# Online Identification of Cascading Events using Measurement Data and Machine Learning


| | |
|---|---|
| Journal: | *IEEE Transactions on Industrial Informatics* |
| Manuscript ID | TII-22-0146 |
| Manuscript Type: | Regular Paper |
| Keywords: | power system dynamics, power system cascading failures, machine learning, phasor measurement units applications, reccurent neural networks |
| | |

# Online Identification of Cascading Events using Measurement Data and Machine Learning

***Abstract*—This paper introduces a framework for online identification of cascading events in power systems, based on measurements. Cascading events involve highly complex dynamic phenomena and can compromise the security of power systems, leading even to large scale blackouts. The framework is based on supervised machine learning (Long short term memory recurrent neural networks), considering uncertainties associated with network operating conditions, contingencies and renewable generation. By utilizing measurements, the proposed method can predict the onset 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 renewable generation and protection devices. Results highlight that the suggested method can successfully identify cases with cascading events with up to 95.6% accuracy. Practical aspects of the method application and performance related to measurements and their availability are also investigated.**

***Index Terms*—cascading failures, dynamic simulation, machine learning, phasor measurement units, power systems, 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. 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 secure operation.

The accurate representation of protection devices, which in some cases might activate before instability limits are reached, is a key element in capturing the cascading events that might appear after a contingency [2], [3]. 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.

In [4], a machine learning based method for online static and dynamic security assessment is described, that can provide possible control actions to avoid insecure operation. The offline phase of this platform is presented in [5], where a large amount of dynamic simulations, taking into account load and renewable generation related uncertainty, is performed and decision trees are used to extract security rules.

Various machine learning techniques have been applied to address 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 and investigates the impact of topology. In [7], the original network data is transformed into an abstract representation state using a deep belief network, considering the power system topology. 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. The proposed 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 a dynamic equivalent model, reducing the model order and complexity. 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, to train a convolutional neural network (CNN). The results show higher performance of CNNs compared to conventional machine learning classifiers. A similar use of bitmaps and CNNs, applied to small-signal stability is introduced in [13].

A few methods have also addressed online voltage stability. The online short term voltage stability (SVS) assessment is addressed in [14] using an LSTMs. A random-forest-based method is presented in [15] for online voltage stability identification. In [16], a PMU-based method is proposed to predict SVS, combining support vector machine and online learning.

The above mentioned methods, mainly aim at identifying specific stability or security related phenomena and do not investigate the possibility of cascading events. 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], using DC power flow simulations. In [18] an influence model based on a hybrid learning method is used for predicting cascading failures from simulations carried out on a DC/AC power flow model. A probabilistic approach for predicting cascading events using support vector machine and static simulations is presented in [19]. An early method for the prediction of cascading events is described in [20], using temporal trees to predict the appearance of events only related to voltage collapse phenomena. As results from [21] have shown, dynamic simulations can capture cascading events in more detail than static, as some of the events during the later stages of 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, [22], [23] highlight that not including protection devices in dynamic studies might result in inaccurate assessment of system behaviour.

The main contribution of this study is the use of supervised machine learning and measurement data from dynamic simulations to predict the appearance of cascading events, as defined by the actions of protection devices. The accurate modelling of system dynamics (capturing phenomena related to voltage, frequency and transient instability) as well as the discrete action of protection devices are considered, which play a significant role in the realistic representation of the appearance of cascading events. This approach for online identification of cascading events including the action of protection devices 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. The method is applied on a power system with renewable generation, as the uncertainty that comes with it has shown to affect the power system operation after a contingency. 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. Additional contributions of the proposed method related to practical aspects of its application and performance in realistic settings that can offer useful insights are the following: i) the investigation of the impact that the time window length, used for the online prediction, has on the model performance. ii) How individual operating conditions affect the performance of the prediction model. 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 PMUs 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 method online and offline stages, as described below, is presented in Fig. 1.a)

### A. Detailed Procedure

The method presented in this paper consists of two main stages: i) the offline generation of the dataset and the training 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 in this study is defined as the trip of a component caused by the activation of protection device after (and in addition to) the initial disturbance. 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, 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).

