It is observed that as the system loading increases and reaches the nominal value (100%) the accuracy of the model improves. For this loading value, only cases without cascading events have appeared in the dataset, and the model predicts these cases more accurately.

Following a similar approach, an investigation on how the accuracy of the model is affected by the wind generation output is also presented. The percentage of wind generation can affect the amount of synchronous generation that is disconnected and the network topology. This consequently might affect the predictive power of the model due to changes in the appearance of particular cascading events, e.g. wind generator NSG2 has been shown to cause several trips related to voltage in this particular network and cases studied. The wind generation output percentage is expressed here as the percentage of the combined output of the three wind generators to the total nominal wind generation capacity (e.g. 100% wind generation output means that in this simulation the output of the three wind generators equals their nominal capacity). When the wind generation is lower (6.7%-26.7%, bars no. 2-6 in Fig. 10 there is a higher number of false predictions (41 cases in total). For these wind generation values, the appearance of cases that include only the tripping of wind generator NSG2 due to over-voltage are common, which the model falsely predicts as safe cases as explained previously. When the wind generation is higher (40%-100%) the model achieves a very high accuracy. The analysis in this Section highlights that machine learning model performance can vary for different operating conditions of the system and

this is something that should be taken into consideration and could provide useful knowledge and potentially increased confidence when applying machine learning based methods.

TABLE 4. Impact of system loading on prediction performance.

| System loading [%] | 70 | 80 | 90 | 100 |
|---------------------|-------|-------|----|-------|
| Correct predictions | 147 | 382 | XX | 427 |
| False predictions | 9 | 22 | XX | 13 |
| Accuracy [%] | 94.23 | 94.55 | XX | 97.05 |

D. FEATURE IMPORTANCE

After training and evaluating the model, a feature importance analysis using the permutation technique as described in Section III-E is performed, in order to identify which features, in this case representing PMU measurements, are mostly affecting the model performance. Because of the nature of neural networks, each feature acquires an individual weight and affects the training of the model differently.

In Fig. 11 the 20 features that when permuted result in the largest drop in the accuracy of the model are presented. These are the features that have the highest impact on the model performance, and therefore the most important ones. All but one of the most important features correspond to PMU measurements of active (14 features) and reactive power (5 features) on lines. The most important feature, which when permuted causes a 4.8% drop in the accuracy, is the active power measurement of Line 03-04, that connects two buses in the centre of the grid on which loads are connected. When disconnected, this line changes the network topology leading to an alternative flow of power. The second most important feature is the active power measurement of Line 16-19, which when is disconnected creates an islanded part of the system and causes the frequent appearance of cascading events. The only voltage measurement included in these features is that of the wind generator NSG2 bus, the tripping of which due to over-voltage is the most common appearing cascading event.

E. CONSIDERING AVAILABILITY AND NOISE OF PMU MEASUREMENTS

In large real-life power systems, the increased number of buses makes it infeasible to install PMUs at every bus of the system. For this reason, the performance of the model when limited PMU measurements are available is investigated. The proposed LSTM model is trained and evaluated using only the 10, 15 and 20 most influential features, as these have resulted from the feature importance analysis. The results in Table 5 show that when 10 and 15 features are considered, the model performs with 84% and 90.9% accuracy respectively (12.1% and 4.9% reduction in accuracy compared to the original LSTM model with 178 features). When the number of features is increased to 20, all of the model performance metrics improve, performing with 94.4% accuracy (1.26% reduction in accuracy). For this particular study it can be concluded that the model performance is satisfactory when including only the 20 most influential features. These 20 features can provide

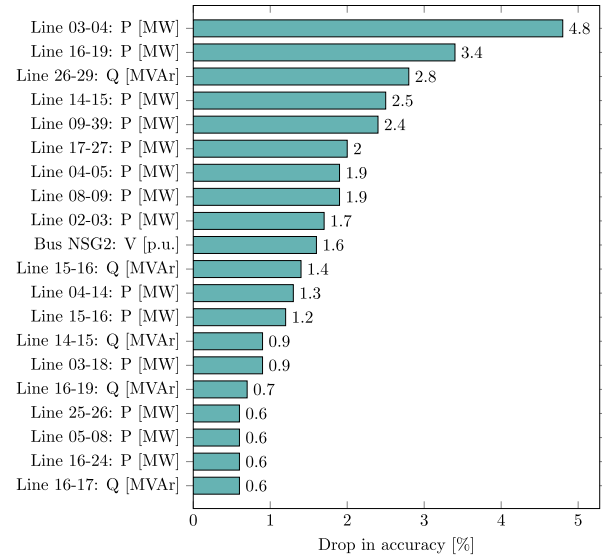


FIGURE 11. Permutation feature importance.

locational information about the buses at which the PMUs should be installed.

