# Power system frequency monitoring and emergency control with neural ordinary differential equations

Karacelebi, Mert; Cremer, Jochen L.

**DOI**
10.1016/j.segan.2025.101815

**Publication date**
2025

**Document Version**
Final published version

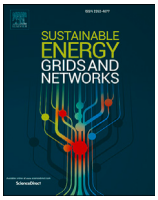
**Published in**
Sustainable Energy, Grids and Networks

Sustainable Energy, Grids and Networks 43 (2025) 101815

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan



# Power system frequency monitoring and emergency control with neural ordinary differential equations

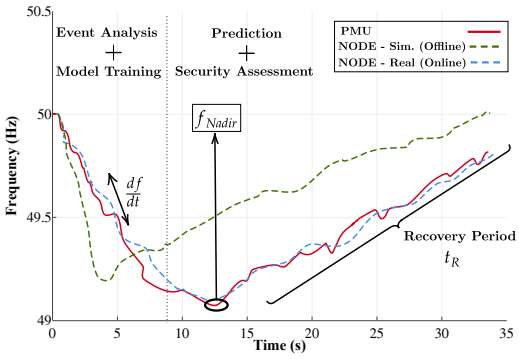
Mert Karacelebi*, Jochen L. Cremer

*Department of Electrical Sustainable Energy, Delft University of Technology, Mekelweg 4, Delft, 2628CD, the Netherlands*

## HIGHLIGHTS

- Power system frequency prediction with online training of Neural Ordinary Differential Equations (NODE) using only local measurements.
- Irregular weighted downsampling of training data to improve the efficiency of the online training.
- Prediction-based trigger function for early activation of emergency corrective control actions.
- Testing NODE predictions against noisy PMU data from the Dutch high voltage transmission system.

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Keywords:*
Neural ordinary differential equations
Frequency stability
Collocation method
Irregular sampling
Emergency control

## ABSTRACT

Increasing renewable energy supply and distributed generating sources in the power grid lead to lower inertia levels. Lower inertia combined with higher uncertainty in operation can cause drastic frequency fluctuations when a disturbance occurs. System operators must know whether the transmission system is secure against a disturbance. Real-time models attempt to predict the frequency in near-real time however require pretraining on a large variety of possible disturbances. Training models in real-time would not require pre-training as they are directly trained on the occurring disturbance. However, training in real-time is not feasible until now as fast-occurring system dynamics require shorter prediction (and training) times for security and operation which standard machine learning models are not capable of.

For the first time, this work proposes a fast training strategy that learns Neural Ordinary Differential Equations (NODE) in near real-time directly on the occurring disturbance, simultaneously addressing the inaccuracy issue of model-based dynamic studies. NODE learns the dynamics or derivatives of an ODE system that standard ODE solvers can solve. NODE provides a continuous function for predicting future dynamics in a decentralized way, hence faster frequency stability assessment for longer time spans. We propose a collocation-based sampling using the collocation gradients. Case studies on the IEEE 39-bus system show the approach is feasible for near real-time operation, accurately predicts future system states, and enables operators to apply predesigned corrective control actions, potentially making the system secure for future disturbances.

* Corresponding author.
*Email address:* m.karacelebi@tudelft.nl (M. Karacelebi).

https://doi.org/10.1016/j.segan.2025.101815
Received 11 December 2024; Received in revised form 25 March 2025; Accepted 23 July 2025
Available online 25 July 2025

## 1. Introduction

The transition towards variable renewable energy resources (VRES) significantly affects the power system frequency stability and the security of the operation. The system inertia slows down the change of frequency caused by imbalances between generation and demand. The system inertia is expected to be lower in the future than in the past due to the increasing share of VRES. A lower frequency nadir and elevated rate of change of frequency (ROCOF) can create challenges, including measurement inaccuracies, generation disconnection, relay malfunctions, and regional frequency variation [1]. Phasor measurement units (PMUs) provide system operators with frequency and voltage phasor measurements. Operators use measurements to develop models identifying frequency dynamics by estimating the ROCOF, the frequency nadir (minimum). PMUs can enhance the efficiency of frequency controllers such as automatic/continuous frequency control, contingency detection, system integrity protection systems (SIPS), and emergency control like under-frequency load shedding (UFLS) [2].

Wide area monitoring with PMUs can detect disturbances and assess the system's dynamic security for emergency control actions. These emergency control actions can be either response-based, similar to continuous UFLS based on the frequency measurements [3], or event-based, which trigger under a specific condition [4]. However, the number of potential disturbances is vast and requires hand-crafted detection rules, and estimating their performance is challenging. While the accurate dynamic simulation model can solve differential-algebraic equations (DAE) for precise frequency dynamics, the computation time during operation is impractical. High-performance computing offers some speed-up [5], but there is not enough time for applying corrective control actions. Multi-agent system models allow decentralized computation of local power flows to adjust UFLS levels [6]. However, omitting the system's dynamic response would cause inaccurate power flow computations that hinder the applicability of models against unpredictable system disturbances. Reduced order models like Unscented and Extended Kalman Filters (UKF, EKF) combine model knowledge with measurements to estimate dynamic states like rotor angle and frequency, providing a limited prediction horizon [7,8]. Machine learning (ML) models may help to automate the analysis of nonlinear, complex post-disturbance events for emergency control actions by predicting large-scale power systems' stability or dynamic trajectories [9].

ML models are applicable to the frequency stability problems in power systems due to their rapid computation capabilities and high performance [10]. ML models can learn from historical and synthetically simulated data to estimate stability against a predefined disturbance and topology [11–13]. For instance, using decision trees to estimate the transient stability enables faster out-of-step protection [14]. Multiple ML models are trained with the synthetic data to predict the frequency nadir against the predefined frequency event [15]. For frequency protection, neural networks can predict the required amount of UFLS [16]. However, ML has limited generalization capabilities against disturbances that have not occurred before or have not been simulated before, such as cascading disturbances and changes in the network topology. Training one ML model for each topology or disturbance is impractical for large-scale power systems as it requires large training databases studying all possible disturbances [17].

Several types of ML model examples have been studied to predict frequency nadir or trajectory following a disturbance. The deep belief network model predicts the maximum frequency deviation but requires high dimensional features that are hard to monitor accurately in real-time [18]. The long short-term memory model (LSTM) forecasts the subsequent minute system frequency depending on system inertia and loading during the regular operation, not against severe frequency changes [19]. The convolutional neural networks and LSTM models can predict the system frequency response when all the bus's voltage phasors, power demands, generation, and mismatches are available as features requiring a large number of measurements [20]. Increased input dimensions cause slower training procedures. Alterations in the generation or network put the generalization capabilities at further risk and often require retraining with the new training data [21]. High dimensional input and the limited generalization requiring large databases represent current barriers to implementing these ML models for real-time frequency prediction and control.

The neural ordinary differential equation (NODE) model represents an ODE system, where neural network layers approximate system dynamics similar to the residual neural network structure [22]. An ODE solver can compute the system response using the trained NODE and initial conditions. The adjoint sensitivity method can efficiently update NODE parameters during the training while using fewer data points, offering an advantage over chain rule-based gradient computation. Having an ODE representation also enables NODEs for continuous control [23], simulating a system response [24], and long-term forecasting [25]. Power system dynamic simulations can provide sufficient training data for NODEs to estimate dynamic system model parameters [26] and dynamic component modeling [27]. Using physics knowledge from the mathematical model enables NODE to generate the dynamic equivalent for the power system using a large number of dynamic states [28]. Having a large number of states causes NODEs to suffer from long training times. Alternatively, approximation of local dynamics with local measurements allows NODEs to forecast system dynamics in near real-time for security assessment [29]. However, the high number of NODE training points coming from high-resolution measurement devices would create more checkpoints for the ODE solver using the adjoint sensitivity method [30] and is therefore "off-the-shelf" not applicable for real-time frequency prediction and control. The downsampling method can reduce the number of training points without significantly compromising the data quality and possibly make NODE applicable in near real-time tasks.

The downsampling is essential for any ML model when learning from time series data [31]. One downsampling method is adjusting the sampling rate in equally distributed sampled data with data mining [32] or deep sampling [33]. Alternatively, using ML in an active learning structure can obtain the most representative downsampled data [34]. The convex alternating direction method (ADMM) of multipliers [35] or binary non-convex formulations [36] minimize dissimilarity, hence enabling the computation of optimal downsampling that represents the target. Unfortunately, both convex ADMM and binary non-convex formulations rely on the static distribution of data that does not change with time, whereas power system dynamics are diverse and change with time as they depend on the system condition and type (and magnitude) of the disturbance event. Moreover, cascading events occur following a severe disturbance, resulting in unique system trajectories that cannot represent a proper distribution. Alternatively, [37] proposes a Latent ODE workflow that combines recurrent neural networks and NODE to learn from irregular samples but sacrifices the efficient training.