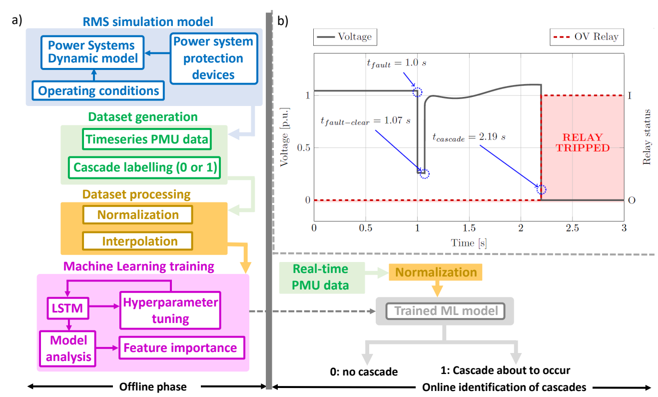


Fig. 1. a) Flowchart illustrating the steps of the proposed framework. b) Example of a cascading event and the method application.

In practical applications, the time domain measurement data during the online phase can be obtained from PMU measurements 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. 1.b). In this plot, the cause of a cascading event, the tripping of a wind generator (NSG2) due to over-voltage, and the signal of the protection relay are presented. After the fault clearance (1.07s), the violation of the over-voltage protection limits (in this case 1.1 p.u. for over 0.15s) 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 more trips 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 before 2.19s).

### B. Dataset Generation

In this study, the sampling of possible initial operating conditions is based on the discretization of their operating range, towards creating a large dataset of case studies. The parameters considered for this purpose are RES output generation (for each RES unit), system loading and line fault location.

After the sampling of the initial operating conditions an AC OPF problem is solved to determine the dispatch of the SGs. Each SG is allocated a cost curve, 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. 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 RES penetration. 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=1s and gets cleared by disconnecting the faulted line after 70ms. If the network conditions following the contingency cause the activation of the protection devices 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.

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 PMUs 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.

### C. 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 ensure that they are on the same scale, a common method for efficient machine learning applications. 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].

We then perform interpolation to ensure evenly sampled time steps across all simulations. We use first order spline interpolation and set the time interval $\delta$ 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. 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: $h_t^l = f\left(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l\right)$ , 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 and $x_t$ is the input at time step $t$. Furthermore, $f$ denotes the activation function where $f \in \{sigmoid, \tanh\}$.

However, RNNs suffer from a significant problem: vanishing or exploding gradients. To train a neural network, the total loss is backpropagated 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.

The LSTM networks are a kind of RNN 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 $c_t^l \in \mathbb{R}^n$.

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

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

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{2}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g\left(W_{cx}x_t + W_{cm}m_{t-1} + b_c\right) \tag{3}$$

$$o_t = \sigma\left(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o\right) \tag{4}$$

$$m_t = o_t \odot h\left(c_t\right) \tag{5}$$

$$y_t = \phi\left(W_{ym}m_t + b_y\right) \tag{6}$$

where the $W$ denote weight matrices, $b$ denote bias vectors, $\sigma$ 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 functions, $tanh$ and $\phi$ are the hyperbolic tangent and softmax activation functions respectively.

### B. Using LSTMs to predict cascading events

LSTMs offer a good fit for the problem of predicting cascading events, as they 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 evolution in time. The LSTM model is trained using the pre-processed data, described in Sections II-B and II-C, 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.

To pose the occurrence of a cascades 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).

### C. Model Training

To train the LSTM models, we use the pre-processed data as outlined in Section V-F 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.

Batch gradient descent is used to perform back propagation and parameter updates over batches of input data, as this method 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. 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 (7-10) are used. Accuracy, Precision, Recall and F1 score are typical measures used in machine learning that capture different aspects of the performance of a binary classifier.

$$Accuracy\ (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN}} \tag{7}$$

$$Precision\ (\%) = \frac{n_{TP}}{n_{TP} + n_{FP}} \tag{8}$$

$$Recall\ (\%) = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{9}$$

$$F1\ Score\ (\%) = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

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. 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:

$$Accuracy'\ (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN} + n_{MC}} \tag{11}$$

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 important 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 expect ordered time-series as input, a permutation of an important feature would cause a drop in accuracy. 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 is used, by performing dynamic RMS simulations in DIgSILENT PowerFactory. The ten synchronous generators (SGs) in the network are represented by full detail four winding models (6th-order), equipped with Automatic Voltage Regulator (AVR), Power System Stabilizer (PSS), and Governor (GOV). More details about the system parameters can be found in [24]. The control of the synchronous generators has been augmented with the action of OELs with inverse time characteristic. The original network model has been modified with the addition of three wind farms that are connected to three different network buses and are implemented using International Electrotechnical Commission (IEC) Type 4A wind turbines. 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 following protection devices are implemented in the network to capture the dynamic phenomena related to voltage, frequency and transient stability. 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. The relays settings comply with the transmission system limits from the UK grid code. More details about the protection devices settings can be found in [25].

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 [26]. The duration of the RMS simulations has been set to 120s to capture both fast and slower evolving dynamic phenomena. More information on the modelling procedure can also be found in [27].