In certain cases, communication issues with PMUs might disrupt the transmission of information during the online phase. Most machine learning models, including the proposed LSTM based model, can not handle missing features, so in this case the respective column of the missing feature is filled with 0's. When the most important feature, the active power measurement of Line 03-04, is missing then the accuracy of the model drops at 90.4% (5.44% reduction compared to no missing features). However, when a less important feature is missing, e.g. the active power measurement of Line 06-11, then the accuracy of the model is 95.2% (0.42% reduction). It should be noted that the feature importance analysis presented as part of our method can inform the introduction of measures to counteract issues with missing data by reinforcing the important communication channels, e.g. through redundancy.

According to [38], the PMU signals include an indicator code about the time quality of the signal, which indicates the maximum network delay. In cases of network delays during the online application of the model, this code can provide information to system operators about any potential delay in the measurement data and also indicate if the measurement time is not reliable.

In practical applications, the PMU measurements may contain noise introduced by errors related to transducers and signal processing. The pre-trained LSTM model with 178 features is tested using test data measurements with added noise signal. The noise in PMU measurements is simulated by Additive white Gaussian noise (AWGN) with a standard deviation of 0.002 p.u. [39]. The results show that the added noise has no effect on the model performance, as the performance metrics, in Table 5, are identical to those of the original LSTM model without added noise. This highlights the robustness of

TABLE 5. Performance considering limited availability and noise of PMU measurements.

| LSTM | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|--------------|--------------|---------------|------------|--------------|
| /10 features | 84 | 92.1 | 74.4 | 82.3 |
| /15 features | 90.9 | 90.2 | 91.8 | 90.9 |
| /20 features | 94.4 | 96.1 | 92.6 | 94.3 |
| /w noise | 95.6 | 96.5 | 94.6 | 95.5 |

the proposed method, as PMU measurements with noise do not affect its performance.

F. MODEL APPLICATION ON A DIFFERENT SYSTEM TOPOLOGY

Due to the increasing renewable penetration, new RES units may connect to the grid, changing the network topology. In such cases, new data might not be already available for re-training the model. To simulate such a situation, a new system topology is considered, by adding a new wind farm (NSG4) on Bus 26 of the modified IEEE-39 bus model (Fig. 6). The aim is to assess the performance of the pre-trained LSTM model on this version of the network. In a similar manner, after defining the initial operation conditions, in this case the power output of the 4 wind farms and system loading, 1000 dynamic RMS simulations are performed. The performance of the LSTM model for the prediction of appearance of cascading events in this test dataset is presented in Table 6.

For this network, the LSTM model performs with a 94.9% overall accuracy. Compared to the performance of the LSTM model for the original network (Table 1), the accuracy of the model decreases by 0.73%. When more data from this network topology become available, the model weights could be updated by fine-tuning the pre-trained model on the new data.

TABLE 6. Pre-trained LSTM model performance on a different system topology.

| | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|------|--------------|---------------|------------|--------------|
| LSTM | 94.9 | 80.3 | 95.6 | 87.3 |

G. COMPUTATIONAL TIME, PRACTICAL CONSIDERATIONS AND SCALABILITY

The simulations have been performed using the DIgSILENT Powerfactory RMS solver with the adaptive time step option enabled. The approximate averaged running time of one simulation without cascading events is 22s, and the running time of a simulation with cascading events is 86s. The interface between Python and DIgSILENT Powerfactory has been used to set up the dynamic simulations running multiple simulated cases in parallel in order to speed up the process of generating the described dataset. The LSTM models are trained on a Nvidia Quadro RTX 6000 GPU, and the average training time is 538s. These processes take place during the offline stage, where more time is available.

During the online stage, the average time that a single inference takes after performing 1000 predictions on the GPU using the pre-trained model is 0.042s, which highlights the fitness of the method for real-time prediction as a fast model response at this stage is critical. The maximum time that a single inference takes is 0.049s, which allows for a prediction before the appearance of the first cascading event for every case in the test dataset, for this particular test system and study cases. This showcases the ability of machine learning estimators to respond significantly faster, compared to the running time of a time-domain simulation.

In a practical application, the dataset used in this study would comprise of measurements gathered approximately over the span of a year. As new operating conditions emerge, and new measurements become available the pre-trained model can be updated using the new data. In this study, the time required to update the model with 1000 unseen cases is 9.88s. So, in a practical application the pre-trained model can be updated over shorter periods of time e.g. every month, and be subsequently used for the online prediction.