This paper introduces a decentralized application for the near real-time downsampling and training of the NODE in a frequency estimator workflow. The offline dynamic simulations provide a representative sample for the NODE pretraining and hyperparameter tuning. The obtained model acts similarly to the warm start condition in optimization problems to reduce the online retraining times. The estimator workflow monitors a single PMU device to detect the frequency event and transform the measured states for the online NODE retraining to predict event-specific future local measurements such as frequency (nadir). The decentralized nature of the workflow considers the online computations in the local substation where the measurement devices are located to avoid any communication delays. Such workflow has specific predictions for each observed event instead of other ML approaches that try to generalize against all possible scenarios and conditions. This generalization makes the workflow adaptive to system model updates and unknown parameter variations, which would limit mathematical modeling capabilities during the operation. System operators can extend the proposed workflow for all available measurement points in the system

to obtain a comprehensive stability analysis in near real-time without additional computational costs, as each model relies on training data collected in a decentralized fashion. The workflow uses a collocation-based loss computation that provides weights for efficient irregular downsampling of the time series data to achieve faster online training. The predicted frequency trajectory enables the triggering of emergency control actions to improve grid security. A novel trigger function validates the accuracy of frequency prediction from various training epochs. The trigger provides a data-driven fast solution against unpredictable frequency events to improve system security.

Overall, the paper contributes to

1. Applying the NODE model in near real-time training to predict system frequency from a single measurement point before the frequency nadir.
2. Collocation-based irregular downsampling method to improve the efficiency of the online NODE training.
3. Trigger function for early activation of emergency corrective control actions.
4. Testing NODE predictions against noisy PMU data collected from a single measurement point during the regular operation of the Dutch high voltage transmission system.

## 2. Power system frequency response and stability

The system frequency $f$ in power networks refers to the rate at which alternating current (AC) oscillates, indicating the balance between the total generation $P_G$ and total demand $P_L$. The swing equation represents the system's simplified power system dynamic characteristics to compute the frequency deviation $\Delta f$ in response to a sudden change in the balance, where $H$ and $D$ are the system inertia and damping.

$$\frac{d\Delta f}{dt} + D\Delta f = \frac{1}{2H}[\Delta P_G - \Delta P_L] \tag{1}$$

This simplified model does not consider electromagnetic phenomena or include further dynamics from the turbine, governor, and frequency controllers. However, the full mathematical model for the dynamic simulation in this paper can be found in [38]. Furthermore, the high number of VRES units in the grid-forming mode can contribute to the frequency containment reserve via proportional control $S = -\Delta f/\Delta P$ if VRES operates below its maximum capacity. Physical dynamical models, like this simplified model based on swing equations, are limited in approximating the system response. Unobserved and unpredictable changes during the operation and the information regarding the disturbance are challenging to model and could result in long computation times.

The system responds to large frequency deviations with special frequency protection units, preventing damage to equipment around the nominal frequency. For example, UFLS schemes disconnect the predefined demand from the system to avoid dangerously low-frequency values. UFLS activation is based on the frequency and its rate of change; hence, these are the critical parameters of the frequency trajectory that ML model $M$ estimates.

Fig. 1 is a graphical illustration of a regular frequency drop event caused by generation loss in the system with two different NODE estimations from offline and online (before and after training). The initial frequency prediction (dashed green line) $\hat{f}$ from the offline pretrained NODE with the dynamic simulation data requires a short online training with the post-event measurements due to the simulation environment's limitations and the variety in the event and operating conditions. The time span (dotted black line) $\tau_{train}$ is used for training the offline NODE model $M$ to obtain the online model $M^*$ that predicts the dashed blue frequency $\hat{f}^*$ with the complete time span $\tau_{comp}$. Offline prediction is faster but less accurate than online prediction, as offline model predictions do not include any event-specific frequency dynamics besides the initial condition. This paper focuses on estimating the local frequency

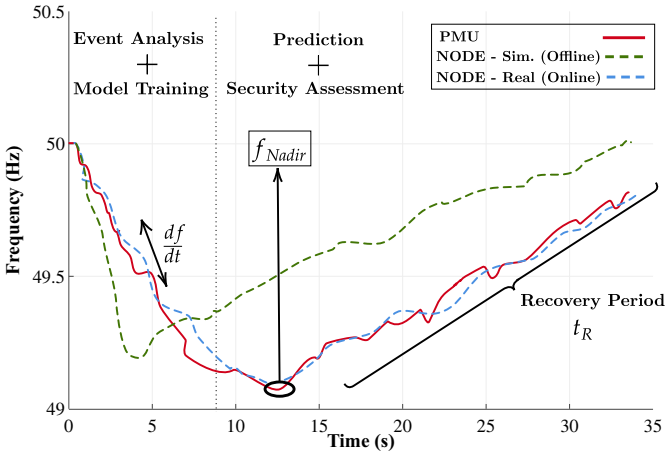



**Fig. 1.** System frequency measurements vs. NODE predictions illustration after and before online training.

nadir and immediately restoring the frequency level from frequency containment reserves and inertial response, ignoring automatic and manual frequency restoration reserves because of slower response times. Capturing restoration response requires additional training following the nadir, which is outside the scope of this paper.

Dynamic simulations use DAE equations with network and controller parameters $\phi$ to compute the dynamic $x$ and algebraic $y$ states from the initial conditions at $t_0$.

$$\begin{aligned} f(x, y, \phi, t) &= \frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x} \\ g(x, y, \phi, t) &= 0 \end{aligned} \tag{2}$$

The numerical ODE solver computes the $x$, approximating the system measurements $\chi$. The solution time sample vector $t^s \in \Re^\kappa$ has irregular time step differences as the solver uses adaptive time step where the number of time steps $\kappa$ depends on the solution complexity and determined time span $\tau_{comp}$.

## 3. NODE frequency estimator

Fig. 2 illustrates the offline (blue) and online (red) workflows of the proposed NODE frequency estimator. The offline workflow is essential for selecting hyperparameters and the pretrained parameters $\theta_0$ of the NODE model $M$ for the online workflow using the representative case $\chi_m^s$ from the dynamic simulation results. The online workflow processes the raw system measurements $\chi\angle\varphi$ from the PMU device to conduct near real-time training starting from the pretrained parameters and predict the future dynamic trajectories $\hat{X}^*$ following a frequency event in the power system. Both pretraining and online training parameters of $M$ are stored in parameter database $\Omega_\theta$ where the final model parameters $\theta^*$ correspond to the minimum validation loss.

### 3.1. Neural ordinary differential equations

Artificial neural network (ANN) models $NN$ construct complex transformations of the given input feature vector $x_{NN}[t] \in \Re^d$ at time $t$ to estimate the future states at the time $t + \Delta t$.

$$NN(x_{NN}[t]) = \hat{x}_{NN}[t + \Delta t] \tag{3}$$

Multiple stacked hidden layers parameterized by $\theta$ with nonlinear activation functions enable ANN to approximate complex functions. The multi-output autoregressive task requires sequentially inputting the prediction results to obtain a total trajectory estimation of the dynamic system response. The prediction horizon and accuracy vary with the

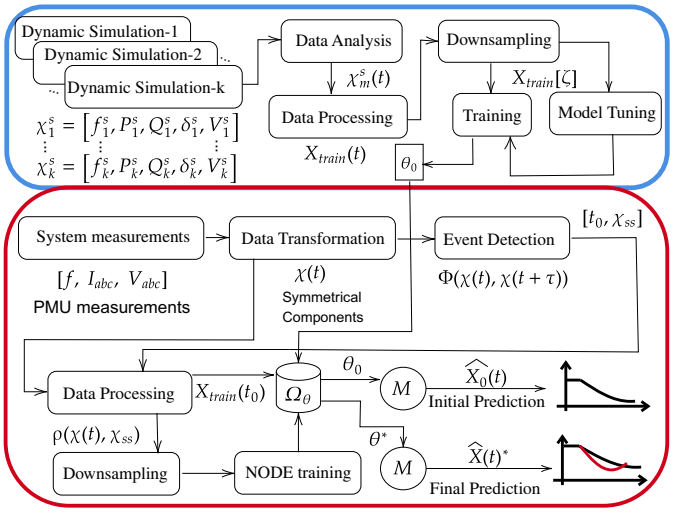



**Fig. 2.** NODE frequency estimator workflow, offline workflow indicated with the blue frame (above), whereas the online workflow with the red frame (below). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

model architecture and input sequence length $\tau_{comp}$. The training data is equally discretized in time, and the gradient descent method updates network parameters $\theta$ in the direction of the negative loss gradient $\frac{\partial L}{\partial \theta}$.

Residual neural networks take the input features to the output of the hidden layers $h$.

$$x(t + \Delta t) = h(x(t), \theta) \tag{4}$$

The formulation resembles Euler's method when $\Delta t \approx 0$. The NODE model combines an ODE solver and residual network layers to learn the hidden state dynamics instead of the state itself. The ODE solves the NODE system from the given initial condition $x_0$ at any point in time [22].

$$\begin{aligned} x(t_0 + \tau_{comp}) &= x_0 + \int_{t_0}^{t_0+\tau_{comp}} h\Big(x(t), \theta\Big) dt \\ &= \text{ODESolve}(x_0, h, \tau_{comp}, \theta) \end{aligned} \tag{5}$$

The backpropagation from the last to the first time step is computationally inefficient as the ODE solver has adaptive time steps depending on the error. Instead of the direct application of the chain rule, the adjoint sensitivity method computes the gradients by constructing a new ODE system using the adjoint states $a(t) = \frac{\partial L}{\partial x}$.

$$\frac{\mathrm{d}a}{\mathrm{d}t} = -a(t)^T \frac{\partial h(x(t), \theta)}{\partial x} \tag{6}$$

The loss gradient is equivalent to vector Jacobian pairs computed by the reverse mode automatic differentiation. The adjoint sensitivity ensures the computational efficiency as a single ODE solver call is sufficient for each epoch but requires deep networks when the ODE solver takes many time steps.

