Machine learning and digital twins
monitoring and control for dynamic security in power systems

Brosinsky, Christoph; Karaçelebi, Mert; Cremer, Jochen L.

DOI
10.1016/B978-0-32-399904-5.00010-7

Publication date
2023

Document Version
Final published version

Published in
Monitoring and Control of Electrical Power Systems using Machine Learning Techniques

Machine learning and digital twins: monitoring and control for dynamic security in power systems

4

Christoph Brosinsky[a], *Mert Karaçelebi*[b], *and Jochen L. Cremer*[b]
[a]Technische Universität Ilmenau, Ilmenau, Germany, [b]Delft University of Technology, Delft, Netherlands

Indexes

i	System state
j	Control set-point

Symbols

$\hat{y}_i$	Prediction of time discrete sample i
u_j	Values of control set-point j
$\overline{x}$	Constrained differential state
x_i	Values describing the system state i
N	Horizon length (number of samples)
p	Model parameter set
y, y^m	Output signal obtained by measurements

Acronyms

AC	Alternating current
AI	Artificial Intelligence
ANN	Artificial neural network
AVR	Automatic voltage regulator
DAE	Differential algebraic equations
DSA	Dynamic security assessment
DSO	Distribution system operator
DT	Digital twin
EMS/DMS	Energy management system / Distribution management system
FPR	False positive rate
hf	high fidelity
HVAC	High voltage alternating current
ICT	Information and Communications Technology
KF	Kalman filter
lf	low fidelity
MHE	Moving horizon estimation
ML	Machine learning
MLP	Multi-layer perceptron
PE	Parameter estimation
ROC	Receiver operator characteristics
SSA	Static security assessment
TPR	True positive rate
TSO	Transmission system operator

Monitoring and Control of Electrical Power Systems Using Machine Learning Techniques
https://doi.org/10.1016/B978-0-32-399904-5.00010-7

VSC-HVDC Voltage source converter high voltage direct current transmission system
WAMS Wide area monitoring system

4.1 Introduction

This chapter connects techniques from machine learning (ML) and digital twins (DTs) to provide novel insights into monitoring and control of (dynamic) security for electrical power systems. DTs are validated and verified high-fidelity (hf) models of high simulation accuracy. They can be applied to simulate the supervised process (e.g., power system operation) and provide synthetic data whenever measurement data is scarce. However, such hf simulation models are not always appropriate for real-time applications in monitoring and control as they correspond to high computational effort. The computational effort, i.e., the time required to perform simulations can be significantly reduced by applying surrogate models, which may be mandatory for some applications executed in real-time. There, ML can create an application-specific, low-fidelity (lf) approximation of the hf digital twin. These trained lf models are promising in real-time applications, where the time to react is scarce and lf information is sufficient. This chapter aims at providing a conceptual overview of combining the advantages of hf and lf models.

The combined framework of hf DT and lf ML is illustrated in Fig. 4.1. The two models, hf DT and lf ML, collect data from the power system (measurements and topology) and from the system operator. The hf model processes and generates the required data for the lf model, where the lf model feeds predictions and relevant study

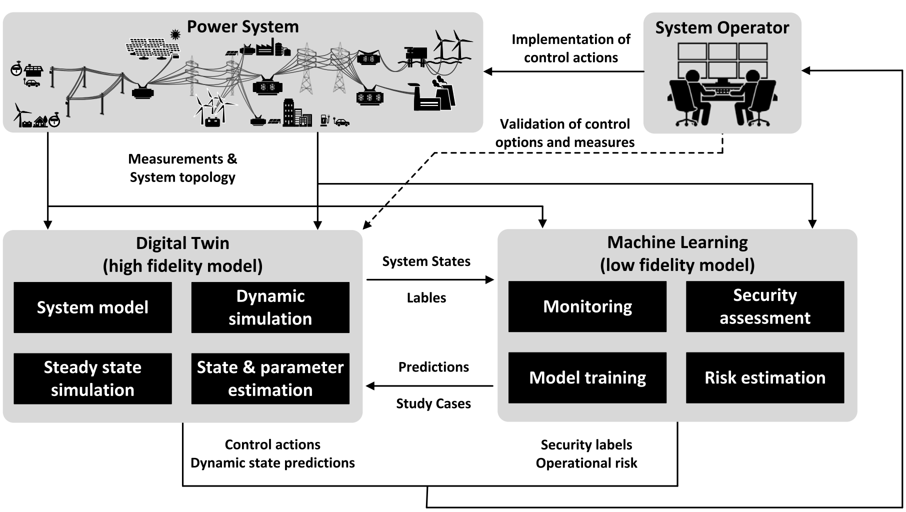

Figure 4.1 Digital twin and machine learning framework.

cases for the hf DT. Combining the two can provide helpful information for system operators to enhance operational security.

The chapter is split into two parts. The first part introduces the individual concepts from ML and DT, and the second part focuses on combining the concepts to predict dynamic security and analyze operational risks. First, the concept of hf DTs for online power system studies and their corresponding model parameter tuning from measurements is introduced. Subsequently, ML approximated lf models and their corresponding training frameworks are introduced. The first part concludes with a concept to apply data from hf models for the training of lf models.

The second part of this chapter introduces concepts for combining hf and lf models for purpose-driven real-time power system dynamic security assessment (DSA) applications. Concepts of conventional DSA are reviewed, and real-time DSA with ML methods are discussed. Finally, both hf and lf models are applied for probabilistic risk analysis in a didactic example.

4.2 Machine learning and digital twins for power system analysis

New techniques applying ML or DTs have individual strengths offering opportunities and limitations for power system analysis. Therefore the two techniques are introduced, and their advantages and disadvantages are discussed.

4.2.1 Machine learning for power system analysis

With the wide availability of measurement data and other information in electric power grids, learning models from this data with ML is the logical continuation to classical rule-based knowledge inference in power system management. The promising applications of ML techniques for power system analysis have been investigated in the last decades [1–4]. These are embedded in so-called decision support, assistant or expert systems, which comprise automatic fault diagnosis, isolation, and evaluation [5], alarm prioritization, fault switching schedules, safety checking, routine switching schedules, automatic switching, and network optimization [6,7].

Recent developments in computing hardware enable the application of these concepts combined with novel ML techniques [8]. These provide the foundation for power system engineers to augment their experience and heuristic knowledge with ML predictions [3]. However, the major limitation of ML is its generalizability, which leads to low accuracies and may make ML unsuitable. Generalization means that a trained ML model may be suitable for a specific trained data, but it may be challenging to apply it beyond that specific data. Thus in case the topology of the observed system changes or the operating conditions are different, the ML model must be retrained [9]. In addition, sometimes ambiguity exists about the ML results and their accuracy. Hence, the trust, accuracy, robustness, and reliability of ML models are of particularly high importance when developing ML-based task applications; otherwise, operators will circumvent the tools and consequently make conservative decisions to avoid security limit violations [10]. Additionally, the interpretability of some ML and the explanation of the predicted results may be difficult, e.g., with deep neural networks [11].

4.2.2 Digital twins for power system analysis

"Digital Twins (DT) are software-based abstractions of complex physical systems or objects which are connected via a communication link to the real object through a continuous data flow from the real world" [12]. Although other approaches are applicable to create a DT [13], the ideal DT model should be of analytic and deterministic character, i.e., physics-based (first principles), and sufficiently accurate while capable in real-time [14]. Thus a major challenge arises to determine the required detail-level. Whereas a simplified model does not unveil the DT value, a highly accurate approach may cause unbearable complexity and exhaustive computation times [12].

The term DT epitomizes the general trend of converging information and operational technology [15], and has been recently identified as a key concept by several industries [16]. It evolved from the broad concept of cyber-physical systems (CPS), i.e., physical systems or processes, which integrate computational applications and information communications technology (ICT). Fig. 4.2 (adapted from [17]) illustrates the difference to the predecessors of DTs. A digital model does not exchange data or information automatically. This exchange is usually done manually. A *digital shadow* is created by telemetry interrogation, but does not automatically report back to the physical world. The counterpart of a *digital shadow* is a *digital generator* [18], which can be applied to design systems (e.g., offline planning studies or compliance tests of control schemes prior to commissioning), or to train ML tools. The digital twin communicates bidirectionally and automatically reports to the physical world (closed loop). Thus the main characteristic that distinguishes a digital model and a digital twin is exchanging data automatically in a bidirectional way with the physical object.

Depending on the purpose of the application, a DT can consist of many executable models, especially when it comes to a high level of complexity, i.e., considering different timeframes or nonlinearities at runtime. These models can provide insights into the system and adapt to the system by causal (first principle) connections. Considering power system models, a high accuracy depends on a hf model. Furthermore, a high granularity in model components allows flexibility to model different power systems, topologies, operating conditions, and time-domains. From the user perspective, a high system model accuracy raises the confidence in the decisions made upon the DT model or the subsequently applied decision support functions [10]. Taking power system models as an example, increasing the level of detail in terms of considered time frames, such as models, would comprise a steady state representation, a phasor or average value model representation for electromechanical interactions, and a model including all necessary components to reflect the electromagnetic timeframes up to the microsecond scale. A suitable metamodel or interface is required to describe these interrelated dependencies to avoid redundancy of the model causalities and model parameters. This seamless integration requires a modular design approach, and suitable a framework for consistent and flexible modeling.