The main challenge with scaling the proposed framework for a larger network, would be the computational time required during the offline stage. To address this, the number of simulations that are performed in parallel could be adjusted accordingly. In addition, due to the increased number of parameters an importance sampling [40] or efficient sampling [41] technique can be deployed to define the simulation cases. Regarding the model application, in a larger network there would be more PMU measurements available, therefore more input features for the training and inference of the model. Depending on the number of features, feature engineering steps might be required to reduce the number of features (e.g. removing correlated features). However, in [8] and [14] LSTM-based models have been applied on large network models, showcasing high performance and fast inference times.

VI. CONCLUSION

This paper introduces a framework for the online identification of cascading events in power systems with renewable generation using measurement data and supervised machine learning, namely LSTMs, a type of RNN. Dynamic RMS simulations on a model with protection devices included have been performed, in order to capture cascading events that appear, which are defined by the action of the protection devices. Simulation data are pre-processed to represent typical PMU data and are used to train a LSTM based model. The pre-trained model is then used to predict online the appearance of cascading events and various aspects of its performance are analysed, including the time window selection, important features and how the performance is affected for different operating conditions. The framework is applied on a modified version of the IEEE-39 bus system, including wind generators and protection devices.

Results show that the proposed approach performs with a 95.6% accuracy within a short fixed-time window

(in the order of 0.1s) following the initial fault clearance, showing improved performance compared to other neural network configurations (MLP and RNN). The model has the ability to predict the appearance of cascading events sequences, as opposed to only early instability violations that is a common approach in existing online prediction methods. After further investigation, the performance of the method appears to vary with the initial operating conditions, either improving or deteriorating. Such behaviour should be taken into account in order to inform the confidence to similar methods when considering real power system applications. Finally, the results of the feature importance analysis highlight important system variables that improve the model performance, with offering useful information in terms of monitoring requirements as well as system variables that are related to the appearance of cascading events. For this particular network, active and reactive power measurements of lines have a high impact on the prediction of cascading events. Also, the measurement of the electrical value that causes the most common cascading event is identified as an important feature. Tests considering limited available PMU measurements and noise in signals have little impact on model performance, verifying that the suggested approach is appropriate for practical applications.

ACKNOWLEDGMENT

The authors would like to extend their gratitude to the Alan Turing Institute for the support and facilitation of the September 2020 Data Study Group challenge “Using machine learning to predict the onset of blackouts” and all the participants and the challenge PI: Domenic Di Francesco, Alara Dirik, Sofia Koukoura, Amarsagar Reddy Ramapuram Matavalam, Maryleen Ndubuaku, Josh Nevin, Oliver Paul, and Chloe Sekkat. All results can be fully reproduced using the methods and data described in this article and references provided.