The collocation-based loss function $L_c$ provides faster training as the loss is based on the system dynamics rather than the simulated solution (label) [39]. The collocation method estimates the derivative of the selected points $\dot{x}[k]$, $k = [1, 2, \dots, \kappa]$ hence, the parameter $\theta$ update does not require any ODE solver.

$$L_c = \frac{1}{\kappa} \sum_{i=1}^{\kappa} \Big( \dot{X}[i] - h(x[i], \theta) \Big)^2 \tag{7}$$

When NODE learns a true ODE system, collocation-based learning is an ideal alternative, as the learned dynamics satisfy actual dynamics. However, the actual dynamics of a single measurement point in a power system depend on internal controllers and unobserved states of generation units. That's why the direct application of the collocation method cannot satisfy an ODE equivalent and has lower approximation capabilities than training with the adjoint sensitivity method.

### 3.2. Offline workflow

The offline workflow acts as an accelerator for NODE training and verifies the model performance based on the analyzed simulation results for the selected measurement point. The dynamic simulation model of the observed system computes the power system states from the measurement point of the system using different system loading conditions $\sum P_L$, share of VRES $\sum P_{RES}/(\sum P_{RES} + \sum P_{syn})$, and dispatch profiles where $\sum P_{RES}$ and $\sum P_{syn}$ indicate total renewable and synchronous generation in the system. Alternatively, (if possible) the previous measurements from the actual events can provide the representative case. The randomized dispatch profile satisfies the system's generation and power flow limits. $k$ number of simulation results in the vector $\chi^s = [f^s, P^s, Q^s, \delta^s, V^s]$ of the measurement bus containing frequency $f^s$, active power injection $P^s$, reactive power injection $Q^s$, phase angle difference $\delta^s$, and voltage magnitude $V^s$.

We define the representative sample that has a nadir frequency value $\bar{f}_m^s = \min(f^s)$ and corresponding time $\bar{t}_m^s$ which is equal to the minimum Euclidean distance to the mean nadir $\eta = [f_{Nad}, t_{Nad}]$ where $f_{Nad} = \frac{1}{k}\sum_i \bar{f}_i^s$ with $t_{Nad} = \frac{1}{k}\sum_i \bar{t}_i^s$. To overcome the unit difference between frequency and time, all values are z-normalized for the consistent distance calculation. Alternative distance metrics are also suitable for the identification.

$$\min_{i \in 1,2,\dots,k} \quad \|z_i - \eta\|_2 \tag{8}$$

$$\text{s.t.} \quad z_i = [\bar{f}_i^s, \bar{t}_i^s] \tag{9}$$

The representative dynamics $\chi_m^s(t)$ are scaled with $\rho$ based on the pre-disturbance steady state operating values $\chi_m^{ss}$. The aim of obtaining $\chi_m^s(t)$ is not to provide an initial security assessment but to reduce the required number of training epochs. Having a representative sample with nadir values outside of the distribution could cause a higher initial error in NODE predictions, but online retraining ensures correct security assessment.

$$\rho(\chi_m^s(t), \chi_m^{ss}) = X(t) = \frac{\chi_m^s(t)}{\chi_m^{ss}}, \quad t \in [t_0, t_0 + \tau_{comp}] \tag{10}$$

The training $X_{train}$, validation $X_{val}$ and test $X_{test}$ data come from the feature matrix $X(t)$ with the corresponding time spans $\tau_{train}, \tau_{val}$, and $\tau_{test}$. The $X_{train}$ with a proper time span that contains sufficient information about the system dynamics includes many time samples. High-resolution PMUs or adaptive ODE solvers provide solutions with short steps to represent the actual dynamics accurately. Sampling from the original dataset ensures faster NODE training.

Fig. 3 illustrates an arbitrary simulation data (b), its collocation points (a), and the absolute error between NODE's NN output and collocation points (c). The proposed irregular sampling method is based on weighted random sampling of measurements to generate training data for the NODE. Randomly sampling the specified number of points $\beta < \kappa$ results in the subset $\zeta \subset t_{train}, |\zeta| = \beta$ containing time samples. However, having equal weight for all time samples in the random sampling might cause more stationary points in training, resulting in NODE fitting simpler dynamics.

Collocation-based sampling assigns a weight vector $\omega \in \Re^{\kappa}$ for each time sample from $t_{train}$ based on the loss $\epsilon$ between the collocated data and the NN output.

$$\omega = \left( \frac{\mathrm{d}X_{train}}{\mathrm{d}t} - h(X_{train}, \theta) \right)^2 = (\epsilon[t_{train}])^2 \tag{11}$$

Hyperparameters of the NODE model $M$ are determined based on the weighted sampled set $X_{train}[\zeta]$. The weighted random sampling on the

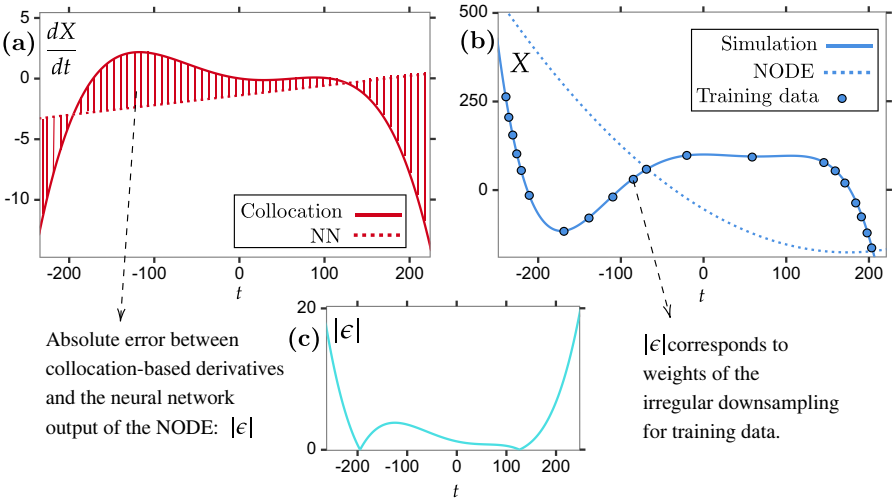



**Fig. 3.** Illustration of collocation-based irregular downsampling. (a) collocation gradients (—) and NN output (···). (b) simulation results (—), NODE predictions (···), and sampled training data (●). (c) the absolute error between gradients (—).

training time samples $t_{train}$ generates the time subset $\zeta$. These hyperparameters can be tuned with state-of-the-art algorithms. ODE solver type and tolerances specify the numeric computation method, which affects the accuracy and the solver time. NN parameters such as the number of neurons, layers, and activation functions adjust the model complexity. The last training hyperparameters are the learning rate and the type of gradient descent solver. $M$ with the selected hyperparameters uses the sampled training data $X_{train}[\zeta]$ to obtain the pretrained model parameters $\theta_0$. The function

$$\text{Train}(h, \theta, X_{train}, \zeta, \tau_{train}) \tag{12}$$

trains one epoch and updates the network parameter $\theta$. Training continues until the total scalar loss $L_1 = L([X_{train}, X_{val}], [\hat{X}_{train}, \hat{X}_{val}])$ converges or the maximum number of epochs $N_{max}$ is reached. Any scalar loss function is possible for selecting $L$, such as mean squared error (MSE) or mean absolute error (MAE). The prediction is $\hat{X} = \text{ODESolve}(x_0, h, t, \tau, \theta)$. The pretrained model supplies the prediction of the dynamics $\hat{X}_0$, which is more accurate than a completely randomly initialized model, even for a fault that has never occurred in the training data. Moreover, the random initialization likely causes an unstable ODE system.

### 3.3. Online workflow

The online decentralized workflow begins with transforming local PMU measurements and processing data into the dynamic features $X$. The workflow computes the initial prediction following the event detection using the initial condition $X_0$. Collocation-based downsampled features are the inputs for the NODE training. The final model predicts this using parameters resulting in a minimum loss.

The PMU provides the bus measurements of frequency $f$, three-phase phasors of current $I_{abc} = [I_a, I_b, I_c]$ and voltage $V_{abc} = [V_a, V_b, V_c]$. Power flow equations compute the active $P$ and reactive $Q$ powers transferred through the PMU where $\bar{x}$ indicates the complex conjugate of $x$.

$$P + jQ = V_a\bar{I}_a + V_b\bar{I}_b + V_c\bar{I}_c \tag{13}$$

As simulations compute system dynamics in a single-phase equivalent network, we apply the following transformation to alter the three-phase to its symmetrical components using the complex rotation operator $\alpha = -\frac{1}{2} + \frac{\sqrt{3}i}{2}$.

$$\begin{bmatrix} V_+ \\ V_- \\ V_0 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{14}$$

The positive sequence state voltage magnitude $|V_+|$, phase angle difference $\delta = \angle V_+ - \angle I_+$, frequency $f$, active power $P$, and reactive power $Q$ constitute the set of dynamic states $\chi$. The decentralized event detection function $\Phi(\chi(t))$ screens the local frequency values from the PMU. The function finds the initial condition of $M$ from the time sample $t_0$ when the frequency deviates from the base frequency $f_b$ by more than the predefined threshold $\varsigma$. Otherwise, the function

$$\Phi(\chi(t)) = \begin{cases} t_0 = t, & \text{if } |f(t) - f_b| \le \varsigma \\ \chi^{ss} = \chi(t), & \text{otherwise.} \end{cases} \tag{15}$$

records the steady state dynamics before the disturbance $\chi^{ss}$ for scaling. Data processing $\rho$ (as shown in Fig. 2) described in Sec.3.2 scales the dynamic states $\chi(t)$ with $\chi^{ss}$ for the time period $t_0 \le t \le t_0 + \tau_{train}$ to obtain the training data $X_{train}(t)$. $M$, with the pretrained parameters $\theta_0$ computes the initial prediction $\hat{X}_0$ before real-time training. The collocation-based downsampling computes $\omega$ and corresponding weighted random sampled points $X_{train}[\zeta]$ for real-time training.

