Online Disturbance Prediction for Enhanced Availability in Smart Grids

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Igor Kaitovic

under the supervision of
Prof. Miroslaw Malek

November 2017

Dissertation Committee

Prof. Fernando Pedone	Università della Svizzera italiana, Switzerland
Prof. Olaf Schenk	Università della Svizzera italiana, Switzerland
Prof. Mario Paolone	École polytechnique fédérale de Lausanne, Switzerland
Prof. Paulo Verissimo	University of Luxembourg, Luxembourg

Dissertation accepted on 17 November 2017

Research Advisor	PhD Program Director
Prof. Miroslaw Malek	**Prof. Walter Binder**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Igor Kaitovic
Lugano, 17 November 2017

To my wife and my mother.

"Ordinary mortals know what’s happening now,
The gods know what the future holds
Because they alone are totally enlightened.
Wise men are aware of future things
Just about to happen."

Constantine P. Cavafy (1863 - 1933)

Abstract

A gradual move in the electric power industry towards Smart Grids brings new challenges to the system's efficiency and dependability. With a growing complexity and massive introduction of renewable generation, particularly at the distribution level, the number of faults and, consequently, disturbances (errors and failures) is expected to increase significantly. This threatens to compromise grid's availability as traditional, reactive management approaches may soon become insufficient. On the other hand, with grids' digitalization, real-time status data are becoming available. These data may be used to develop advanced management and control methods for a sustainable, more efficient and more dependable grid. A proactive management approach, based on the use of real-time data for predicting near-future disturbances and acting in their anticipation, has already been identified by the Smart Grid community as one of the main pillars of dependability of the future grid.

The work presented in this dissertation focuses on predicting disturbances in Active Distributions Networks (ADNs) that are a part of the Smart Grid that evolves the most. These are distribution networks with high share of (renewable) distributed generation and with systems in place for real-time monitoring and control. Our main goal is to develop a methodology for proactive network management, in a sense of proactive mitigation of disturbances, and to design and implement a method for their prediction. We focus on predicting voltage sags as they are identified as one of the most frequent and severe disturbances in distribution networks.

We address Smart Grid dependability in a holistic manner by considering its cyber and physical aspects. As a result, we identify Smart Grid dependability properties and develop a taxonomy of faults that contribute to better understanding of the overall dependability of the future grid. As the process of grid's digitization is still ongoing there is a general problem of a lack of data on the grid's status and especially disturbance-related data. These data are necessary to design an accurate disturbance predictor. To overcome this obstacle we introduce a concept of fault injection to simulation of power systems. We develop a framework to simulate a behavior of distribution networks in the presence of faults, and fluctuating generation and load that, alone or combined, may cause disturbances. With the framework we generate a large set of data that we use to develop and evaluate a voltage-sag disturbance predictor. To quantify how prediction and proactive mitigation of disturbances enhance availability we create an availability model of a proactive management. The model is generic and may be applied to evaluate the effect of proactive management on availability in other types of systems, and adapted for quantifying other types of properties as well. Also, we design a metric and a method for optimizing failure prediction to maximize availability with proactive approach.

In our conclusion, the level of availability improvement with proactive approach is comparable to the one when using high-reliability and costly components. Following the results of the case study conducted for a 14-bus ADN, grid's availability may be improved by up to an order of magnitude if disturbances are managed proactively instead of reactively.

The main results and contributions may be summarized as follows: (i) Taxonomy of faults in Smart Grid has been developed; (ii) Methodology and methods for proactive management of disturbances have been proposed; (iii) Model to quantify availability with proactive management has been developed; (iv) Simulation and fault-injection framework has been designed and implemented to generate disturbance-related data; (v) In the scope of a case study, a voltage-sag predictor, based on machine-learning classification algorithms, has been designed and the effect of proactive disturbance management on downtime and availability has been quantified.

Keywords: Active Distribution Networks, Availability, Data analytics, Dependability, Disturbances, Fault injection, Machine learning, Prediction, Proactive management, Simulation, Smart Grid, Voltage sags

Acknowledgements

First and foremost I would like to thank my advisor, Prof. Miroslaw Malek for the continuous support, numerous valuable discussions and for his friendly and open attitude. Also, I would like to thank some of the members of the ALaRI institute, Faculty of informatics, and the SCCER-FURIES project, particularly Slobodan Lukovic, for collaboration on projects and publications and for fruitful discussions. Besides, my many former students have helped me to widen my knowledge and to enhance my work. Supporting Filip Obradovic in his master's thesis and working with him on a publication had a strong impact on my dissertation results. My work also benefited greatly from comments and suggestions from my committee members and publication reviewers to whom I am very grateful.