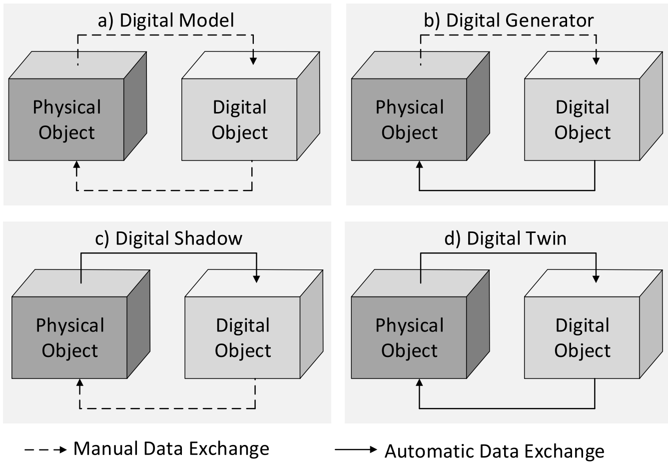

Figure 4.2 Difference between digital model (a), digital generator (b), digital shadow (c), and digital twin (d), adapted from [17].

Limiting factors to the DT approach include high demands in the computation hardware, i.e., the high computation effort respecting the real-time requirements in operational environments (e.g., online DSA for large systems). In particular, the execution of multiple simulations to find an optimal operating state requires long computing times. Another limiting factor is the expertise required from dedicated power system engineers and ICT specialists to create a real-time capable, high-fidelity DT. Tackling these challenges is promising, as DTs in operational systems enable novel automation routines, and automated system control functions up to autonomous systems [19].

4.2.3 Application of digital twins for online power system studies

A DT can estimate a state value, which is not directly observable by sensors, and report this to the real world. This fundamental property of a DT appears to be a similarity to the concept of an observer, which is often applied in control engineering (see Fig. 4.3). According to the control theory, observers apply a mathematical model of the original system and a function of the measurement noise (i.e., a correction term) to estimate the system state from measurements [20]. As an ideal, DT represents the case of a high-fidelity model without uncertainties, in practical application the estimated state is expected to converge to the true state of the observed entity. These properties of the DT concept are illustrated in Fig. 4.3. Depending on the applied DT model, the input signal u_j, the measured output y_j, and the corresponding model response $\hat{y}_j$ is generic and may comprise a single or an array of values. When the model parameter set p is unknown, a suitable model parameter identification method is required to ob-

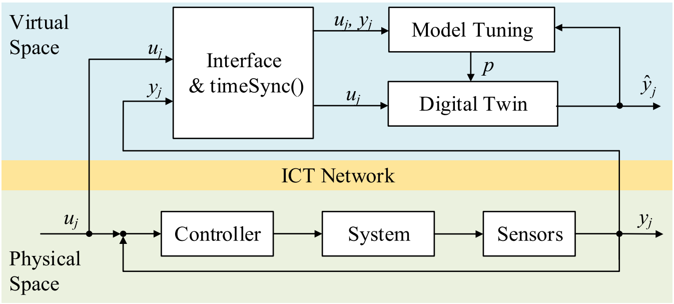

Figure 4.3 Flowchart illustration of the creation or training of a generic digital twin.

tain a valid model response (see subsection on parameter estimation for continuous digital twin model tuning).

Apparent fields of application for power system DTs include:

- Continuous system analysis and anomaly detection (e.g., malfunctions or cyber-attacks)
- Enhanced predictability of future operating conditions, allowing coordinated system operation among distribution system operators (DSOs) and transmission system operators (TSOs)
- Increased power system observability
- Consideration of system dynamics in network operation to prove control actions
- Reduction of unplanned outages by continuous equipment monitoring
- Provision of a realistic operator training simulator
- Implementation of a higher degree automation schemes and assistant systems for power system operation

As illustrated in Fig. 4.4, the study model is derived from a steady state real-time snapshot by conventional state estimation. Thus today's power system control applications do not provide a validated time-domain model. Furthermore, the modeling results derived by screening security-related scenarios, and contingencies are not yet stored in conventional operational environments (e.g., EMS/DMS long term archives). An orderly archiving can help to apply ML methodologies to evaluate critical system states faster and to analyze subordinate mechanisms better. Consequently, DTs provide the basis for advanced operator assistance systems and decision support functions for online system operation.

Fig. 4.4 illustrates the state of the art of modern EMS/DMS environments. The real-time view on the left side of Fig. 4.4 comprises the SCADA, and sometimes WAMS data, applied to derive a steady-state estimation of the power system state. A snapshot from the actual state estimation result is applied by subsequent analytic functions (e.g., contingency analysis, security assessment, or redispatch). These functionalities depend on the scenario-based models illustrated on the right side of Fig. 4.4. Today, the model result and the observed dynamic system response are not considered for model validation. The proposed DT concept, as illustrated in Fig. 4.3, comprises a validated model to close this applicational gap.

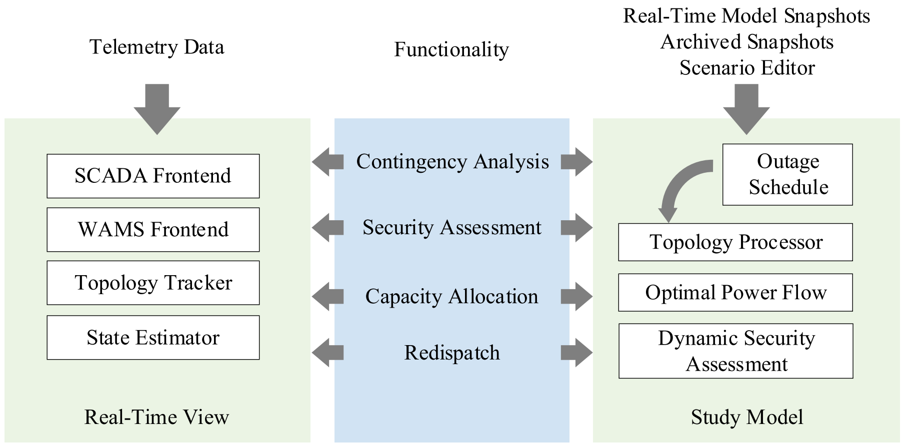

Figure 4.4 Differences between real-time model and study model.

4.2.4 Combining machine learning and digital twins for power system analysis

When combining novel ML and DT concepts, their strength can be fully unfolded for system operation. A DT is a detailed replication (twin) of the real, physical system and can simulate the system in real-time with a high level of accuracy. The strength of DT is to simulate the system accurately. Taking the system state x_i (obtained from sensor measurements) and control set-points u_j as an input, future system states x_{i+1}, x_{i+2}, $\ldots$, x_{i+n} are predictable. A limiting factor is the computational time, which depends on the amount of data, the system size, and the dimensionality of investigated study cases.

In contrast, the strength of ML is a quick prediction at the cost of a lower accuracy score (e.g., resulting from training biases). ML methods train a model from data without prior knowledge of possible model parametrizations. Therefore one concept to combine the two individual strengths by using the DT to generate training data (when computational time is abundant), and then use the data to train the ML model. The advantage is that a DT can generate data, which is rarely observable, and which can only be acquired with great effort, or is unobtainable in the real world. Thus combining the strengths of ML and the DT concept can support real-time power system analysis, whereas ML predictions are fast, but of lower accuracy, DTs can correct the ML prediction.

Furthermore, ML predictions can be proven or validated by the DT model considering only a few in-depth studies and suggested control-set points. One possible application combining ML and DT is a power system perturbation, e.g., caused by an equipment fault. Then, the ML can immediately predict the power system's dynamical response. As illustrated in Fig. 4.5, ML can suggest ad hoc control responses after the occurrence of the fault at time t_1. Subsequently, the proposed primary control actions can be taken considering the time delay between sending the control signal and the pri-

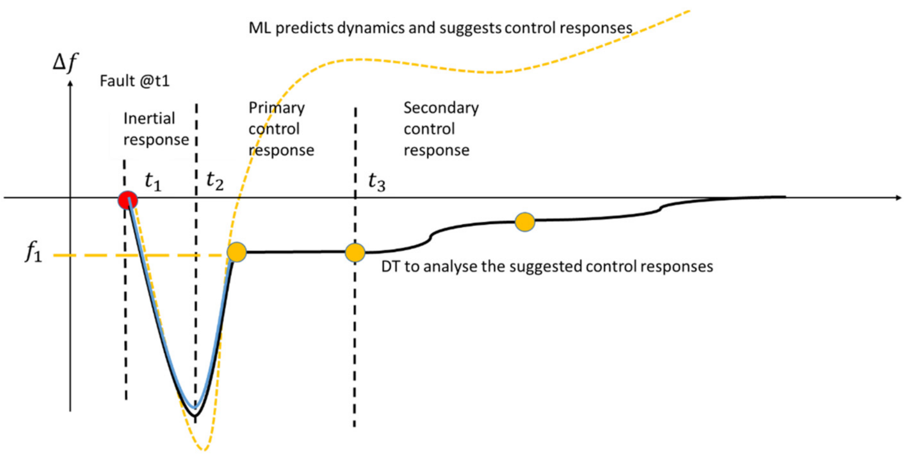

Figure 4.5 Combined benefits from ML and DT for accurate and fast control responses to faults. ML predicts immediately at fault t_1 the possible dynamical trajectory and suggests a control action at t_2. Between time t_1 and t_2, the DT simulates the control action in detail and confirms the suitability of the action (modified from [21]). *Black line* (DT) is available at t_2, *yellow line* (ML) is available at t_1. *Blue line* is the actual system response.

mary control actions at time t_2. In the meantime, the DT can start to execute a detailed dynamic simulation with higher accuracy. When the hf DT simulation is available, it can replace the lf ML prediction to improve the system stability and take secondary control actions. The ML ability to provide a rough prediction of the initial dynamic trajectory very fast, in combination with the advantage of DT to simulate the dynamic trajectory accurately to prove the suggested control action in a hf model, raises the confidence in the control action and maximizes the stabilizing effect of the response to the fault.