REFERENCES

- [1] N. Hatziaargyriou, J. Milanovic, C. Rahmann, V. Ajarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. Sanchez-Gasca, A. Stankovic, T. Van Cutsem, V. Vittal, and C. Vournas, “Definition and classification of power system stability—Revisited & extended,” *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3271–3281, Jul. 2021.
- [2] J. Bialek, E. Ciapessoni, D. Cirio, E. Cotilla-Sanchez, C. Dent, I. Dobson, P. Henneaux, P. Hines, J. Jardim, S. Miller, M. Panteli, M. Papic, A. Pitto, J. Quiros-Tortos, and D. Wu, “Benchmarking and validation of cascading failure analysis tools,” *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4887–4900, Nov. 2016.
- [3] M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller, and P. Zhang, “Risk assessment of cascading outages: Methodologies and challenges,” *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 631–641, May 2012.
- [4] M. H. Vasconcelos, L. M. Carvalho, J. Meirinhos, N. Omont, P. Gambier-Morel, G. Jamgotchian, D. Cirio, E. Ciapessoni, A. Pitto, I. Konstantelos, G. Strbac, M. Ferraro, and C. Biasuzzi, “Online security assessment with load and renewable generation uncertainty: The iTesla project approach,” in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst. (PMAPS)*, Oct. 2016, pp. 1–8.
- [5] I. Konstantelos, G. Jamgotchian, S. H. Tindemans, P. Duchesne, S. Cole, C. Merckx, G. Strbac, and P. Panciatici, “Implementation of a massively parallel dynamic security assessment platform for large-scale grids,” *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1417–1426, May 2017.
- [6] P. N. Papadopoulos, T. Guo, and J. V. Milanovic, “Probabilistic framework for online identification of dynamic behavior of power systems with renewable generation,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 45–54, Jan. 2018.
- [7] L. Zheng, W. Hu, Y. Zhou, Y. Min, X. Xu, C. Wang, and R. Yu, “Deep belief network based nonlinear representation learning for transient stability assessment,” in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2017, pp. 1–5.
- [8] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li, “Intelligent time-adaptive transient stability assessment system,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018.
- [9] A. Irvani and F. de León, “Real-time transient stability assessment using dynamic equivalents and nonlinear observers,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2981–2992, Jul. 2020.
- [10] T. L. Vu and K. Turitsyn, “Lyapunov functions family approach to transient stability assessment,” *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1269–1277, Mar. 2016.
- [11] P. Ju, H. Li, C. Gan, Y. Liu, Y. Yu, and Y. Liu, “Analytical assessment for transient stability under stochastic continuous disturbances,” *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2004–2014, Mar. 2018.
- [12] Z. Shi, W. Yao, L. Zeng, J. Wen, J. Fang, X. Ai, and J. Wen, “Convolutional neural network-based power system transient stability assessment and instability mode prediction,” *Appl. Energy*, vol. 263, Apr. 2020, Art. no. 114586.
- [13] J. H. Arteaga, F. Hancharou, F. Thams, and S. Chatzivasileiadis, “Deep learning for power system security assessment,” in *Proc. IEEE Milan PowerTech*, Jun. 2019, pp. 1–6.
- [14] L. Zhu, D. J. Hill, and C. Lu, “Intelligent short-term voltage stability assessment via spatial attention rectified RNN learning,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7005–7016, Oct. 2021.
- [15] J. D. Pinzón and D. G. Colomé, “Real-time multi-state classification of short-term voltage stability based on multivariate time series machine learning,” *Int. J. Electr. Power Energy Syst.*, vol. 108, pp. 402–414, Jun. 2019.
- [16] H. Yang, W. Zhang, J. Chen, and L. Wang, “PMU-based voltage stability prediction using least square support vector machine with online learning,” *Electric Power Syst. Res.*, vol. 160, pp. 234–242, Jul. 2018.
- [17] R. A. Shuvro, P. Das, M. M. Hayat, and M. Talukder, “Predicting cascading failures in power grids using machine learning algorithms,” in *Proc. North Amer. Power Symp. (NAPS)*, Oct. 2019, pp. 1–6.
- [18] X. Wu, D. Wu, and E. Modiano, “Predicting failure cascades in large scale power systems via the influence model framework,” *IEEE Trans. Power Syst.*, vol. 36, no. 5, pp. 4778–4790, Sep. 2021.
- [19] S. Gupta, R. Kambli, S. Wagh, and F. Kazi, “Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2478–2486, Apr. 2015.
- [20] P. Geurts and L. Wehenkel, “Early prediction of electric power system blackouts by temporal machine learning,” in *Proc. ICML-AAAI*, 1998, pp. 21–27.
- [21] T. Ahmad and P. N. Papadopoulos, “Prediction of cascading failures and simultaneous learning of functional connectivity in power system,” in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Oct. 2022, pp. 1–5.
- [22] J. Song, E. Cotilla-Sanchez, G. Ghanavati, and P. D. H. Hines, “Dynamic modeling of cascading failure in power systems,” *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2085–2095, May 2016.
- [23] E. Ciapessoni, D. Cirio, and A. Pitto, “Cascading in large power systems: Benchmarking static vs. time domain simulation,” in *Proc. IEEE PES Gen. Meeting Conf. Expo.*, Jul. 2014, pp. 1–5.
- [24] P. Anderson, A. Fouad, and H. Happ, “Power system control and stability,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 2, p. 103, Feb. 1979.
- [25] Y. Xiang, T. Wang, and Z. Wang, “Protection information based transient stability margin assessment of splitting islands,” *Int. J. Electr. Power Energy Syst.*, vol. 130, Sep. 2021, Art. no. 107007.
- [26] M. Noebels, I. Dobson, and M. Panteli, “Observed acceleration of cascading outages,” *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3821–3824, Jul. 2021.