I thank the Hasler Foundation and the Commission for Technology and Innovation for funding a part of this work. Also, I thank Università della Svizzera italiana and the administrative director, Albino Zgraggen, for granting me a scholarship for the completion of my work as well as the dean's office of the Faculty of informatics for an excellent administrative support and their friendship.

Finally, I would like to express my limitless gratitude to my wife for her unselfish support and love and for proofreading some of my publications and parts of the manuscript, as well as to my parents, for their support, understanding and unconditional love.

Contents

Figures

Tables

List of Acronyms

AMI	Automatic Metering Infrastructure
AND	Active Distribution Network
ASAI	Average Service Availability Index
AVR	Automatic Voltage Regulator
CAIDI	Customer Average Interruption Duration Index
CIGRE	International Council on Large Electric Systems
CIM	Common Information Model
CPES	Cyber-Physical Energy Systems
CPS	Cyber-Physical System
CSD	Computer Systems Dependability
CTMC	Continuous-Time Markov Chain
DER	Distributed Energy Resource
DG	Distributed Generator
DMS	Distribution Management System
DSM	Demand Side Management
DSO	Distribution System Operator
DSS	Decision Support System
EMS	Energy Management System
EPS	Electric Power System
ES	Energy Storage
EV	Electric Vehicle
FACTS	Flexible AC Transmission System
FDR	Frequency Disturbance Record
FT	Fault Tolerance
HVDC	High-Voltage Direct Current
ICT	Information and Communication Technology
IED	Intelligent Embedded Device
IEEE	Institute of Electrical and Electronics Engineers

IoT	Internet of Things
MPC	Model-Predictive Control
MTTF	Mean-Time-To-Failure
MTTR	Mean-Time-To-Repair
NIST	National Institute of Standards and Technology
OLTC	Online Tap Changer
PCA	Principal Component Analysis
PLC	Power-Line Communication
PMU	Phasor Measurement Unit
PSAT	Power System Analysis Toolbox
PV	PhotoVoltaics
RBD	Reliability Block Diagram
RCM	Reliability-Centered Maintenance
RES	Renewable Energy Source
rms	root mean square
SAFRI	System Average interruption FRequency Index
SAIDI	System Average Interruption Duration Index
SCADA	Supervisory Control And Data Acquisition
SCCER	Swiss Competence Center for Energy Research
SPNP	Stochastic Petri Net Package
SRARPE	Symbolic Hierarchical Automated Reliability and Performance Evaluator
SST	Solid-State Transformer
SVM	Support Vector Machine
UPS	Uninterrupted Power Supply
V2G	Vehicle-To-Grid
VM	Virtual Machine
VPP	Virtual Power Plan
VPS	Virtual Power System
WAMS	Wide-Area Monitoring System
WEKA	Waikato Environment for Knowledge Analysis
WSN	Wireless Sensor Network

Chapter 1

Introduction and Motivation

A concept of Smart Grid is first introduced and frequently used Smart Grid definitions are reviewed. The main needs for modernization of electric power grids, trends and enabling technologies are then explained in more detail. Following this, rising Smart Grid dependability concerns are identified, giving a broader motivation for the work presented in the manuscript. A proactive management paradigm, as one of the ways to address these concerns, is then described. Next, the addressed problem and the problem area are specified followed by a list of major contributions.

1.1 Smart Grids - A Brief Overview

Essentially, Smart Grid may be described as an electric power system that, in the light of growing demand and efforts to reduce the use of fossil fuels, relies on renewable generation as well as on enhanced control that is based on a wider adoption of ICT elements and data analytics in order to ensure sustainable electric power delivery, to keep the system reliable, to make it more efficient and cost effective, as well as to provide new customer services. The Smart Grid concept is still under development and thus it lacks a formal, comprehensive and widely-accepted definition. Instead, a few definitions are used in the community, each emphasizing different aspects of Smart Grids.

The US National Institute of Standards and Technology (NIST) uses the term "Smart Grid" to refer to "a modernization of the electricity delivery system so that it monitors, protects and automatically optimizes the operation of its interconnected elements - from the central and distributed generator through the high-voltage transmission network and the distribution system, to industrial users and building automation systems, to energy storage installations and to end-use consumers and their thermostats, electric vehicles, appliances and other household devices [1]." European Technology Platform's Smart Grid definition is more concise describing it as "an electricity network that can intelligently integrate the actions of all users connected to it - generators, consumers and those that do both - in order to efficiently deliver sustainable, economic and secure electricity supplies [2]." Major industry players such as ABB, General Electric, IBM, and Siemens as well as professional associations including CIGRE and IEEE also have their own definitions. A comprehensive overview of them and other ones may be found in [3].