The capability of the power system to anticipate, absorb, adapt, or recover from faults (and other events), comprises the intra-day operation (handling of disruptions in real-time operation) and the operational planning (optimized topology and structural reinforcements). Combining the benefits at their full potential, DT and ML support the development of novel control schemes. Applying control schemes that are both corrective and preventive enables operational planning closer to the power system security boundary, i.e., allowing a higher asset utilization. The control schemes derived from the combination of ML and DT have the potential to be fast and sophisticatedly robust at the same time. They may even provide decision support for cascading failures by successive and systematic assessments. As shown in Fig. 4.5, following a disturbance at t_1, the ML predicts the system dynamics (yellow dashed line) and the following control action (red points), while in the meantime the DT computes the dynamic trajectory (black line), to correct the control action according to the observed system state (blue line) towards a full system recovery in Fig. 4.6.

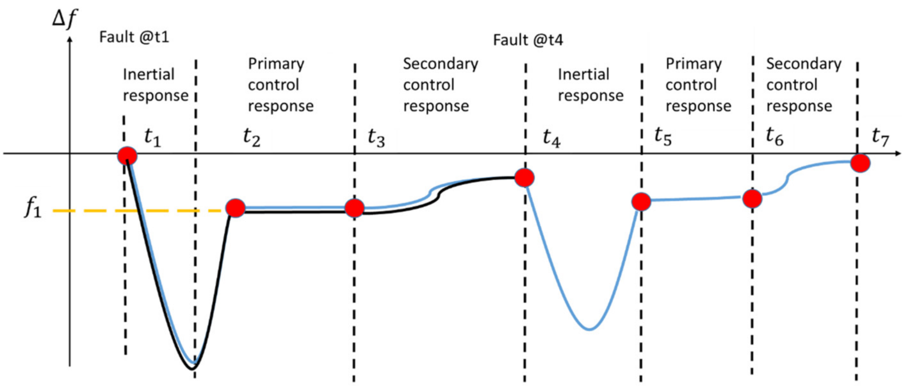

Figure 4.6 The system response with the implemented control actions. Computing control actions in nearly real-time has the potential to even consider cascading failures in real-time (modified from [21]).

4.2.4.1 Parameter estimation for continuous digital twin model tuning

As an accurate and validated high-fidelity dynamic power system model is required, model validation and parameter estimation methods are necessary. Parametric time-domain models contain internal parameters, which describe the physical behavior of the modeled object or control system. These parameters are often unknown and need to be estimated. This is typically done by fitting the mathematical models to measured time series data. Some common techniques for parameter estimation from online measurements are the Kalman filter (KF) and its derivatives (e.g., extended KF, unscented KF, or cubature KF) [22].

Another suitable method is the moving horizon estimation (MHE) [23]. The MHE approach can handle constraints, which is essential to raise the accuracy and validity of the estimated states and parameters of nonlinear system models. The general concept of the moving horizon estimation (MHE) methodology is illustrated in Fig. 4.7. Instead of applying the full information available as input, it only takes a sample of N values. Thus the MHE approach combines the advantages of full information estimation with a tractable online computation [23]. The MHE's computational power is as demanding as the KF and its derivatives. However, in comparison, it can deal with highly nonlinear estimation problems in the presence of state constraints $\overline{x}$. The general form to describe an MHE is given in [23], whereas methods for implementation are discussed in [24].

The fundamental concept of the MHE is to not fit the parameters to the complete data available at time k, i.e., $0, 1, \ldots, k$, but rather to only the previous n time steps, i.e., $k - N; k - N + 1, \ldots, k$. This concept is illustrated in Fig. 4.7. As the time frame moves along the captured time series, the objective function as given in [23], is minimized to estimate the parameters that fit the model response to the captured values. Starting from an initial parameter set, the model parameters are varied in such a way that the objective function reaches a minimum. At each time step k, an optimization problem is solved to minimize the objective function. It penalizes the deviation of the

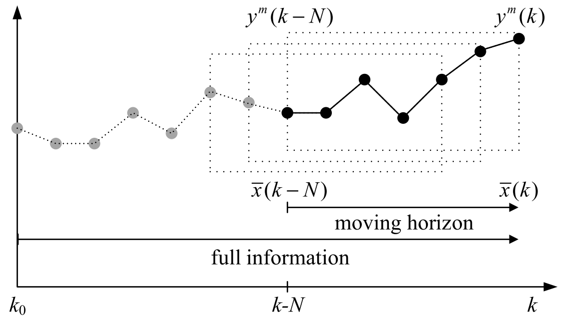

Figure 4.7 Parameter estimation with MHE scheme (adapted from [23]).

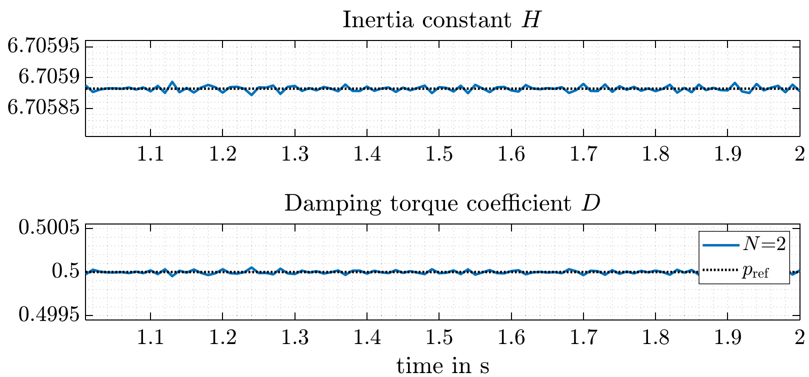

Figure 4.8 MHE example with horizon length $N = 2$.

computed output y from the measured output y^{m}, and the deviations from the system dynamics. The state and parameter guess is based on the previous time step (as illustrated in Fig. 4.7). An example illustrating a MHE parameter estimation result with $N = 2$ for the synchronous machine inertia constant H, and the damping coefficient D is illustrated in Fig. 4.8.

A simplified graphical representation of an analytical time-domain model of a sixth-order synchronous machine model, based on the equivalent circuit, is given in Fig. 4.9. The model comprises a dynamic state vector x, including a generic representation of its controllers (see also Fig. 4.9), and state names can be found in Table 4.1. In the study model applied for the didactic example (see section 4.5), the excitation system IEEE AC4A and a simple speed governor IEEE HyGOV are applied. These internal states, which are hard to obtain by sensors and are not observable from a power system control room, add valuable information for the ML training procedure, as this information can be related to system stability.

4.2.4.2 Training the ML model from data

The supervised ML technique can derive purpose-driven surrogate models from data. These ML models are applicable for classification and regression, and take the feature

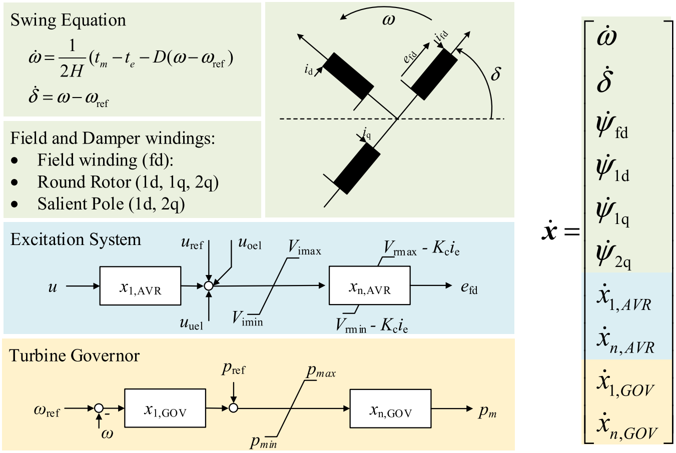

Figure 4.9 Simplified analytical model and the dynamic state vector of a synchronous machine model including governing and excitation system.

Table 4.1 Dynamic system states corresponding to Fig. 4.9.