The parameters database $\Omega_\theta$ collects the NODE parameters $\theta$ after each epoch until the validation data $X_{val}$ is collected from the processed system measurements $X(t) = \rho(\chi(t), \chi^{ss})$ where $t_0 + \tau_{train} \le t \le t_0 + \tau_{val}$. After the validation set is available, the parameter that provides the smallest $L_1$ loss is identified.

$$\min_{\theta \in \Omega_\theta} \quad L([X_{train}, X_{val}], [\hat{X}_{train}, \hat{X}_{val}]) \tag{16}$$

$$\text{s.t.} \quad \hat{X}_{train} = \text{ODESolve}(X_{train}[t_0], h, \tau_{train}, \theta) \tag{17}$$

$$\hat{X}_{val} = \text{ODESolve}(X_{val}[t_0], h, \tau_{val}, \theta) \tag{18}$$

The decentralized training continues until the maximum epoch $N_{max}$ is reached while updating the $\Omega_\theta$ when a new prediction $\hat{X}$ has a smaller $L_1$. System operators can access intermediate predictions from $\Omega_\theta$ to analyze the learning loss and take actions before the algorithm reaches the $N_{max}$. Moreover, operators can monitor the frequency error of the model with upcoming PMU data in near real-time.

### 3.4. NODE-based trigger for emergency corrective control actions

The decentralized frequency prediction of the NODE model can activate certain corrective control actions earlier than the system frequency

**Algorithm 1:** The NODE-based trigger function.

**Input:** $M, \tau_{train}, \tau_{val}, \Omega_\theta, \lambda, \gamma, \rho, f_{train}, f_{val}$

$count = 0$
**for** $\theta$ in $\Omega_\theta$ **do**
  $e_{train} = \frac{1}{N}\sum_{k=1}^{N} |M(\theta, \tau_{train})^f[k] - f_{train}[k]|$
  $e_{val} = \frac{1}{N}\sum_{k=1}^{N} |M(\theta, \tau_{val})^f[k] - f_{val}[k]|$
  **if** $e_{train} < \gamma$ & $e_{val} < \gamma$ **then**
    $count = count + 1$
  **end**
  **else**
    $count = 0$
  **end**
  **if** $count = \rho$ & $\min(M(\theta, \tau)) < \lambda$ **then**
    **return** 1
  **end**
**end**

$f(t)$ reaches the predefined threshold value $\lambda$ at any time $t$. The main principle of control actions is adding generation or reducing demand to balance the power mismatch causing frequency drop. The triggering function $c(f(t), \lambda) = 1$ activates the predefined control action when the system frequency reaches $\lambda$ where,

$$c(f(t), \lambda) = \begin{cases} 1, & f(t) < \lambda \\ 0, & f(t) \geq \lambda \end{cases} \tag{19}$$

The early triggering of the control actions would stop the frequency drop in the system and increase the frequency nadir. This trigger could save the power system in emergency situations like generation disconnection from low-frequency values. The NODE-based trigger function described in Algorithm 1 $c'(M, \Omega_\theta, \lambda, \gamma, \rho)$ triggers the predefined control action before system frequency reaches the limit. Alg.1 checks if $c(\hat{f}(t), \lambda) = 1$ and the frequency MAE of the training $e_{train}$ and validation $e_{val}$ are lower than $\gamma$ for at least $\rho$ consecutive models from $\Omega_\theta$. As described in Sec. 3.3, $\Omega_\theta$ stores model parameters $\theta$ during the training. $\gamma$ and $\rho$ are controllable parameters to set the confidence of NODE predictions during the training. For more convenient reading, the predicted frequency of the NODE is denoted as $M(\theta, \tau)^f$ in the Alg.1. System operators can adjust the algorithm's controllable parameters by training more epochs with stricter error limits to improve the nadir prediction.

## 4. Case study

The studies use a modified version of the IEEE 39-bus test system. Buses 1, 9, 16, 17, and 28 contain WECC Type 3 wind turbines [40] with 1000MW capacity to investigate the proposed workflow's performance under various VRES shares. Bus 39 is the measurement point, as the bus represents the interconnection point of the system. Dynamic simulations consider four system loading levels [90 %, 75 %, 60 %, 45 %] of the maximum generation capacity of the synchronous machines and three VRES shares of the total generation [40 %, 20 %, 0 %]. Synchronous machines and load dispatches are randomized 100 times, with the fixed voltage point for PV buses and power factor for PQ buses, while wind turbines have equal generation output. Dispatches without converged power flow results are replaced with other random feasible operating conditions. The final dynamics database contains 1200 samples for the frequency estimator to test the proposed framework against other ML baselines.

To investigate the frequency stability, dynamic simulations computed and recorded the selected states, as described in Sec 3.2, from the interconnection point against the disconnection of the largest synchronous machine, excluding the swing and interconnection. A 3-phase short circuit with $0.1 + j0.1\Omega$ fault impedance is applied to the high voltage bus containing the disconnected generator for 150ms, and the total simulation duration is 40s. If frequency measurements go above 65Hz or below 55Hz, the simulation is terminated due to the blackout condition for the 60Hz system, and no nadir values are observed. The NODE can still learn the frequency trajectories of these extreme cases, but we cannot provide any error information regarding the nadir as it does not occur. Generators 3 and 8 are turned off when the system loading is low and the VRES share is high. Dynamic simulations do not include electromagnetic transients (EMT) models because the time scale of the frequency response does not require the computation of switching actions in power electronic devices. The proposed NODE frequency estimator is suitable with EMT models, but higher downsampling would be required to achieve similar training times.

The event detection threshold was selected as $\varsigma = f_b \pm 0.1$Hz for the test system. Following the event detection, the preprocessed simulation results were split into training $\tau_{train} = 4.95$s, validation $\tau_{val} = 3.47$s and test $\tau_{test} = 31.32$s sets selected based on experience where longer training times could be used if the system responses are slower which is feasible for larger grids compared to our test network. The NODE model, which has the lowest mean squared error (MSE) loss on training and validation time span ($\tau_{train} + \tau_{val}$), predicts the complete horizon ($\tau_{train} + \tau_{val} + \tau_{test}$) trajectory $\hat{X}$. The base training and complete sets contain 497 and 3977 time sample points, whereas random and collocation sampling use only 100 for the NODE training. The NODE model has 2 hidden layers with 50 neurons per layer using the hyperbolic tangent activation function. Tsitouras 5/4 Runge-Kutta method solves the ODE system to compute the MSE loss for the NODE training with 850 and 60 epochs for offline and online, respectively. ADAM solver with 0.005 learning rate updates NODE parameters $\theta$. ANN (same architecture as NODE and trained with 500 epochs), SVM (Gaussian kernel), and linear regression models used the same training data for comparison against the NODE model. All models have autoregressive structures to train and predict measurement values. Besides the ANN, each predicted state has a separate model. $R^2$, MSE, and discrete Fréchet distance (DFD) [41] metrics are used for performance measurement.

PMU recordings from a real-life operation have been investigated by the NODE and reduced-order models. SINDy algorithm [42] provides a reduced order model required by UKF [7] and EKF [8] models to predict future PMU states. The SINDy model is a third-degree polynomial model where initial and measurement noise covariances are selected as 0.1 for UKF and EKF.

DigSilent PowerFactory 2022 [43] computed the dynamic trajectories. The studies were conducted in Julia v1.7 [44] using the NODE modeling DiffEqFlux.jl, time studies BenchmarkTools.jl, and other ML models MLJ.jl, Flux.jl libraries. The experiments were conducted on a virtual machine with "Intel(R) Xeon(R) Gold 6148 CPU@2.4 GHz" and 16GB RAM.

### 4.1. NODE frequency estimator performance in offline workflow

The first case study investigates the regression and training performance of the decentralized NODE models for the selected pretrained case with three training methods: base and collocation without any sampling, random sampling, and collocation sampling.

The pretraining case $\chi_m^s$ has the minimum Euclidean distance to each trajectory's z-normalized nadir frequency and corresponding time instant as in Sec. 3.2. The pretraining case has 0 % VRES share and 60 % system loading. The frequency nadir of the pretraining case is 59Hz at 15.3s, and the average frequency nadir of the entire dataset equals 58.93Hz at 17.2s.

The adjoint sensitivity and collocation-based training methods were applied to various samplings of the training data computed from the pretraining case $\chi_m^s$. While evaluating the performance metrics, NODE computed the final solution for the same time points as the original training data. The parameter sets $\theta^*$ with the lowest validation MSE loss during the training were used for the final model prediction.