Considering these and other definitions including those given in [4, 5, 6, 7, 8], in a very broad sense, Smart Grid may be described as an evolved electric power system that supports integration of Distributed Energy Resources (DERs), Renewable Energy Sources (RES'es), Energy Storages (ES'es), and Electric Vehicles (EVs). It relies on massive adaptation of ICT components such as meters and sensors for better grid visibility at all levels, reliable and cyber-secure two-way communication, and the use of online information for provision of new services to both system operators and customers as well as methods for optimizing management to meet efficiency, cost effectiveness, dependability, resilience and sustainability requirements.

An overview of the main Smart Grid structural elements, with the focus on the distribution network is presented in Figure 1.1. This is an extended version of a figure that has been originally published in [9].

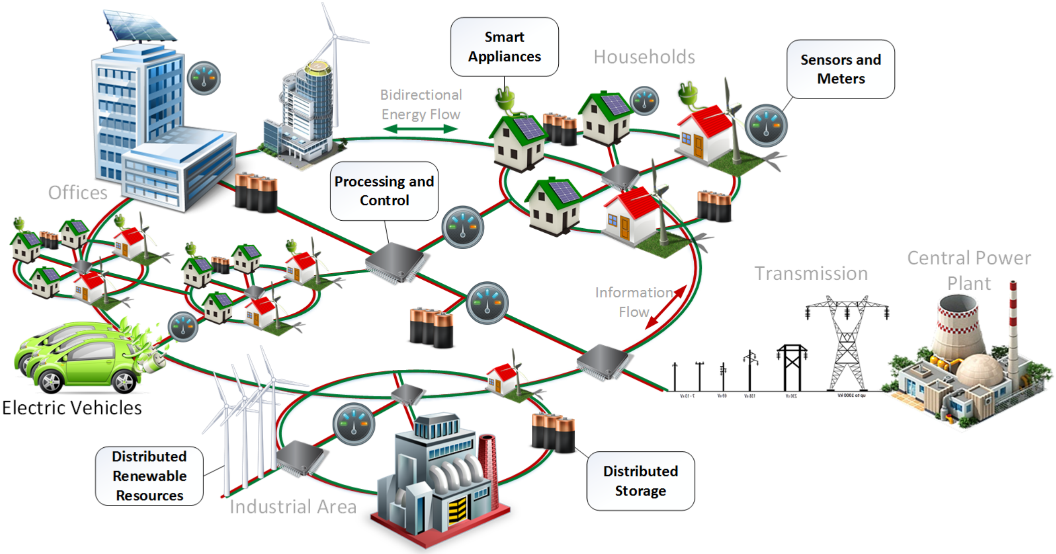


Figure 1.1. Overview of Smart Grid elements.

1.1.1 Needs and Trends

The structure of today's electric power system has not essentially changed since the end of the 19$^{\text{th}}$ century. It is based on the generation-transmission-distribution-consumption paradigm and on a unidirectional power flow [10]. With some aged and, with respect to today's requirements, outdated components, parts of the power system are already operating at their full capacity [11] as the grid is approaching its operational limits [12]. On the other hand, despite more efficient industrial processes, a demand for electricity is gradually increasing. According to EUREL's[1] forecast from 2013 [13], annual electricity demand in Europe will rise to approximately 4300 TWh by 2050. For the comparison, electricity consumption in Europe in 2008 was 3043 TWh. This requires an increase in power generation but also a grid modernization in a sense of structural changes and new management strategies in order to support delivery of more power while maintaining the system's stability. Moreover, as the process of electrification in developing countries is still ongoing, the grid is also gradually expanding.