Symbol	System state
δ	Rotor angle (rad)
ω	Angular frequency (rad s^{-1})
t_e	Electrical torque (p.u.)
t_m	Mechanical torque (p.u.)
ψ_{fd}	Excitation flux (p.u.)
ψ_{1d}	Flux in 1d-axis damper winding (p.u.)
ψ_{1q}	Flux in 1q-axis damper winding (p.u.)
ψ_{2q}	Flux in 2q-axis damper winding (p.u.)
u	Control set point
x_{AVR}	Automatic voltage regulator state
x_{GOV}	Governor state

vector x_i as an input to apply the trained function

$$\hat{y}_i = \hat{f}(x_i)$$

for predicting the output data $\hat{y}_i$. While for the purpose of classification, the function $\hat{f}$ predicts discrete values, the values for regression are continuous $\in R$.

To train the function $\hat{f}$, a data set of Ω value pairs (x, y) is required. One sample corresponds to the pair $(x_i, \ y_i)$. As shown in Fig. 4.10, the data should be preprocessed before training the model. This includes the imputation of missing data, and normalization i.e., the data within each feature's vector is scaled, so each feature is between [0, 1], as the features may have different dimensions. This can be done by

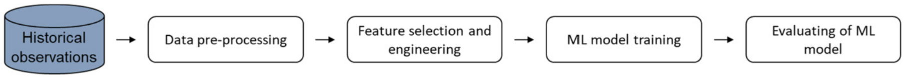

Figure 4.10 Framework to train a ML model.

applying

$$x_{ij}^{scaled} = \frac{x_{ij} - \min\left(x_j\right)}{\max\left(x_j\right) - \min\left(x_j\right)},$$

where $\min\left(x_j\right) = \min\left(x_{0j}, x_{1j}, \ldots, x_{\Omega j}\right)$ and $\max\left(x_j\right)$ denote the smallest and largest value among all data withing Ω, considering each respective feature j. The following expression assumes a normalized x and drops the superscript "*scaled*." Subsequently, the typical objective of the training procedure is to minimize the training loss, i.e., the squared difference between predicted and actual labels

$$loss = \frac{1}{\Omega} \sum_i \left(y_i - \hat{f}(x_i)\right)^2.$$

The function $\hat{f}$ is a suitable candidate to represent the data when this loss is low. However, in order to infer the suitability of this function, it is important to consider that the function may overfit or underfit the data and that the initial candidate parameterizations for $\hat{f}$ must be selected.

For instance, the engineer may decide on an initial way to parameterize the function $\hat{f}$, and the decision could be to select a polynomial function,

$$\hat{f} = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_n x^n,$$

which has n terms with n parameters to fit. In case, that n is suboptimally selected, the function tends to under- and over fitting. An exhaustive overview on selecting appropriate loss functions, including the parameterization of function $\hat{f}$, and the consideration of feature selection, is given in [25]**.** Different algorithms are applicable to minimize the loss function . For instance, an iterative algorithm is the gradient descent. Assuming $n = 1$, the function $\hat{f} = \theta_0 + \theta_1 x^1$ has only two parameters. Then the loss function becomes

$$loss = \frac{1}{\Omega} \sum_i \left(y_i - (\theta_0 + \theta_1 x_i^1)\right)^2,$$

and the partial derivatives with respect to θ_1 and θ_0 are

$$D_{\theta_1} = \frac{-2}{\Omega} \sum_i x_i^1 \left(y_i - \left(\theta_0 + \theta_1 x_i^1\right)\right),$$

$$D_{\theta_0} = \frac{-2}{\Omega} \sum_i y_i - \left(\theta_0 + \theta_1 x_i^1\right).$$

The gradient descent algorithm updates the parameters accordingly to these gradients:

$$\theta_k = \theta_k - LD_{\theta_k} \quad \forall k = \{0, 1\}.$$

These updated parameters result in an updated candidate function $\hat{f}$, and the loss of this candidate function can be computed. This procedure to update the parameters is repeated iteratively, until the termination criterion is met. An effective design of ML training and the performance of the trained ML model involves engineering and selecting the most valuable features and the objective to evaluate and train an effective model. This feature engineering transforms the data $x^{engineered} = g(x)$ into a more meaningful representation. For instance, the data may comprise the current x_{ic} and the voltage x_{iv} and the objective is to predict overheating of a device. So an effective technical feature here can be the power, since it correlates strongly with overheating, hence the engineered feature may be $x^{engineered} = x_{ic}\ x_{iv}$. Feature selection can also enhance the performances and can be based on statistical methods. Many approaches exist to select features, and their approach can be either to forward or backward select features. In backward selection, the ML model is trained with all features, and then the least-relevant feature is removed to retrain the ML model. This iteration repeats until the ML model does not improve the selected performance metrics. The forward selection starts with only randomly selected features and adds one feature in each iteration until the performance metrics for the ML model do not improve anymore.

The approximation capability of ML is determined by the structure of function $\hat{f}$. To illustrate this by an example, a model based on an artificial neural network (ANN) is described. The ANN constructs the function $\hat{f}$ out of many chained functions inspired by neurons in the human brain. This results in a multi-layer perceptron (MLP), i.e., a feed-forward type network with a unidirectional information flow. MLP architectures start with features x_i at the input layer, and end at the output layer which contains the model prediction $\hat{y}_i$. An example MLP is illustrated in Fig. 4.11. There, neurons are located in the hidden layers between input and output layers. Each neuron contains a weight vector w and a corresponding scalar bias b. The neurons act like a linear map $g(X) = W^T X + B$. However, linear functions can only construct a linear relationship between features, and the activation functions generate the non-linear interactions. The activation functions σ are applied to the resulting output of each neuron. Popular activation functions are sigmoid, rectified linear unit (ReLU) or hyperbolic tangent.

Assume a neural network $\hat{f}(x)$ approximates an unknown function $f(x) = y$ applying 6 neurons. These are distributed equally to 2 hidden layers with an activation function σ, where $x, y \in R$. The prediction $\hat{y}$ then results in

$$\hat{f}(x) = \hat{y} = w_y(\sigma\left(W_2^T \sigma\left(W_1^T x + b_1\right) + b_2\right)) + b_y.$$

These neuron weights and biases are optimized during the training. The most basic parameter update algorithm is the gradient descent algorithm applied to the predefined loss function. Many variants and improvements (e.g., Momentum, RMSProp, Adam, etc.) to gradient descent were proposed, such as stochastic gradient descent [26].

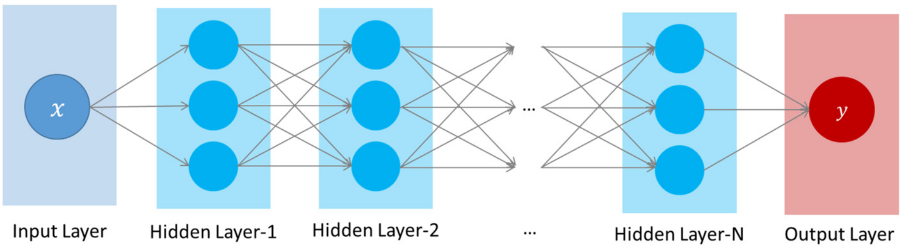

Figure 4.11 Example architecture of a multi-layer perceptron.

There are many other metrics to evaluate the training of the ML model beyond the mean squared difference. For instance, when the ML task is binary classification ($y_i = \{0, 1\}$) and not regression, the error may be suitable, which refers to the ratio of inaccurately predicted values

$$error = \frac{1}{\Omega} \sum_i | \, y_i - \hat{y}_i \, | \, .$$

On the other hand, the error might not be suitable for classification problems with an unbalanced dataset. For example, historical data regarding a power system security assessment problem usually contains more secure cases over insecure cases. The precision metric calculates the ratio of correctly predicted actual values ($\hat{y}_i = y_i = 1$) over all true predictions ($\hat{y}_i = 1$). Similarly, the recall metric or the true positive rate (TPR) calculates the ratio of correctly predicted true values over all true values $y_i = 1$. The metric F_1 calculates the weighted average of precision and recall.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

The computational complexity of the ML training depends on the selected model complexity and size of the training data. The overall training time is in the order of the computational time of hf DT simulations, but this process can be done offline, without time limitations. However, the prediction speed is very fast and almost constant, as the trained ANN model predicts the output with simple algebraic operations.

Besides the model and metrics described in this section, many more ML models can be applied in combination with DTs.

4.2.4.3 Generating ML training data by digital twin models

The training of ML models is typically performed offline when a DT of the power system examines possible operating scenarios an related data, which then can be used to train the ML model. The relevant data typically involves an operating condition as input feature, and a possible output label, which the ML model later in during aims to predict, where the output labels could be control actions, a fault analysis, or a classification of an expected risk-level. The data archive can contain an initial database of the input and output pairs of historical observations. However, a DT can generate data

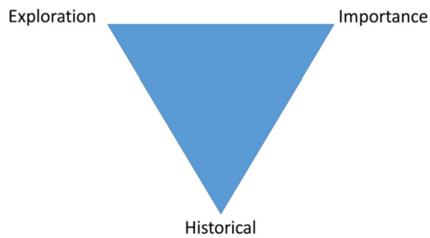

Figure 4.12 The three dimensions of an ideal ML model training database.

beyond these observed and recorded historical system conditions. Thus, it can enrich the database to enhance generalization capability by extrapolating important data samples and examine specific scenarios of interest. A suitable approach to sample the data from a DT balances (i) the data importance with (ii) exploration needs, and (iii) with data similar to historical observations. Therefore the ideal training database combines all three dimensions as shown in Fig. 4.12**.**

The idea behind the generation of data similar to historical observations is that these consider operating conditions most likely to occur. To generate such similar data, typically, probability distributions are fitted to the historical data, then Monte Carlo sampling techniques are applied to sample data that follow this fitted probability distribution [27]**.** Suitable probability distributions need to capture the complex dependency for many variables, which can be nonlinear. Therefore, copula models can be applied to model such dependencies [28]**.** For example the canonical vines (C-vines) copula models, are determined by the following probability density function

$$f\left(x_1, \ldots, x_m\right) = \left(\prod_{i=1}^{m} f_i\left(x_i\right)\right) \times c_{1\ldots m}\left(F_1(x_1, \ldots, F_m(x_m)\right),$$

$c_{1\ldots d} : [0,1]^m \to \mathbb{R}$ is an m-dimensional copula, and $U = \{U_1, U_2, \ldots, U_m\} = \{F_1\left(X_1\right), F_2\left(X_2\right), \ldots, F_m\left(X_m\right)\}$ are uniform marginal distributions. Then, the product of its marginal distribution and the multivariate copula density functions builds the joint density function.