**Table 1**
Comparison of various training methods of NODE for the selected pretraining case on the training data.

| Training Method | Sampling | MSE | DFD | $R^2$ | Time(Epoch/s) |
|---|---|---|---|---|---|
| Adjoint Sens. | None | 4.17 | 0.523 | 0.94 | 0.23 |
| Collocation | None | 39.2 | 17.6 | 0.05 | 0.12 |
| Adjoint Sens. | Random | 3.44 | 0.32 | 0.96 | 0.15 |
| Adjoint Sens | Collocation | 4.87 | 0.55 | 0.94 | 0.14 |

**Table 2**
Comparison of various training methods of NODE for the selected pretraining case on the complete data.

| Training method | Sampling | MSE | DFD | $R^2$ |
|---|---|---|---|---|
| Adjoint Sens. | None | 12.39 | 0.523 | 0.847 |
| Collocation | None | 170.22 | 247.26 | 0.001 |
| Adjoint Sens. | Random | 26.8 | 0.89 | 0.729 |
| Adjoint Sens. | Collocation | 2.65 | 0.55 | 0.96 |

Tables 1 and 2 contain various NODE training methods' performances on the training and complete data (the combination of training, validation, and test sets). The training time of a single epoch for each case in Table 1 shows that training takes a significant amount of time in the base case, while collocation-based training is the fastest but results in the least accurate prediction due to the direct computation of the loss gradients from the estimated collocations instead of using an ODE solver. Using collocation loss as a training method caused an unstable ODE system because the proposed workflow approximates an ODE system only from local measurements, where larger and more complex system dynamics can affect observed dynamics. Although random sampling performs slightly better on training data, collocation sampled data has a ten times lower MSE for the complete data set because of more training samples from regions with higher initial collocation loss. Moreover, using centralized training with measurements from multiple locations in the system will increase the training time significantly as solving ODEs becomes much more computationally expensive. The centralized training will be infeasible during near real-time training.

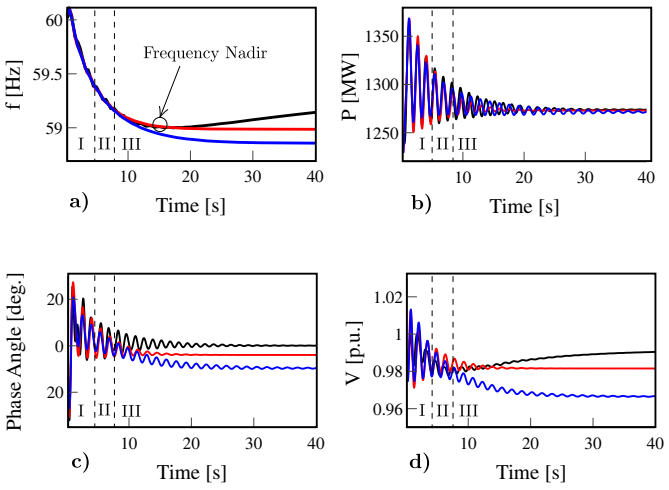



**Fig. 4.** Simulation data (—) of the power system against an under-frequency event is divided into training (I), validation (II), and test (III) sets. NODE predictions of frequency (a), active power injection (b), phase angle difference (c), and voltage (d) trained with the adjoint sensitivity method without any sampling (—), and collocation sampled (—).

**Table 3**
Online Frequency Prediction Performance (mean ± std.) of the NODE and other ML models for the dynamic simulations of the modified IEEE 39-bus system.

| Model | Data | MSE | DFD | $R^2$ | $\Delta f_{Nad}$ (Hz) |
|---|---|---|---|---|---|
| Linear | Train | $0.9 \pm 1.6$ | $0.2 \pm 0.5$ | $1 \pm 0.1$ | – |
| SVM | Train | $0.6 \pm 1.8$ | $0.1 \pm 0.4$ | $1 \pm 0.1$ | – |
| ANN | Train | $2 \pm 1.9$ | $0.2 \pm 0.3$ | $1 \pm 0.1$ | – |
| NODE | Train | $14 \pm 5.1$ | $0.6 \pm 0.5$ | $0.8 \pm 0.4$ | – |
| Linear | Complete | $(3 \pm 9) \times 10^6$ | $25 \pm 28$ | $0.7 \pm 0.5$ | $12 \pm 20$ |
| SVM | Complete | $83 \pm 24$ | $0.9 \pm 2$ | $0.6 \pm 0.5$ | $0.4 \pm 0.6$ |
| ANN | Complete | $97 \pm 25$ | $0.9 \pm 2$ | $0.5 \pm 0.6$ | $0.3 \pm 0.6$ |
| NODE | Complete | $26 \pm 10$ | $0.7 \pm 0.8$ | $0.8 \pm 0.4$ | $0.3 \pm 0.4$ |

Fig. 4 shows actual and NODE predictions of power system simulation results from the two models with collocation sampling and without any sampling. Both results captured a significant portion of the training and complete data. The proposed collocation sampling algorithm has the least frequency and voltage estimation error, where both dynamic states can be crucial for the dynamic security assessment. This study shows that the approximated ODE system has a similar oscillation characteristic to the simulation results, as the NODE could approximate available measurements. In the case where a mathematical model of the system does not exist, historical frequency recordings could be used for the training. This could cause the pretrained prediction in blue curves to be further away from the actual frequency because of unseen operating conditions, system topologies, and frequency events. However, the online training ensures that NODE adapts to the system changes to capture the frequency.

### 4.2. NODE frequency estimator performance in online workflow

This case study investigates the feasibility and performance of real-time collocation-based sampled training of the NODE for the decentralized nadir prediction task using the dynamic simulation results under various operating conditions and VRES shares. The NODE's performance is compared against several baseline models, such as linear regression, SVM, and ANN models, in an autoregressive fashion where the predictions are used as new input features.

Table 3 contains performance metrics and nadir errors for all ML models in both training and complete trajectories by taking the mean and standard deviation of the metrics. The NODE has the highest MSE loss in training data, whereas the other models provide similar outputs that fit the observed data perfectly. However, as these models do not have any ODE system properties, the test performance is worse than that of the NODE. ANN and SVM models perform similarly as the two models can learn nonlinear functions better than the linear model. All models' output error increases when the system has growing oscillations and large excursions in voltage or frequency. Higher DFD values in other ML models showed larger deviations in the predictions that could cause false or missed alarms. Smaller DFD and higher $R^2$ scores show that NODE has the highest prediction capability. The number of epochs in the training of the NODE (60) is much lower than that of ANN (500). At 60 epochs, ANN was underfitted, and its predictions had high errors.

Further analyzing the NODE predictions revealed insights into trajectory complexity and variation in the learning performance in Table 4 under various system loadings and VRES shares. When the system loading is lower, the impact of losing the largest generation is a less severe disturbance, so the NODE learns the ODE dynamics more accurately. As system loading increases, unstable system responses cause higher frequency nadir errors due to unconfined frequency excursions. Even though a higher VRES share in generation lowers the system inertia, the distribution of the total generation improves the system's resiliency. The highest average nadir error occurs when the system loading reaches 90 % of the capacity, and the generation only comes from the synchronous machines.

**Table 4**
Online Frequency prediction (complete) performance (mean ± std.) of the NODE model for the dynamic simulations.

| Load | VRES | MSE | DFD | $R^2$ | $\Delta f_{Nad}$ (Hz) |
|---|---|---|---|---|---|
| 45 % | 0 % | 7 ± 0.6 | 0.2 ± 0.2 | 0.9 ± 0.1 | 0.2 ± 0.2 |
| 45 % | 20 % | 5 ± 3 | 0.5 ± 0.3 | 0.9 ± 0.1 | 0.1 ± 0.1 |
| 45 % | 40 % | 5 ± 1 | 0.3 ± 0.1 | 0.9 ± 0.03 | 0.1 ± 0.1 |
| 60 % | 0 % | 11 ± 4 | 0.5 ± 0.3 | 0.9 ± 0.1 | 0.2 ± 0.2 |
| 60 % | 20 % | 15 ± 5 | 0.6 ± 0.3 | 0.9 ± 0.1 | 0.2 ± 0.3 |
| 60 % | 40 % | 10 ± 5 | 0.5 ± 0.4 | 0.9 ± 0.1 | 0.2 ± 0.3 |
| 75 % | 0 % | 22 ± 7 | 0.7 ± 0.5 | 0.9 ± 0.2 | 0.3 ± 0.3 |
| 75 % | 20 % | 21 ± 6 | 0.7 ± 0.4 | 0.8 ± 0.2 | 0.3 ± 0.3 |
| 75 % | 40 % | 27 ± 7 | 0.8 ± 0.6 | 0.7 ± 0.2 | 0.3 ± 0.4 |
| 90 % | 0 % | 54 ± 8 | 1 ± 0.8 | 0.6 ± 0.5 | 0.8 ± 0.5 |
| 90 % | 20 % | 77 ± 19 | 1 ± 1.4 | 0.8 ± 0.4 | 0.3 ± 0.3 |
| 90 % | 40 % | 51 ± 10 | 1 ± 1 | 0.7 ± 0.5 | 0.3 ± 0.3 |

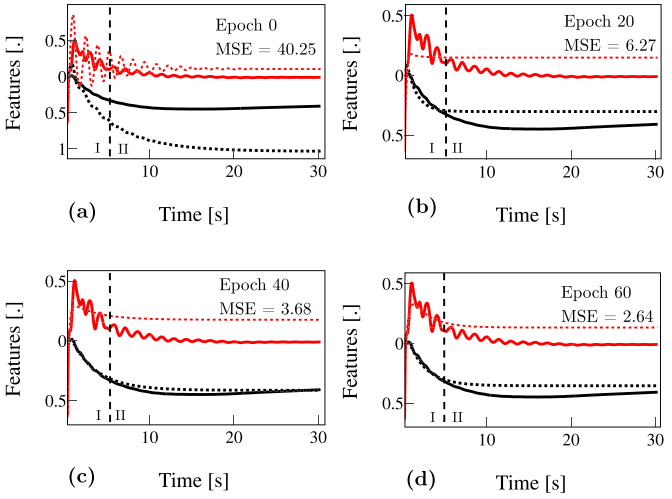



**Fig. 5.** Scaled simulation results of frequency deviation $\Delta f$ (—), and active power deviation $\Delta P$ (—) with corresponding scaled NODE predictions of $\Delta \hat{f}$ (·····), $\Delta \hat{P}$ (·····) at 0 (a), 20 (b), 40 (c), and 60 (d) training epochs.