- [27] P. N. Papadopoulos, T. Guo, X. Wang, and J. V. Milanovic, "Impact of measurement signals on the accuracy of online identification of power system dynamic signature," in *Proc. IEEE Eindhoven PowerTech*, Jun. 2015, pp. 1–6.
- [28] G. A. Nakas and P. N. Papadopoulos, "Investigation of cascading events in power systems with renewable generation," in *Proc. IEEE PES Innov. Smart Grid Technol. Eur. (ISGT-Europe)*, Oct. 2020, pp. 211–216.
- [29] T. Van Cutsem, M. Glavic, W. Rosehart, C. Canizares, M. Kanatas, L. Lima, F. Milano, L. Papangelis, R. A. Ramos, J. A. D. Santos, B. Tamimi, G. Taranto, and C. Vournas, "Test systems for voltage stability studies," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 4078–4087, Sep. 2020.
- [30] C. M. Ferreira, F. P. M. Barbosa, and C. I. F. Agreira, "Transient stability preventive control of an electric power system using a hybrid method," in *Proc. 12th Int. Middle-East Power Syst. Conf.*, 2008, pp. 141–145.
- [31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [32] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 37–45.
- [33] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, Sep. 2014, pp. 338–342.
- [34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.
- [35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [36] *The Grid Code, National Grid Electricity System Operator Limited*, Nat. Grid Electr. Syst. Operator, U.K., 2020.
- [37] G. A. Nakas and P. N. Papadopoulos, "Investigation of the impact of load tap changers and automatic generation control on cascading events," in *Proc. IEEE Madrid PowerTech*, Jun. 2021, pp. 1–6.
- [38] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, IEEE Standard C37.118.2-2011 (Revision of IEEE Standard C37.118-2005), 2011, pp. 1–53.
- [39] D. Shi, D. J. Tylavsky, and N. Logic, "An adaptive method for detection and correction of errors in PMU measurements," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1575–1583, Dec. 2012.
- [40] R. Preece and J. V. Milanovic, "Efficient estimation of the probability of small-disturbance instability of large uncertain power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1063–1072, Mar. 2016.
- [41] V. Krishnan, J. D. McCalley, S. Henry, and S. Issad, "Efficient database generation for decision tree based power system security assessment," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2319–2327, Nov. 2011.



PANAGIOTIS N. PAPAPOPOULOS (Member, IEEE) received the Dipl.-Eng. and Ph.D. degrees from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, in 2007 and 2014, respectively. He is currently a Senior Lecturer and a U.K. Research and Innovation (UKRI) Future Leaders Fellow with the Department of Electronic and Electrical Engineering, University of Strathclyde. His research interests include power system stability and dynamics, focusing on tackling the increasing uncertainty and complexity in the system dynamic behavior.



AMARSAGAR REDDY RAMAPURAM MATAVALAM (Member, IEEE) received the B.Tech. degree in electrical engineering and the M.Tech. degree in power electronics and power systems from IIT Madras, Chennai, India, and the Ph.D. degree in electrical engineering from Iowa State University, Ames, IA, USA. He is currently an Assistant Professor with Arizona State University, Tempe, AZ, USA. His research interests include classical and data-driven power system stability analysis and control, reinforcement learning for enhancing power grid operations, and characterizing uncertainty in power systems.



OLIVER PAUL was a Researcher with the Idiap Research Institute, where he was involved in surrogate deep learning models to solve optimization problems in district heating and building management systems. He is currently the Co-Founder of CheckSolar, a roboadviser platform for prosumers looking to get into the domestic solar PV market in Germany. He has diverse experience in engineering and data science across the energy space, having worked in energy teams at ABB, The World Economic Forum, Schlumberger, and various startups.



GEORGIOS A. NAKAS (Graduate Student Member, IEEE) received the Dipl.-Eng. degree from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K. His research interests include modeling of electric power systems and machine learning applications on power system dynamics.



ALARA DIRIK is currently pursuing the Ph.D. degree with Boğaziçi University, Istanbul, Turkey. She is a Machine Learning Engineer with Hugging Face. Her research interests include deep learning, generative modeling, and 3D vision.



DIMITRIOS TZELEPIS (Member, IEEE) is currently a Visiting Lecturer with the University of Strathclyde and the CTO and the Co-Founder of SMPnet. His axiomatic passion for the energy transition, drives him to establish intelligent solutions, to promote secure, reliable, and fully-automated operation of software-defined electricity grids. He has led research and commercial multi-million projects, has developed patented solutions for the power sector has contributed to the preparation and drafting of grid codes for the U.K. and Ireland. His research interests include power system control, protection, and automation, incorporating increased penetration of renewable energy sources, and high-voltage direct current interconnections. During his research and professional activities, he has acquired profound knowledge of the integration of international standards and protocols relevant to the development of commercial and non-commercial automation, protection, and control systems.

He is occupying strategic roles in several technical committees to decarbonize the energy sector, including CIGRE; Global Smart Grids Innovation Hub, Spain; and Research and Development Committees to draft and advance Grid Codes.

...