The Monte Carlo sampling method can be applied to sample up the historical data using C-vine copulas. Generated conditions define the input space (x_i) for the DT simulation. Subsequently, the DT emulates the system behavior against the predefined disturbance scenarios with new operating conditions to obtain the desired labels (y_i) for ML training.

Obtaining an information-rich knowledge base is essential for ML applications. Therefore, additional samples can be obtained from certain samples with higher information value to improve the prediction performance. For instance, samples close to the decision boundary in a classification problem provide more information, and thus can improve the prediction performance. In power systems security assessment, the decision boundary corresponds to the system security boundary, which is illustrated in Fig. 4.13, with respect to two generators' power output. However, often the majority of cases in Fig. 4.13(a) [29] share the same class with their direct neighbors so their contributing information to training is low. However, Fig. 4.13(b) shows also cases located near the security boundary, which are more relevant, and the generation of these new samples from the interested region improves the prediction performance of ML models.

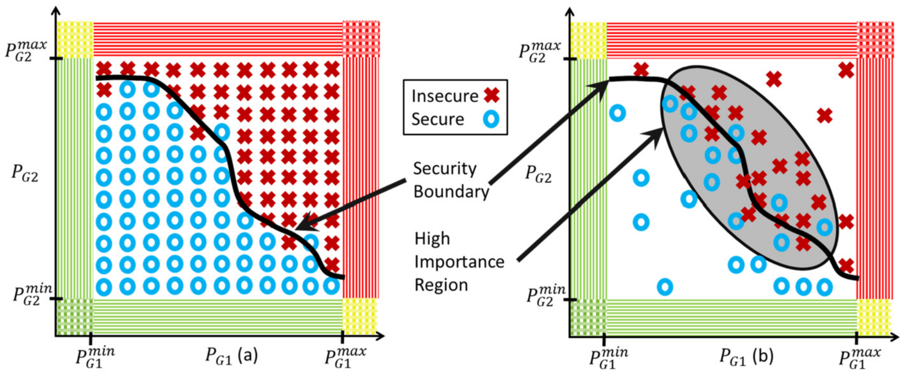

Figure 4.13 Uniform random sampling (a) and importance sampling in (b), details in [29], [30].

One method to obtain influential samples from the original dataset is importance sampling, as shown in Fig. 4.14. The sample importance is measured by the trained classifiers to filter out the points with low information value. The DT conducts dynamic simulations for the reduced dataset, hence the overall simulation dimensionality, i.e., the computational time is reduced without compromising the ML performance. Following this procedure, simulated samples are added to the training database, aiming for a higher prediction or classification accuracy.

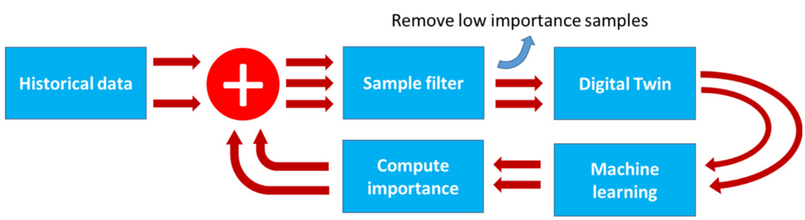

Figure 4.14 Example of an importance sampler for DSA applications.

This importance of the samples can be computed using ensemble learning methods, such as Adaboost, extreme trees, or random forests.

The ensemble models construct multiple learners $k \in \Omega^E$. The prediction score $s(x_i)$ is computed from the hypotheses of the individual learners $h_k(x_i) \in \{0, 1\}$ for sample x_i as

$$s(x_i) = \frac{1}{|\Omega^E|} \sum_{k \in \Omega^E} h_k(x_i).$$

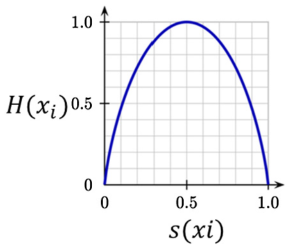

Figure 4.15 Example plot of the entropy vs. score.

The sample importance $I\left(x_i\right)$ is calculated as the measurement of disagreement, expressed as the entropy, as illustrated in Fig. 4.15, by the following formula:

$$I\left(x_i\right) = -\frac{1}{\log_2 2}\left[s\left(x_i\right)\log_2 s\left(x_i\right) + \left(1 - s\left(x_i\right)\right)\log_2\left(1 - s\left(x_i\right)\right)\right].$$

4.2.4.4 *Summary of advantages and disadvantages considering the interaction of digital twins and ML techniques*

Table 4.2 summarizes the advantages and disadvantages of traditional modeling and seeking solutions with assistance by ML methods. When it is possible to formulate the relationships between the physical variables using mathematical equations, building a coherent hf DT simulation model of reality has many advantages. On the other hand, a well-trained neural network can also find correlations that are difficult to investigate by conventional modeling techniques. Thus to combine both techniques is promising for decision support systems, where both accuracy and prediction speed are required.

Table 4.2 Comparison of model-based and data driven methodologies.

	Digital Twin (hf model)	**Machine Learning (lf model)**
Advantages	A simulation model is explainable, extendable, and reproducible by experts Verified models are applicable outside explicitly tested areas High trust in results	Detailed knowledge of the modeled system or process is not required ML has the potential to recognize pattern or correlations in data that are not perceptible by a human Fast execution times
Disadvantage	The creation of validated models is time-consuming and requires highly skilled domain experts System states that do not correspond to the normal operating condition are hardly reproducible with reasonable effort High execution time	To obtain high quality training data in the required quantity of data is often difficult The model can lead to inaccurate decisions The solution approach of a neural networks is not traceable by humans and the implementation of solutions require human supervision Trust in results not always given

4.3 Purpose driven surrogate models

Following the previous section, where the functionality and possible interactions between hf DT and lf ML models have been described, this section gives an example for real-time power system dynamic security applications. Therefore, the previously discussed concepts are utilized to obtain a purpose-driven surrogate model for DSA applications.

4.3.1 Dynamic security assessment with digital twins

DSA studies of electric power systems are mainly based on time-domain simulations. These mathematical models often contain stiff or moderately stiff nonlinear differential algebraic equation (DAE) systems of high order. The solution of these systems of equations involves a relatively high computational effort. Additionally, the applicable numerical methods can only approximate the system dynamics. Consequently, a validation procedure involving real process data is required. One major obstacle for meaningful DSA implementations in practice is to gather enough data for such validation, especially for security-related situations (e.g., critical contingencies, dynamic security boundary violation, severe disturbances). Thus, two main obstacles for online DSA can be identified:

- Obstacle 1: DAE solver-based solutions for screening of many scenarios in large power system models must be real-time capable to support online operation.
- Obstacle 2: Validation data sets for security-related conditions are lacking (e.g., suitable disturbance records).

4.3.1.1 Security assessment

Following the definition of security given in [31,32], security assessment tools evaluate the ability of the power system to survive perturbations and to work reliable under adverse conditions.

To analyze power system security, either static or dynamic security assessment tools are applied [37]. Static security assessment (SSA) tools evaluate post-disturbance conditions, i.e., violation equipment ratings or voltage constraints based on power flow calculations. Thus, the dynamic transition from pre- to post-contingency is not considered. Dynamic security assessment (DSA) methodologies also evaluate the phase of dynamic transition. When the system returns to a secure steady state after the disturbance, the system is considered as secure. As the conventional concept of SSA, cannot sufficiently guarantee the system integrity in all situations [38], it is mandatory for TSOs to assess dynamic stability to identify stability limits [39]. The criteria applied by DSA systems to evaluate contingencies and their impact on system security include rotor angle stability, voltage stability, and frequency stability; as well as the damping of oscillatory modes within the interconnected power system [33,34]. They must comply with boundary definitions from grid codes or other specified thresholds. Some indices considered in DSA applications for transmission systems are given in [35,36].