### 4.3. Online training performance of NODE frequency estimator

This case study illustrates the impact of online training on the frequency and active power predictions collected from different training epochs starting from the pretrained model for a random case.

Fig. 5 shows four plots of the simulated frequency and scaled active power deviations with corresponding NODE predictions. The initial prediction at epoch 0 has the highest MSE= 40.25, DFD= 0.34, the lowest $R^2$ = 0.94 and wrong nadir estimation, but the active power estimation converges to the correct steady-state value. NODE predictions after 20 epochs had a minimal error in nadir frequency. The adaptive nature of the proposed workflow allows system operators to obtain prediction results from earlier training epochs. Considering all predicted dynamic states, the model reached the optimum frequency response at 40 epochs, but the highest performance was observed at 60 epochs.

The online training takes approximately 3.3 s for 60 epochs in the described virtual machine, whereas shorter training is possible with higher computational power or optimized software. NODE prediction of the frequency trajectory requires solving the ODE system, which takes around 20ms and is significantly shorter than the training. Neural network architecture adjusts the complexity of the ODE system and predicted trajectories therefore, the total training time depends on the selected model and ODE solver hyperparameters.

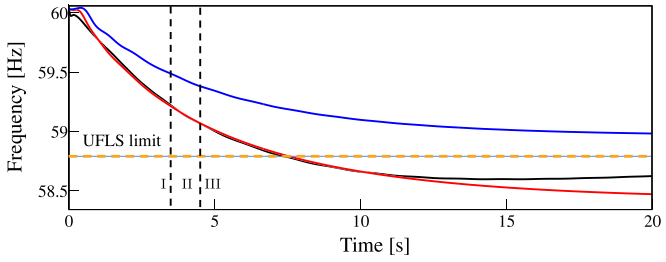



**Fig. 6.** NODE frequency predictions before (—) and after (—) pretraining. The simulated frequency response (—) is divided into training (I), validation (II), and test (III).

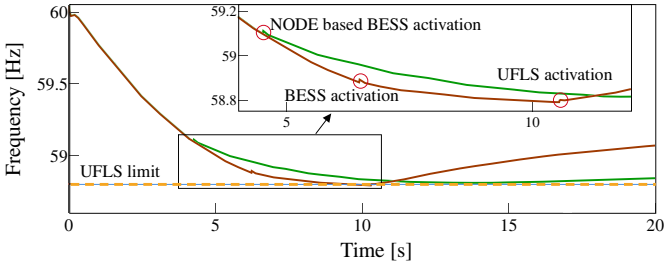



**Fig. 7.** The simulated frequency responses with (—) and without (—) NODE aided early BESS activation to avoid the UFLS limit.

### 4.4. NODE-based activation of emergency corrective control action

In this case study, we analyzed two cases showing the consequences of NODE-based corrective control actions. Both cases used a three-phase short circuit to disconnect generator 3 from the network. The generation loss was 600MW in the first case and 1000MW in the second, where the total generation was 3845MW, and the VRES share was 54.3 %.

Fig. 6 contains actual frequency and NODE predictions before and after the pretraining. High initial error before the pretraining in frequency prediction indicated an inaccurate UFLS triggering condition assessment. Although the system frequency dropped below the UFLS limit, we turned off the UFLS to monitor the nonlinear system response in the simulation. The final model had the minimum loss for the validation data among the candidate models where the trigger function $c'$ activated the control action described in Sec. 3.4.

Simulation results in Fig. 7 show the early triggering of the BESS arrested frequency nadir above the UFLS limit without providing additional energy compared to the later threshold-based activation. The later activation caused system frequency to drop below the UFLS limit and disconnect the 50MW load. However, the NODE-based activation had a lower frequency at the end of the simulation since UFLS contributed as an additional frequency reserve.

The second case had a more severe generation loss than the first, as illustrated in Fig. 8. Similar to Fig. 6, machine frequency protection systems were turned off to observe the frequency trajectory where, in real operation, low frequencies would trigger protection systems. The initial prediction couldn't predict the system frequency dropping below the machine limit. Following the pretraining, NODE detected the critical limit violation despite the severe frequency drop.

Fig. 9 shows that the NODE-activated SIPS prevented further generation disconnections that caused extremely low-frequency nadir. The SIPS disconnected two predefined lines between buses 23–24 and 16–21, which caused the isolation of buses 21, 22, and 23 with connected loads and generation units. The total loss of generation and load was 280MW and 1300MW respectively. Although the load disruption was significant,

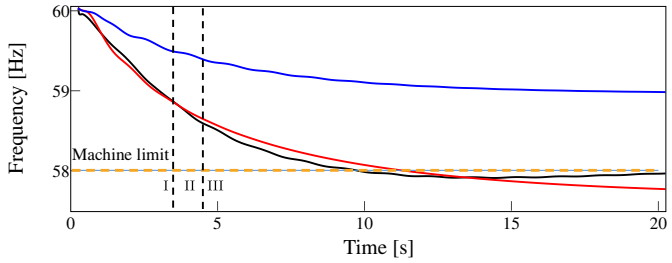



**Fig. 8.** NODE frequency predictions before (—) and after (—) pretraining. The simulated frequency response (—) is divided into training (I), validation (II), and test (III).

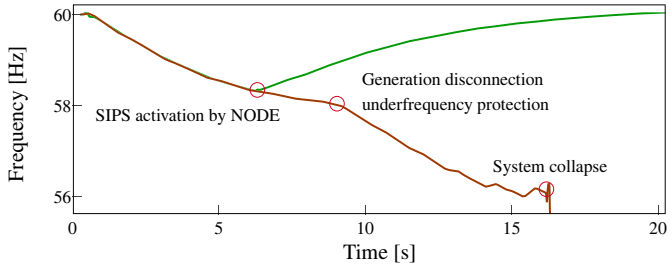



**Fig. 9.** The simulated frequency (Hz) responses with (—) and without (—) NODE aided SIPS to avoid system collapse.

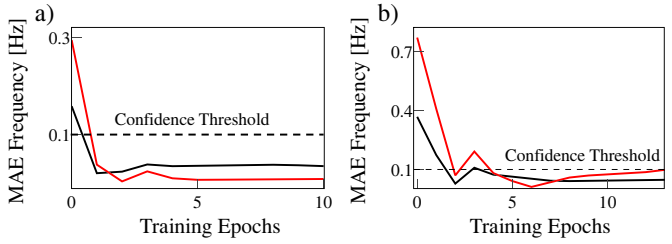



**Fig. 10.** MAE of frequency during NODE training for training (—) and validation (—) data for the first (a) and second (b) case.

without the SIPS system, cascading generation disconnections would have caused the system to collapse.

Fig. 10 shows the training performance of the NODE for both cases for training and validation data. The NODE trigger function $c'$ had 0.1Hz confidence threshold $\gamma$, higher than the NODE prediction error for $\rho = 10$ consecutive epochs. In the first case, after the first epoch, the model reached a good approximation for the frequency trajectory; in the second case, after the 4th epoch, MAE dropped below the threshold. The total training time is computed using Table 1 as the first and second cases take 1.4 and 1.96 s, respectively. Overall, the model trained fast enough for real-time operation for two different frequency trajectories starting from the same initial parameters.

### 4.5. NODE performance on PMU data

In this case study, we investigated the learning performance of the NODE against highly noisy, real, normal operation data. The Netherlands' transmission system operator (Tennet) provided a PMU recording of the 380kV high voltage network for 23.09.2022 from an undisclosed location and time of the day. We applied the transformations provided in 3.3 to extract the positive sequence from the three-phase data for training and then preprocessed it as in the previous studies.