### 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 loading values and a 20% step for sampling the output of each wind generator. This approach ensures that equally divided areas of the search space are considered. 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). In a larger network model, due to the increased number of parameters an importance sampling [28] or efficient sampling [29] can be deployed to define the simulation cases.

The percentage of cases with cascading events is higher compared to practical applications, as the lines that are disconnected in reality could be comprised of double circuits. So, each initial contingency represents potentially the disconnection of two parallel circuits at a time, which in some cases causes an area of the system to become islanded, leading to the appearance of cascading events. 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. The time of the first cascading event 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. 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.

TABLE I
IMPACT OF TIME WINDOW ON ONLINE PREDICTION.

| Time window [s] | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|
| False negative | 27 | 31 | 12 | 2 | 2 |
| False positive | 17 | 16 | 45 | 45 | 42 |
| Cases missed | 13 | 21 | 30 | 39 | 45 |
| Accuracy' [%] | 94.3 | 93.3 | 91.5 | 91.7 | 91.5 |

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 Table I. 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. 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. The results show that a time window of 0.1s results in the best $accuracy'$, as driven by the low number of missed cases (13). Moreover, we observe that the number of false positives increase and the number of false negatives decrease as the time window length increases, leading to a loss in precision. Hence, the model exhibits a tendency to overfit when trained on data with window lengths of over 0.2s.

## 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. The same test set of 1000 cases is used for the three models, which is pre-processed as described in Section II-C before being used as input to the trained models, and the performance metrics are presented in Table II. 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 shown in Table III 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 %).

To further investigate the cause of false predictions, boxplots of the $Y$ output value are presented in Fig. 2. 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 [0.5,1] and the 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.

TABLE II
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 III
CONFUSION MATRIX FOR THE LSTM MODEL.

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

TABLE IV
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 |

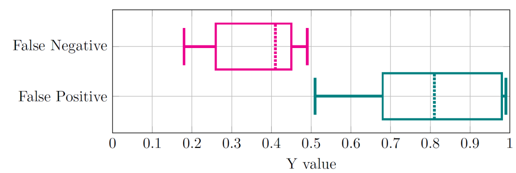


Fig. 2. Boxplots of the model output Y for false predictions.

A summary of the cases with cascading events is presented in Table IV 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.

## 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 V the number of cases as correct or false predictions and the accuracy that the model achieved for each system loading state appearing in the test dataset are presented (there have been no cases at 90% loading, represented by $XX$ values). 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, how the accuracy of the model is affected by the wind generation output is also investigated. The percentage of wind generation can affect the amount of synchronous generation disconnection and the network topology.

This consequently might affect the model performance 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 case 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. 3.a) 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. These results highlight that the model performance can vary for different operating conditions. This behaviour should be considered and could provide useful knowledge and potentially increased confidence when applying machine learning based methods.

TABLE V
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 |

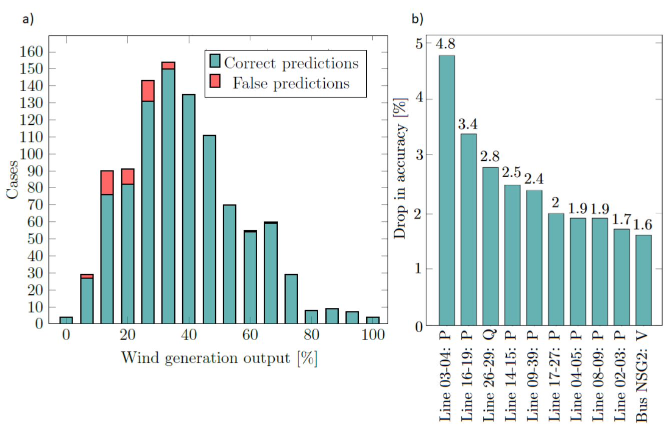


Fig. 3. a) Wind generation impact on performance. b) Permutation feature importance.

## D. Feature Importance

After training and evaluating the model, a permutation feature importance analysis 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. 3.b) the 10 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 (8 features) and reactive power (1 feature) 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 PMUs

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

TABLE VI
PERFORMANCE WITH LIMITED AVAILABILITY AND NOISE OF PMUS.

| 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 pre-trained LSTM model with 178 features is tested using test data measurements with added noise signal, which in practical applications can be by errors related to transducers and signal processing. The noise in PMU measurements is simulated by Additive white Gaussian noise (AWGN) with a standard deviation of 0.002 p.u. [30]. The results show that the added noise has no effect on the model performance, as the performance metrics, in Table VI, are identical to those of the original LSTM model without added noise. This highlights the robustness of the proposed method, as PMU measurements with noise do not affect its performance.