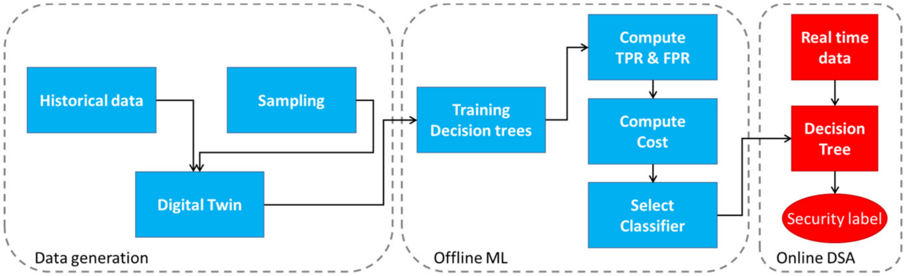

Figure 4.16 Approach of using ML for real-time DSA.

We furthermore distinguish between offline and online DSA. Though offline DSA is often part of power system planning studies, these offline studies inherit several uncertainties, as these do not consider all possible operational conditions [40]. These uncertainties can be eliminated by implementing online DSA. By including real-time process information, i.e., the actual point of operation and previously unexpected operational conditions can be considered. As online DSA can support decisions during real-time operation, the ENTSO-E recommends a continuous validation and fine tuning of system models (e.g., by WAMS) to avoid invalid results and, in the worst case, wrong decisions [38]. As the system modeling requirements vary for DSA studies, these need to be balanced in terms of model detail and required computation time [38]. This is one of the main arguments in favor of using a digital twin, as DT models are suitable for both online and offline DSA studies.

4.3.2 Real-time DSA with ML

Real-time dynamic security assessment (DSA) for system operation becomes important in the future as power system operation is done closer to operating limits. However, online DSA has its limits for real-time operation. One idea for an approach to real-time DSA is to apply a trained ML model to predict the outcome of the simulation.

Offline training requires an extensive dataset, which is generated by DT, as described in the section entitled "Generating ML training data by digital twin models." There, the operating condition x_i is defined by active and reactive power at the generator node , and voltage magnitude and phase angles given by PMUs. Subsequently, the security of the operating condition is predicted with the trained ML model for when the contingency c occurs. As shown in Fig. 4.16, the training of the ML model includes various $x \times c$ combinations within the function $f(x)$ to predict security label $\hat{y}$.

The decision tree shown in Fig. 4.16 is a common, nonparametric ML model. The model name "decision tree" comes from the upside tree structure that has a single root node, many terminals (leaf), and internal nodes. Starting from the root node, the training data is split into 2 subsets based on the decision rule that generates the maximum homogeneity between subsets. A decision rule is a simple comparison of selected fea-

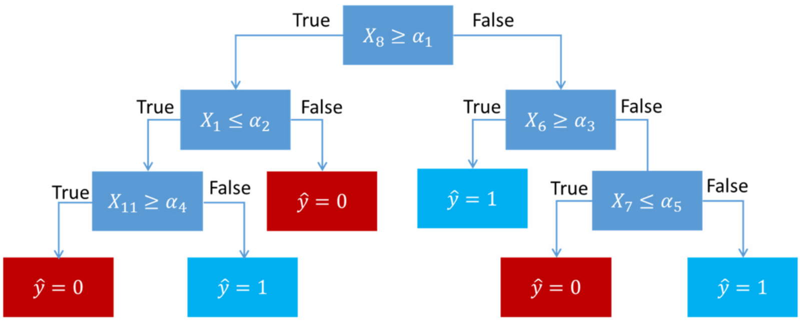

Figure 4.17 Decision tree structure for binary classification.

ture X_i to the optimized threshold value α_i. The homogeneity is measured by the impurity functions as an value of entropy or the Gini impurity. The Gini impurity $H(\theta_k)$ is formulated from the data proportion (p_{km}) of class$m$ and θ_k is the subset at node k as

$$p_{km} = \frac{1}{N_k} \sum_{y \in \theta_k} 1(y = m)$$

$$\text{Gini} : H(\theta_k) = \sum_{m} p_{km}(1 - p_{km}).$$

The decision tree represents an approximation of a high-dimensional piecewise constant function. Therefore the decision tree can define nonlinear interactions between features. Besides the predicted class, the prediction score $s(x) \in [0, 1]$ can be retrieved from the decision tree, which is equal to the ratio of the same class training samples over all samples in the prediction leaf. The binary classification tree model predicts the class (0–1) by comparing the output score $s(x_i)$ with the decision threshold Z. For the security assessment problem, the operation condition x_i is classified as secure ($\hat{f}(x_i) = \hat{y}_i = 1$) when the corresponding prediction score $s(x)$ outperforms Z, i.e., $s(x) \geq Z$. The default threshold value $Z = 0.5$ is suitable for ideal conditions, where there is no imbalance between classes in the training database considering equal costs for misclassification errors.

Furthermore, decision trees can handle high-dimensional feature spaces, as irrelevant (low information value) features are excluded automatically in the tree learning process. A decision tree structure for binary classification is illustrated in Fig. 4.17.

As large scale power system security problems require high-dimensional operation space, decision trees are suitable to address these. Many applications of decision trees for power system security problems can be found in the literature, such as the prediction of stability limits [41], the identification of preventive and corrective actions against contingencies [42], or security assessment [43]. System operators can obtain additional information from the tree path of prediction, because the decision tree is an interpretable white box model [44]. An other advantage of decision trees and other ML models in decision making processes is that these do not raise computational costs (between $O(n)$–$O(1)$). As a DT solves high-order nonlinear differential algebraic equations, the computation effort here is expected to be higher. Thus both applications can be executed on the same computation platform.

Online DSA can have critical impact on secure power system operation. While false alarms (false negative) of the DSA can trigger unnecessary or undesired control actions, any missed alarm (false positive) can jeopardize the secure power system operation. These two types of errors in the assessment process impose different follow-up costs for system operators, e.g., cost of missed alarms C_{FP} and cost of false alarms C_{FN}. Undetected insecure cases can cause a power outage that have severe effects on society, industry, and service facilities, hence the alarm costs must comply to $C_{FP} >> C_{FN}$. However, during the training stage of ML models, the main objective is minimizing the misclassification error or the other commonly used classification metrics that were introduced in the section entitled "Training the ML model from data." However, without considering the error costs, ML models cannot prioritize a specific class over the others. To approach this problem the receiver operating characteristics (ROC) curve is used; the latter is a visual metric that shows how true positive rate $TPR = \frac{TP}{\pi_+}$ and false positive rate $FPR = \frac{FP}{\pi_+}$ evolves with respect to Z, where π_+ corresponds to positive samples in the dataset. Fig. 4.18 shows the sensitivity of Z on the ROC of a decision tree model.

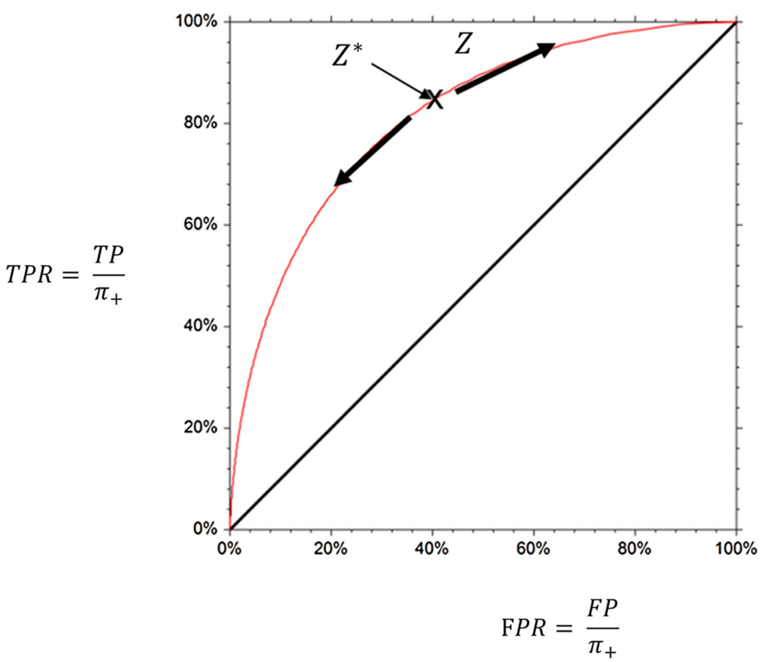

Figure 4.18 ROC and decision boundary.

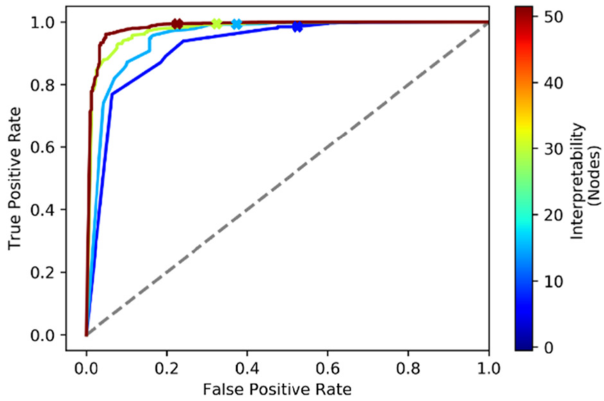

Figure 4.19 Selection with the ROC curve for IEEE 68 bus system [43].