Fig. 11 shows scaled PMU frequency, voltage, and active power flow measurements and corresponding NODE predictions. The frequency

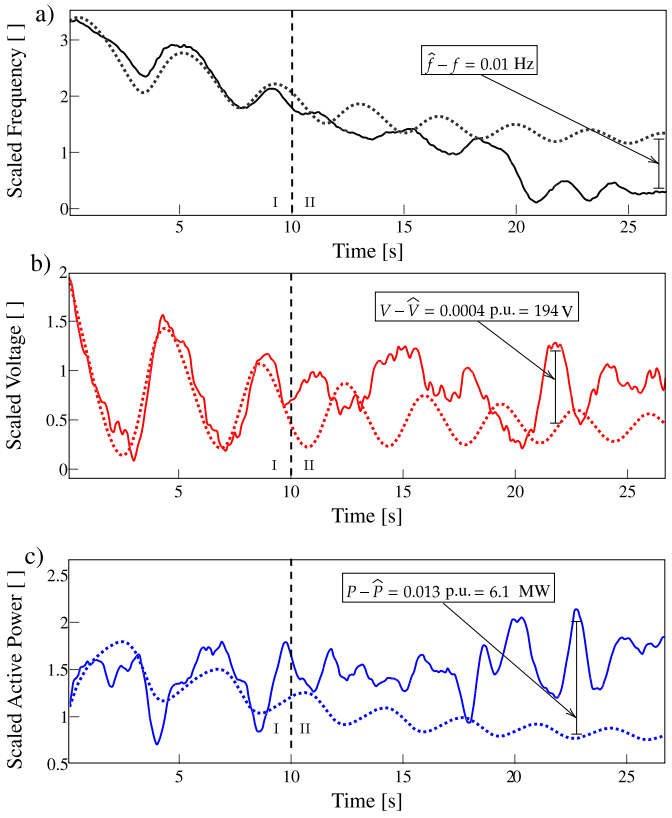



**Fig. 11.** Scaled and transformed NODE predictions and PMU measurements of frequency (a) ($\hat{f}$ (·····), $f$ (—)), voltage magnitude (b) ($\hat{V}$ (·····), $V$ (—)), and active power (c) ($\hat{P}$ (·····), $P$ (—)). Measurements are divided with a dashed line into training and validation (I) and test (II) sets.

drop was not due to an extreme event as the minimum recorded frequency is 49.978Hz, and the initial recorded frequency is 50.02748Hz. The NODE captured the frequency characteristics even under highly noisy system data. Under the noisy positive sequence PMU voltage profile, NODE learned an ODE representation for the real system measurements. Unlike other PMU states, high variability in generation and demand decreased the model's prediction capability. However, the reduction in prediction did not significantly impact the operational decisions since the variation in the measured power could be regarded as noise.

Fig. 12 contains the provided PMU data and predictions from SINDy, EKF, and UKF models. All three approaches have similar prediction capabilities to other supervised learning models in 4.2. The SINDy model has the closest prediction to the PMU data because the identified ODE system started from a steady-state initial condition which caused a constant response during the testing. EKF and UKF suffer from growing errors due to auto-regressive predictions for long prediction horizons.

Unlike other case studies computed with mathematical models, real power systems contain an uncountable number of dynamic elements that change dynamic measurements continuously. That's why retraining following each event is necessary to learn the existing dynamic elements and their responses to unpredictable events.

### 4.6. Discussion

This paper proposes the NODE frequency estimator workflow for the power system frequency stability problem using real-time training with the collocation-based downsampling technique. The pretraining and weighted sampling process enable online and fast NODE training. The downsampling reduces the number of training points while

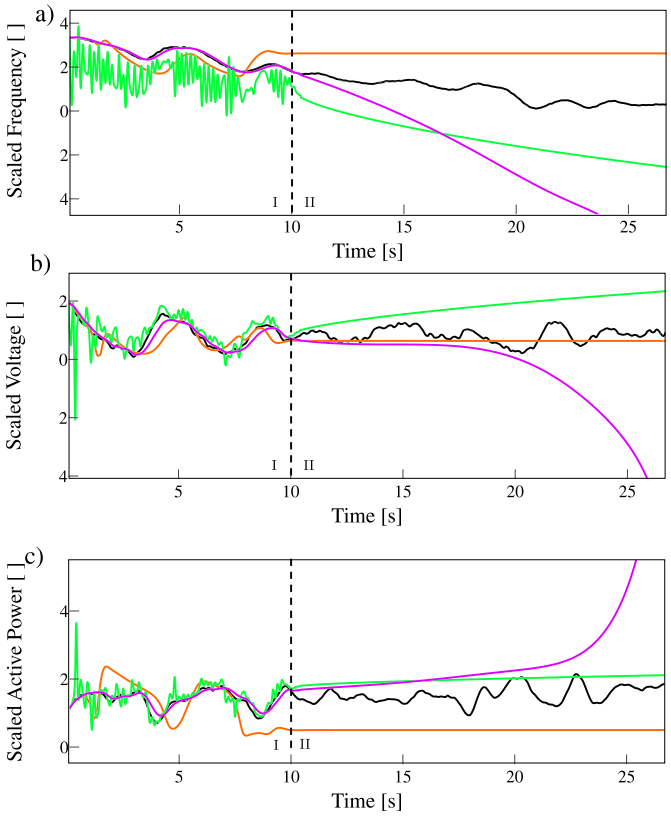



**Fig. 12.** Scaled and transformed PMU measurements (—), SINDy prediction (—), EKF prediction (—), and UKF predictions (—) of frequency (a), $f$, voltage magnitude (b), $V$, and active power (c) $P$. Corresponding SINDy model solution Measurements are divided with a dashed line into training and validation (I) and test (II) sets.

collocation-based weighting preserves the dynamic characteristics, resulting in higher performance in NODE prediction. System operators can take corrective control actions using the provided future frequency trajectory and nadir value. Moreover, the predictions for PMU measurements show that the model can work with noisy real-life data even without experiencing a severe event. The reduced order models cannot provide realistic dynamics as they are more suitable for problems like dynamic state estimation, where the prediction horizon is extremely short, and the data to learn from is massive, contrary to NODE training data. NODE frequency estimator workflow could also be designed with limited historical measurements when the mathematical model is not available whereas other data-driven approaches require extensive data for generalizing unobserved conditions. The low computational power and memory requirements due to the decentralized local training also allow operators to run the workflow constantly, where each event corresponds to a separate NODE model. This situation can create trust in AI solutions as system operators can validate the predictions for many dynamic recordings collected under different events and system conditions.

The proposed workflow in this paper has several limitations. Extreme downsampling of the measurements could reduce the quality of training data, which might cause longer training times. If the measured bus is physically far away from the source of the disturbance, the NODE might not find a suitable ODE system. Another limitation arises if the system's dynamics change after collecting the training data. The change might occur due to the controllers' actions or cascading events. Therefore, the unobserved new dynamics could hinder the NODE's prediction capabilities. Multiple NODE models in different system locations can be trained in a moving window approach to mitigate these challenges.

## 5. Conclusion

Predicting the power system frequency after severe events will become more critical for system operators to maintain frequency stability due to higher VRES share. The proposed NODE frequency estimator workflow can approximate the dynamics of the observed bus in near real-time to provide critical information for system operators using only a single measurement. The collocation-based sampling method improves the learning performance and efficiency of the NODE training. The system operators can apply the proposed workflow for advanced monitoring and corrective control actions to improve frequency security. Future research will be conducted on the NODE prediction capabilities for the frequency restoration response of the system following a frequency nadir to predict the system recovery period. Moreover, NODE predictions for different types of instabilities will be investigated.

## CRediT authorship contribution statement

**Mert Karacelebi:** Visualization, Data curation, Validation, Formal analysis, Conceptualization, Methodology, Writing – review & editing, Software, Investigation, Writing – original draft. **Jochen L. Cremer:** Writing – review & editing, Resources, Funding acquisition, Supervision, Conceptualization, Project administration.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jochen L. Cremer reports financial support was provided by Dutch Research Council. Jochen L. Cremer reports a relationship with Dutch Research Council that includes: funding grants. The manuscript contains anonymized PMU data from Dutch transmission system operator TenneT. Necessary acknowledgments are included in the manuscript and TenneT accepted the usage of data. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available upon request.

## References

[1] N.H. Shazon, A. Jawad, Frequency control challenges and potential countermeasures in future low-inertia power systems: a review, Energy Rep 8 (2022) 6191–6219, https://doi.org/10.1016/j.egyr.2022.04.063.
[2] J. Zhang, D. Chen, On the application of phasor measurement units to power system stability monitoring and analysis, 2012 IEEE Power And Energy Conference At Illinois, PECI 2012 (2012) 1–6, https://doi.org/10.1109/PECI.2012.6184607.
[3] C. Li, Y. Wu, Y. Sun, H. Zhang, Y. Liu, Y. Liu, V. Terzija, Continuous underfrequency load shedding scheme for power system adaptive frequency control, IEEE Trans. Power Syst. 35 (2) (2020) 950–961, https://doi.org/10.1109/TPWRS.2019.2943150.
[4] N.K. Rajalwal, D. Ghosh, Recent trends in integrity protection of power system: a literature review, Int. Trans. On Electr. Energy Syst. 30 (10) (2020) 1–43, https://doi.org/10.1002/2050-7038.12523.
[5] R. Huang, S. Jin, Y. Chen, R. Diao, B. Palmer, Q. Huang, Z. Huang, Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform, in: 2017 IEEE Power & Energy Society General Meeting, 2017, pp. 1–5, https://doi.org/10.1109/PESGM.2017.8274505.
[6] A.Q. Santos, R.M. Monaro, D.V. Coury, M. Oleskovicz, A new real-time multi-agent system for under frequency load shedding in a smart grid context, Electr. Power Syst. Res. 174 (2019) 105851.
[7] R. Moreno, H.R. Chamorro, R. Rye, H. Khazraj, F. Gonzalez-Longatt, V.K. Sood, W. Martinez, Online dynamic assessment of system stability using unscented kalman filter, in: 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), 2020, pp. 923–928, https://doi.org/10.1109/ISIE45063.2020.9152588.
[8] P. Dash, A. Pradhan, G. Panda, Frequency estimation of distorted power system signals using extended complex kalman filter, IEEE Trans. Power Deliv. 14 (3) (1999) 761–766, https://doi.org/10.1109/61.772312.
[9] A. Elandaloussi, A. Apostolov, A. Bidram, A. Rajapakse, C. Arbona, D. Sabin, J. Ponraj, J. Raymond, J. Blumschein, J.F. Piñeros S., M. Reno, N. Nair, R. Das, R. Fowler, S. Billaut, S. Brahma, S. Kamalasadan, V. Madani, Y. Liu, Y. Yin, Practical applications of artificial intelligence and Machine learning in power system protection and control, Technical Report September, 2023.