## F. Computational time and Practical considerations

The RMS simulations have been performed using DIgSILENT Powerfactory with adaptive time step. The approximate averaged running time of one simulation without cascading events is 22s and 86s when cascading events occur. 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, when more time is available. During the online stage, the average time that a single prediction 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 and is 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, the pre-trained model can be updated using the new measurement 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.

## 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 have been performed, in order to capture cascading events that appear, which are defined by the action of the protection devices implemented in the model. Simulation data are pre-processed to represent typical PMU data and are used to train a LSTM model. The pre-trained model is then used to predict online the appearance of cascading events. Various aspects related to the choice of prediction window, availability and importance of measurements and noise that could impact performance in a realistic setting, are also analysed. 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 MLP and RNN configurations. The model can predict the appearance of cascading events sequences, as opposed to only early instability violations, different to common approaches 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 considered to inform the confidence in similar methods for real life power system applications. The feature importance analysis highlights important system variables for the model performance, offering useful information about monitoring requirements and system variables that are related to the appearance of cascading events. For this particular network, active/reactive power measurements of lines and measurements related to frequently appearing protection device trips (e.g. voltage of a wind farm) have a high impact on the prediction of cascading events. 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.

## References


[1] N. Hatziargyriou *et al.*, "Definition and classification of power system stability – revisited & extended," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3271–3281, 2021.

[2] J. Bialek *et al.*, "Benchmarking and Validation of Cascading Failure Analysis Tools," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4887–4900, 2016.

[3] M. Vaiman *et al.*, "Risk assessment of cascading outages: Methodologies and challenges," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 631–641, 2012.

[4] M. H. Vasconcelos *et al.*, "Online security assessment with load and renewable generation uncertainty: The itesla project approach," in *2016 PMAPS*, 2016, pp. 1–8.

[5] I. Konstantelos *et al.*, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1417–1426, 2017.

[6] P. N. Papadopoulos *et al.*, "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, 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 *2017 IEEE PES GM*, 2017, pp. 1–5.

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

[9] A. Iravani 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, 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, 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, 2018.

[12] L. Zheng *et al.*, "Deep belief network based nonlinear representation learning for transient stability assessment," in *2017 IEEE PES GM*, 2017, pp. 1–5.

[13] J.-M. H. Arteaga *et al.*, "Deep learning for power system security assessment," in *2019 IEEE PowerTech*, 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 Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7005–7016, 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. Journ. of El. Power & Energy Systems*, vol. 108, pp. 402 – 414, 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 Systems Research*, vol. 160, pp. 234 – 242, 2018.

[17] R. A. Shuvro, P. Das, M. M. Hayat, and M. Talukder, "Predicting cascading failures in power grids using machine learning algorithms," in *2019 North American Power Symposium (NAPS)*, 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 Transactions on Power Systems*, vol. 36, no. 5, pp. 4778–4790, 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. Industrial Electronics*, vol. 62, no. 4, pp. 2478–2486, 2015.

[20] P. Geurts and L. Wehenkel, "Early prediction of electric power system blackouts by temporal machine learning," in *Proceedings of ICML-AAAI 98 Workshop*, 1998, pp. 21–27.

[21] J. Song, E. Cotilla-Sanchez, G. Ghanavati, and P. D. Hines, "Dynamic modeling of cascading failure in power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2085–2095, 2016.

[22] P. Anderson, A. Fouad, and H. Happ, "Power system control and stability," *IEEE Trans. Syst., Man, and Cybernetics*, vol. 9, no. 2, 1979.

[23] Y. Xiang, T. Wang, and Z. Wang, "Protection information based transient stability margin assessment of splitting islands," *International Journal of Electrical Power  Energy Systems*, vol. 130, 2021.

[24] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Trans. on Power App. and Syst.*, vol. PAS-98, no. 2, pp. 573–584, 1979.

[25] G. A. Nakas and P. N. Papadopoulos, "Investigation of cascading events in power systems with renewable generation," in *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2020, pp. 211–216.

[26] T. Van Cutsem *et al.*, "Test systems for voltage stability studies," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 4078–4087, 2020.

[27] G. A. Nakas and P. N. Papadopoulos, "Investigation of the impact of load tap changers and automatic generation control on cascading events," in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.

[28] R. Preece and J. V. Milanović, "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, 2016.

[29] 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, 2011.

[30] D. Shi, D. J. Tylavsky, and N. Logic, "An adaptive method for detection and correction of errors in pmu measurements," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1575–1583, 2012.