Model predictions can be altered by changing the value of Z. For example, selecting a higher threshold reduces the number of secure predictions, which can reduce the overall model performance, but reduces the number of missed alarms. Also, the cost-optimum threshold value Z^* on the ROC curve is determined by the ratio of insecure cases π_- and the predefined error costs C_{FP} and C_{FN}. This cost optimal point is calculated by

$$Z^* = \frac{\pi_- C_{FP}}{\pi_- C_{FP} + \pi_+ C_{FN}}.$$

Power systems have a dynamic structure, where cost of the contingency (cost of load loss) and probability of the contingency are parameters that can change in real-time [45]. Fortunately, these parameters can be obtained and updated online to calculate the new optimal threshold value, without retraining the ML model.

The ROC curve can be further used to analyze several decision tree models with the optimal threshold. Although decision trees are trained with the same training data, their interpretabilities are different, because parameters such as maximum depth have been varied between models.

The ROC curves of four separate decision tree models for a case study conducted with the IEEE 68 bus test system are illustrated in Fig. 4.19. Here, the dots on the curve show the optimal cost point Z^*. As the tree grows interpretability, the capabilities are lower. The power system nodes are represented by numbers within the color bar. Based on such visual graphs, system operators can select the best classifier that suits the operation [43].

4.3.3 *Probabilistic and risk analysis with DT and ML*

Most power system disturbances are unforeseeable and highly unpredictable, but their impact can be modeled with scenario-based probabilistic risk analysis. A probabilistic approach to security assessment is illustrated in Fig. 4.20, where the approach

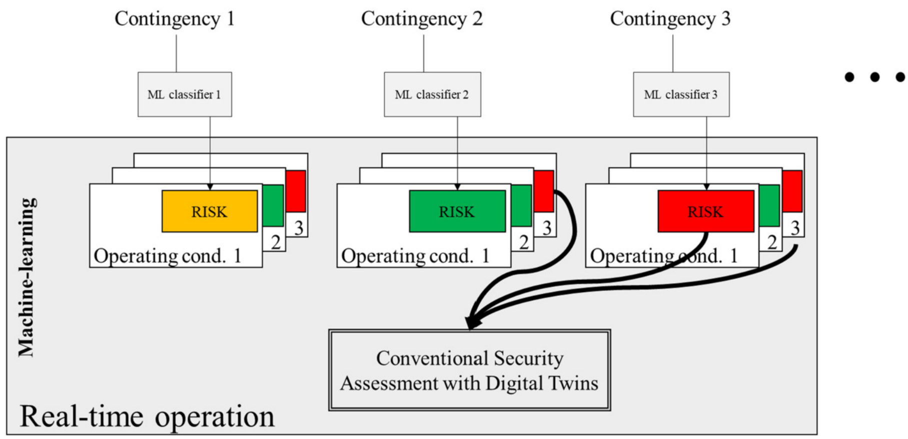

Figure 4.20 Use of ML for security assessment.

evaluates the risk in short-term operation range for multiple contingency scenarios simultaneously by the corresponding ML classifiers [45]. Estimated high-risk scenarios are further investigated by DT simulations to reveal overall system response with high accuracy.

The ML classifier models compute the individual risks of all scenarios $s \in \Omega^S$, which is equivalent to the Cartesian product of operating conditions Ω^I and contingencies Ω^C, $\Omega^S = \Omega^C \times \Omega^I$. The probability of a scenario is $p_s^S = p_c^C \, p_i^I$, or the joint probability of conditions p_i^I and contingencies p_c^C. Then, the ML classifiers compute the scenario risk $R_s = R_c(x_i)$ based on the associated security rules.

Subsequently, the scenario set Ω^S is divided into high risk $\Omega^{S,H}$ and low risk $\Omega^{S,L}$, where $\Omega^S = \Omega^{S,H} \cup \Omega^{S,L}$. Low-risk samples are excluded from DT simulations, since these samples are relatively less costly and DT simulations are computationally more expensive than ML. The total risk (low risk) computed by ML is

$$RISK^{ML} = \sum_{s \in \Omega^{S,L}} p_s^S R_s.$$

The size of the high-risk subset is limited by the DT computational capacity C, which must be greater than the size of high-risk subset $C \geq \left|\Omega^{S,H}\right|$. Thus, the short-term computational capacity of the DT simulations is determined by the amount of time domain security assessments. The DT conducts the conventional security assessment for the high-risk scenario samples only. The DT-based risk assessment is computed as follows

$$RISK^{DT} = \sum_{s \in \Omega^{S,H}} p_s^S \gamma_s.$$

The parameter γ_s represents the severity level of the scenario s security impact, which is computed by time domain simulations from the DT. The final estimated residual risk is computed with the combination of both DT and ML.

$$RISK^{TOT} = RISK^{SA} + RISK^{ML}.$$

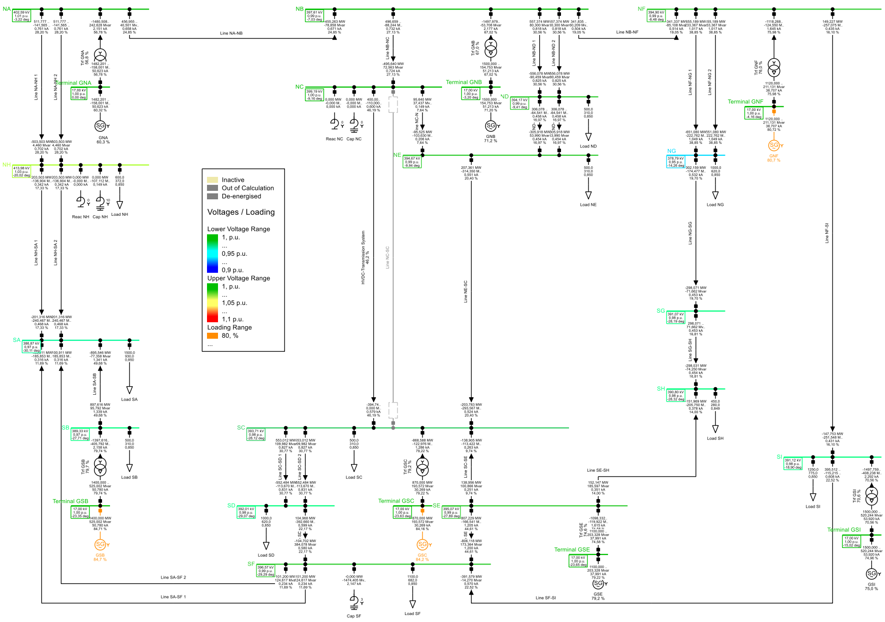

Figure 4.21 Didactic example: base scenario. The time-domain model is adapted from Cigré Technical Brochure 536 [46].

4.4 Conclusion

Power system operation is a promising example that DT simulations and ML can complement one another. Combined, the reliability of DT modeling can be met, while increasing solution speed significantly.

In conclusion, this chapter has revisited concepts from DT and ML and showcased the opportunity in combining them for power system operation. The example of dynamic security assessment has the promise to combine probabilistically the two approaches, ML and DT. This presents the opportunity to investigate ML and DTs for other promising applications, such as emergency control, corrective and preventive control, or congestion management.

4.5 Didactic example

This chapter aimed at providing a strong high-level overview for the reader with references to deepen the learnings. Therefore a didactic example is provided that comprises a selection of case studies from real-world research. The learning objective of the didactic example is to analyze the simulation data (digital twin results) and the training of a machine learning model. Therefore the risk assessment example contains data from a power system model, which is illustrated in Fig. 4.21. The power system model is adapted from Cigré Technical Brochure 536 [46]. It comprises 17 400-kV buses connected by several AC transmission lines, 7 synchronous generators, 11 loads, several shunts, and a VSC-HVDC transmission system connecting bus NC and SC. Details on the didactic example are outlined in the corresponding Jupyter notebook "Dynamic_Risk_Assessment.ipynb" and the related files "FeatureMatrix.csv," "Label.npy" and "SimulationExample.csv" (see Appendix 4.A).

Appendix 4.A Supplementary material

Supplementary material related to this chapter can be found online at https://doi.org/10.1016/B978-0-32-399904-5.00010-7.