[10] E.M. Voumvoulakis, A.E. Gavoyiannis, N. Hatziargyriou, Application of machine learning on power system dynamic security assessment, in: 2007 International Conference on Intelligent Systems Applications to Power Systems, 2007, pp. 1–6, https://doi.org/10.1109/ISAP.2007.4441604.
[11] Y. Xu, Y. Dai, Z.Y. Dong, R. Zhang, K. Meng, Extreme learning machine-based predictor for real-time frequency stability assessment of electric power systems, Neural Comput. Appl. 22 (3–4) (2013) 501–508, https://doi.org/10.1007/s00521-011-0803-3.
[12] C. Ma, L. Wang, C. Gai, D. Yang, P. Zhang, H. Zhang, C. Li, Frequency security assessment for receiving-end system based on deep learning method, in: 2020 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia), 2020, pp. 831–836, https://doi.org/10.1109/ICPSAsia48933.2020.9208491.
[13] J.J. Yu, D.J. Hill, A.Y. Lam, J. Gu, V.O. Li, Intelligent time-adaptive transient stability assessment system, IEEE Trans. Power Syst. 33 (1) (2018) 1049–1058, https://doi.org/10.1109/TPWRS.2017.2707501.
[14] M.R. Aghamohammadi, M. Abedi, DT based intelligent predictor for out of step condition of generator by using PMU data, Int. J. Electr. Power Energy Syst. 99 (2018) 95–106, https://doi.org/10.1016/j.ijepes.2018.01.001.
[15] X. Liu, J. Xie, X. Fang, H. Yuan, B. Wang, H. Wu, J. Tan, A comparison of machine learning methods for frequency nadir estimation in power systems, in: 2022 IEEE Kansas Power and Energy Conference (KPEC), 2022, pp. 1–5, https://doi.org/10.1109/KPEC54747.2022.9814724.
[16] R. Hooshmand, M. Moazzami, Optimal design of adaptive under frequency load shedding using artificial neural networks in isolated power system, Int. J. Electr. Power Energy Syst. 42 (1) (2012) 220–228.
[17] Y. Zhang, X. Shi, H. Zhang, Y. Cao, V. Terzija, Review on deep learning applications in frequency analysis and control of modern power system, Int. J. Electr. Power Energy Syst. 136 (2022) 107744, https://doi.org/10.1016/j.ijepes.2021.107744.
[18] Y. Zhang, H. Zhu, X. Wang, Prediction for the maximum frequency deviation of post-disturbance based on the deep belief network, in: 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), 2019, pp. 683–688, https://doi.org/10.1109/ISGT-Asia.2019.8881786.
[19] O. Yurdakul, F. Eser, F. Sivrikaya, S. Albayrak, Very short-term power system frequency forecasting, IEEE. Access 8 (2020) 141234–141245, https://doi.org/10.1109/ACCESS.2020.3013165.
[20] J. Xie, W. Sun, A transfer and deep learning-based method for online frequency stability assessment and control, IEEE. Access 9 (2021) 75712–75721, https://doi.org/10.1109/ACCESS.2021.3082001.
[21] F. Bellizio, J.L. Cremer, G. Strbac, Machine-learned security assessment for changing system topologies, Int. J. Electr. Power Energy Syst. 134 (2022) 107380, https://doi.org/10.1016/j.ijepes.2021.107380.
[22] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: S. Bengio; H. Wallach; H. Larochelle; K. Grauman; N. Cesa-Bianchi, R. Garnett, (Eds.), in: Advances in Neural Information Processing Systems, vol. 31, Curran Associates, Inc., 2018.
[23] V.M.M. Alvarez, R. Roşca, C.G. Falcuţescu, DyNODE: neural ordinary differential equations for dynamics modeling in continuous control, (2020) 1–9.
[24] C. Legaard, T. Schranz, G. Schweiger, J. DrgoÅa, B. Falay, C. Gomes, A. Iosifidis, M. Abkar, P. Larsen, Constructing neural network based models for simulating dynamical Systems, ACM Computing Surveys 55 (11) (2023) 1–35, https://doi.org/10.1145/3567591.
[25] A.J. Linot, J.W. Burby, Q. Tang, P. Balaprakash, M.D. Graham, R. Maulik, Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems, J. Comput. Phys. 474 (2023) 111838, https://doi.org/10.1016/j.jcp.2022.111838.
[26] X. Kong, K. Yamashita, B. Foggo, N. Yu, Dynamic parameter estimation with physics-based neural ordinary differential equations, IEEE Power & Energy Society General Meeting (PESGM) (2022) https://doi.org/10.1109/PESGM48719.2022.9916840.
[27] T. Xiao, Y. Chen, S. Huang, T. He, H. Guan, Feasibility study of neural ode and dae modules for power system dynamic component modeling, IEEE Trans. Power Syst. 38 (3) (2023) 2666–2678, https://doi.org/10.1109/TPWRS.2022.3194570.
[28] Q. Shen, Y. Zhou, Q. Zhang, S. Maslennikov, X. Luo, P. Zhang, Physics-aware neural dynamic equivalence of power systems, IEEE Trans. Power Syst. 39 (1) (2024) 2341–2344, https://doi.org/10.1109/TPWRS.2023.3328162.
[29] M. Karacelebi, J.L. Cremer, Online neural dynamics forecasting for power system security, Int. J. Electr. Power Energy Syst. 167 (2025) 110566, https://doi.org/10.1016/j.ijepes.2025.110566, https://www.sciencedirect.com/science/article/pii/S0142061525001176.
[30] J. Zhuang, N. Dvornek, X. Li, S. Tatikonda, X. Papademetris, J. Duncan, Adaptive checkpoint adjoint method for gradient estimation in neural ODE, in: H.D. III, A. Singh, (Eds.), in: Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 11639–11649.
[31] V. Romanuke, Time series smoothing improving forecasting, Applied Computer Systems 26 (1) (2021) 60–70, https://doi.org/10.2478/acss-2021-0008.
[32] R. Cachucho, M. Meeng, U. Vespier, S. Nijssen, A. Knobbe, Mining multivariate time series with mixed sampling rates, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Association for Computing Machinery, New York, NY, USA, 2014, pp. 413–423, https://doi.org/10.1145/2632048.2632061.
[33] I.-S. Oh, J.-S. Lee, Dense sampling of time series for forecasting, IEEE. Access 10 (2022) 75571–75580, https://doi.org/10.1109/ACCESS.2022.3191668.
[34] R. Singh, N. Palmer, D. Gifford, B. Berger, Z. Bar-Joseph, Active learning for sampling in time-series experiments with application to gene expression analysis, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 832–839, https://doi.org/10.1145/1102351.1102456.
[35] E. Elhamifar, G. Sapiro, S.S. Sastry, Dissimilarity-based sparse subset selection, IEEE Transactions On Pattern Analysis And Machine Intelligence 38 (11) (2016) 2182–2197, https://doi.org/10.1109/TPAMI.2015.2511748.
[36] E. Elhamifar, M.C.D.P. Kaluza, Subset selection and summarization in sequential data, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 1036–1045.
[37] Y. Rubanova, R.T. Chen, D. Duvenaud, Latent ODEs for irregularly-sampled time series, in: Advances in Neural Information Processing Systems, vol. 32, 2019.
[38] F. Milano, Power System Modelling and Scripting, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, https://doi.org/10.1007/978-3-642-13669-6, volume 54 of Power Systems.
[39] E. Roesch, C. Rackauckas, M.P. Stumpf, Collocation based training of neural ordinary differential equations, Stat Appl Genet Mol Biol 20 (2) (2021) 37–49, https://doi.org/10.1515/sagmb-2020-0025.
[40] P. Pourbeik, WECC type 3 wind turbine generator model-phase II, Technical Report X, Electric Power Research Institute, 2014.
[41] T. Eiter, H. Mannila, Computing discrete Fréchet distance, Technical Report, Information Systems Department, Technical University of Vienna., Vienna, 1994.
[42] A.M. Stanković, A.A. Sarić, A.T. Sarić, M.K. Transtrum, Data-driven symbolic regression for identification of nonlinear dynamics in power systems, in: 2020 IEEE Power & Energy Society General Meeting (PESGM), 2020, pp. 1–5, https://doi.org/10.1109/PESGM41954.2020.9281935.
[43] DIgSILENT, PowerFactory 2022, (2022).
[44] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: a fresh approach to numerical computing, Society for Industrial and Applied Mathematics (SIAM) Review 59 (1) (2017) 65–98, https://doi.org/10.1137/141000671.