References

[1] B.F. Wollenberg, T. Sakaguchi, Artificial intelligence in power system operations, Proceedings of the IEEE 75 (12) (1987) 1678–1685, https://doi.org/10.1109/PROC.1987.13935.
[2] R.P. Schulte, S.L. Larsen, G.B. Sheble, J.N. Wrubel, B.F. Wollenberg, Artificial intelligence solutions to power system operating problems, IEEE Transactions on Power Systems 2 (4) (1987) 920–922, https://doi.org/10.1109/TPWRS.1987.4335274.
[3] K.P. Wong, Applications of artificial intelligence and expert systems in power engineering, in: S. Tzafestas (Ed.), Expert Systems in Engineering Applications, Springer Berlin, Heidelberg, Berlin, Heidelberg, 1993, pp. 160–178.
[4] T.E. Dy-Liacco, Enhancing power system security control, IEEE Computer Applications in Power 10 (3) (1997) 38–41, https://doi.org/10.1109/67.595291.
[5] D. Niebur, Expert systems for power system control in Western Europe, IEEE Control Systems 11 (4) (1991) 34–40, https://doi.org/10.1109/37.88588.
[6] C.-C. Liu, T. Dillon, State-of-the-art of expert system applications to power systems, International Journal of Electrical Power & Energy Systems 14 (2–3) (1992) 86–96, https://doi.org/10.1016/0142-0615(92)90031-4.
[7] Tharam S. Dillon, Michael Laughton (Eds.), Expert System Applications in Power Systems, Prentice Hall International (UK) Ltd., 1990.
[8] M. Nazari-Heris, S. Asadi, B. Mohammadi-Ivatloo, M. Abdar, H. Jebelli, M. Sadat-Mohammadi, Application of Machine Learning and Deep Learning Methods to Power System Problems, Springer International Publishing, Cham, 2021.
[9] F. Bellizio, J.L. Cremer, G. Strbac, Machine-learned security assessment for changing system topologies, International Journal of Electrical Power & Energy Systems 134 (5) (2022) 107380, https://doi.org/10.1016/j.ijepes.2021.107380.
[10] C. Brosinsky, A. Kummerow, M. Richter, A. Naumann, P. Wiest, S. Nicolai, D. Westermann, The role of digital twins in power system automation and control: necessity,

requirements, and benefits, in: ETG-Fachberichte, ETG-Kongress: Das Gesamtsystem Im Fokus der Energiewende, 18–19 Mai 2021, Online-Veranstaltung, 1st ed., VDE Verlag, Berlin, 2021.
[11] A.M. Prostejovsky, C. Brosinsky, K. Heussen, D. Westermann, J. Kreusel, M. Marinelli, The future role of human operators in highly automated electric power systems, Electric Power Systems Research 175 (2019), https://doi.org/10.1016/j.epsr.2019.105883.
[12] C. Brosinsky, D. Westermann, R. Krebs, Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers, in: 2018 IEEE International Energy Conference (ENERGYCON), Limassol, Cyprus, 2018, pp. 1–6.
[13] M. Mazhar Rathore, S.A. Shah, D. Shukla, E. Bentafat, S. Bakiras, The role of AI, machine learning, and big data in digital twinning: a systematic literature review, challenges, and opportunities, IEEE Access (2021) 1, https://doi.org/10.1109/ACCESS.2021.3060863.
[14] L. Wright, S. Davidson, How to tell the difference between a model and a digital twin, Advanced Modeling and Simulation in Engineering Sciences 7 (1) (2020) A6, https://doi.org/10.1186/s40323-020-00147-4.
[15] ARUP, Digital twin: towards a meaningful framework, [Online]. Available www.arup.com/digitaltwinreport. (Accessed 18 December 2020).
[16] A. Kummerow, S. Nicolai, C. Brosinsky, D. Westermann, A. Naumann, M. Richter, Digital-twin based services for advanced monitoring and control of future power systems, in: 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2020, pp. 1–5.
[17] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: a categorical literature review and classification, IFAC-Papers OnLine 51 (11) (2018) 1016–1022, https://doi.org/10.1016/j.ifacol.2018.08.474.
[18] B. Tekinerdogan, C. Verdouw, Systems architecture design pattern catalog for developing digital twins, Sensors (Basel, Switzerland) 20 (18) (2020), https://doi.org/10.3390/s20185103.
[19] Christoph Brosinsky, Rainer Krebs, Dirk Westermann, Embedded digital twins in future energy management systems: paving the way for automated grid control, Automatisierungstechnik 68 (9) (2020) 1–15.
[20] W. Kang, A.J. Krener, M. Xiao, L. Xu, A Survey of Observers for Nonlinear Dynamical Systems, Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications, vol. II, Springer, Berlin Heidelberg, 2013.
[21] A. Agarwal, A. Pandey, L. Pileggi, Robust event-driven dynamic simulation using power flow, Electric Power Systems Research 189 (2020) 106752.
[22] IEEE Power System Operation, Planning and Economics (PSOPE) Committee: Task Force on Power System Dynamic State and Parameter Estimation, Power System Dynamic State and Parameter Estimation-Transition to Power Electronics-Dominated Clean Energy Systems, Technical Report PES-TR88, 2021.
[23] J.B. Rawlings, D.Q. Mayne, M.M. Diehl, Model Predictive Control: Theory, Computation, and Design, 2nd ed., Nob Hill Publishing, Madison, Wisconsin, 2017.
[24] M. Diehl, H.J. Ferreau, N. Haverbeke, Efficient numerical methods for nonlinear MPC and moving horizon estimation, Lecture Notes in Control and Information Sciences (2009) 391–417.
[25] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning: With Applications in R, 8th ed., Springer, New York, Heidelberg, Dordrecht, London, 2017.
[26] S. Ruder, An overview of gradient descent optimization algorithms, 2016.

[27] I. Konstantelos, M. Sun, S.H. Tindemans, S. Issad, P. Panciatici, G. Strbac, Using vine copulas to generate representative system states for machine learning, IEEE Transactions on Power Systems 34 (1) (2019) 225–235, https://doi.org/10.1109/TPWRS.2018.2859367.

[28] D.P. Kroese, T. Taimre, Z.I. Botev, Handbook of Monte Carlo Methods, John Wiley & Sons, Hoboken, New Jersey, 2011.

[29] C. Liu, F. Tang, C. Leth Bak, An accurate online dynamic security assessment scheme based on random forest, Energies 11 (7) (2018) 1914, https://doi.org/10.3390/en11071914.

[30] C. Liu, Z.H. Rather, Z. Chen, C.L. Bak, P. Thogersen, Importance sampling based decision trees for security assessment and the corresponding preventive control schemes: the Danish case study, in: 2013 IEEE Grenoble Conference, Grenoble, 2013, pp. 1–6.

[31] IEEE/CIGRE Joint Task Force on Stability Terms and Definitions, Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions, IEEE Transactions on Power Systems 19 (3) (2004) 1387–1401, https://doi.org/10.1109/TPWRS.2004.825981.

[32] N. Hatziargyriou, et al., Definition and classification of power system stability revisited & extended, IEEE Transactions on Power Systems (2020) 1, https://doi.org/10.1109/TPWRS.2020.3041774.

[33] U. Kerin, C. Heyde, R. Krebs, E. Lerch, Real-time dynamic security assessment of power grids, The European Physical Journal Special Topics 223 (12) (2014) 2503–2516, https://doi.org/10.1140/epjst/e2014-02272-1.

[34] C.O. Heyde, Dynamic Voltage Security Assessment for on-Line Control Room Application (Dynamische Spannungsstabilitätsrechnungen als online Entscheidungsgrundlage für die Leitwarte), Diss., 1st ed., Magdeburg, Univ., Fak. für Elektrotechnik und Informationstechnik, Magdeburg, 2010.

[35] O. Ruhle, E. Lerch, Ranking of system contingencies in DSA systems – first experiences, in: IEEE PES General Meeting, Minneapolis, MN, 2010, pp. 1–6.

[36] J.M. Gimenez Alvarez, P.E. Mercado, Online inference of the dynamic security level of power systems using fuzzy techniques, IEEE Transactions on Power Systems 22 (2) (2007) 717–726, https://doi.org/10.1109/TPWRS.2007.895161.

[37] Cigré WG C4.601, Review of on-line dynamic security assessment tools and techniques, Technical Brochure No. 325, 2007.

[38] ENTSO-E, Dynamic security assessment (DSA), SPD DSA Task Force, RG-CE System Protection & Dynamics Sub Group [Online]. Available https://docstore.entsoe.eu/Documents/SOC%20documents/Regional_Groups_Continental_Europe/2017/DSA_REPORT_Public.pdf, Apr. 2017.

[39] Commission Regulation (EU) 2017/1485 of 2 August 2017 establishing a guideline on electricity transmission system operation: OJ L 220 Official Journal of the European Union (2017), [Online]. Available http://data.europa.eu/eli/reg/2017/1485/oj.

[40] S.C. Savulescu, Real-Time Stability in Power Systems: Techniques for Early Detection of the Risk of Blackout, 2nd ed., Springer International Publishing, Cham, 2014.

[41] L. Wehenkel, M. Pavella, Decision trees and transient stability of electric power systems, Automatica 27 (1) (1991) 115–134, https://doi.org/10.1016/0005-1098(91)90010-Y.

[42] E.S. Karapidakis, N.D. Hatziargyriou, On-line preventive dynamic security of isolated power systems using decision trees, IEEE Power Engineering Review 22 (4) (2002) 75, https://doi.org/10.1109/MPER.2002.4312129.

[43] A.-A.B. Bugaje, J.L. Cremer, M. Sun, G. Strbac, Selecting decision trees for power system security assessment, Energy and AI 6 (5) (2021) 100110, https://doi.org/10.1016/j.egyai.2021.100110.

[44] J.L. Cremer, I. Konstantelos, G. Strbac, From optimization-based machine learning to interpretable security rules for operation, IEEE Transactions on Power Systems 34 (5) (2019) 3826–3836, https://doi.org/10.1109/TPWRS.2019.2911598.
[45] J.L. Cremer, G. Strbac, A machine-learning based probabilistic perspective on dynamic security assessment, International Journal of Electrical Power & Energy Systems 128 (5) (2021) 106571, https://doi.org/10.1016/j.ijepes.2020.106571.
[46] Cigré Joint Working Group C4/B4/C1.604, Influence of Embedded HVDC Transmission on System Security and AC Network Performance, Technical Brochure 536, 2013.