Efforts to reduce global greenhouse gas emission drive additional changes in power systems. This includes an increase in use of electric power in industry and transportation (e.g. electric vehicles) and decrease of share of fossil fuels in electricity generation-mix by using more RES'es. EU is particularly interested in wider adaptation of renewable generation as a part of decreasing its dependency on fossil fuel import. Initiative of the EU commission is to increase the share of generation from RES'es from today's 25% to 40% until 2020. Also, the forecasts are that renewable capacities production by 2050 will be between 50% and 80% [13]. This causes a major paradigm shift from traditional, centralized, to distributed electric power generation as most of the renewable generation is based on DERs. It particularly effects distribution part of the grid as it has to support a bidirectional energy flow in order to accept a larger number of distributed generators. Management strategies and protection mechanisms will also have to be adapted and improved to ensure system's efficiency and stability especially considering volatile nature of renewable generation. In fact, when well controlled, DERs may even provide an opportunity to enhance grid's stability and dependability [14, 15]. Higher adaptation of DERs is also in line with energy market deregulation (liberalization) efforts that aim at increasing competition to motivate better, more efficient and less expensive electric power system services. On the other hand, this calls for more changes in the system management due to decentralized decision making, interdependencies and a lack of fully centralized

[1] EUREL - The Convention of National Association of Electrical Engineers of Europe, www.eurel.org/

control over system components [16].

Massive adaptation of (plug-in hybrid) EVs is another opportunity to significantly reduce green-house gas emission provided a high emission share of transpiration in developed countries [17]. On the other hand, power demanding EVs will add a significant load to the existing distribution system and, in some cases may cause overloads as well as power quality degradations. Coordinated charging of EVs may help in overcoming some of these problems [18].

Volatile nature of DERs, their low controllability and increasing number of EVs, trigger another important change in the system operation approach from traditional, consumption-driven generation to generation-driven consumption. In fact, various Demand Side Management (DSM) strategies are already being put in place to change power demand through dynamic pricing, greater customer engagement and various other initiatives, in order to tailor consumption to the current production and state of the grid and to maintain the power balance [19]. For example, with DSM a consumption at peak hours may be decreased so that generation can meet the demand such that system's stability is not jeopardized. At the same time DSM aims at maximizing local consumption from DERs which decreases transmission and distribution losses as well as avoiding investments in long-distance transmission lines [12].

Better control and integration of DERs, may be realized through the concept of Virtual Power Plan (VPP) [20, 21]. A VPP is a logical aggregation of DERs that, considering their place in the grid and the current status information, manages generation and presents the entire aggregation, to the rest of the system, as one, virtual generator equivalent to a traditional centralized power plant. This way, a VPP increases visibility of DERs and allows their integration into the grid. The concept of VPPs has been further expanded to Virtual Power Systems (VPS'es) that represent logical aggregations of DERs and controllable loads. That way, energy balance is optimized within a VPS and, from the perspective of the rest of the grid, the aggregation is considered as a "prosumer" that, depending on the current power balance, may be seen as a load or as a generator [22]. A similar approach is taken in Microgrids. These are distribution grids that may operate interdependently but may still be connected to the central grid.

Higher adaptation of DERs and EVs also brings new opportunities for provision of ancillary services. These are additional services provided on top of real power generation and distribution, including reactive power supply, voltage and frequency control, operating reserves, energy balance, and network stability [23]. For example, a vehicle-to-grid (V2G) concept proposes to use batteries of stationary EVs to provide power to the grid in the case of unpredicted load increase and insufficient generation [24].

1.1.2 Enabling Technologies

The backbone to fulfilling the aforementioned needs and to realize described concepts and trends is the ICT infrastructure that includes monitoring, data acquisition, and communication infrastructures as well as advanced data analytics and simulation tools to improve grid status awareness and to provide a basis for post-mortem analysis as well as for online detection and prediction of disturbances. This information is then used for planning grid improvements and extensions and for real-time control. In addition, for implementing real-time control, it is necessary to deploy responsive actuators and control devices in order to be able to quickly react on detected or predicted events.

1.1.2.1 Monitoring and Communication Infrastructure

Grid monitoring and data acquisition infrastructure, at power distribution level, includes various meters, sensors as well as systems for data collection and aggregation. Automatic Metering Infrastructure (AMI) which relies on smart metering technologies allow not only to automatically collect consumption readings and device status information but also, using two-way communication, to control these devices remotely [4]. In most of today's implementations the control through meters, is limited to switching the power in an entire household whereas switchers and actuators may be used to control individual devices.

Various types of sensor networks, for example, the one proposed in [25], may be installed to observe the grid health-status at distribution level. In that sense, Phasor Measurement Units (PMUs) are becoming increasingly important as they provide UTC time-tagged and synchronized estimations of electricity waveforms (frequency, rate-of-change-of-frequency, and voltage/current phasors) [26] in resolution much higher than the traditional Supervisory Control And Data Acquisition (SCADA) systems may provide [8]. Typical sampling period of a SCADA system is between 2 s and 4 s [8], whereas modern PMUs provide measurements at every 20 ms [27]. The principle of PMU operation is based on sampling waveforms and synchronizing them using an accurate time reference. A synchronphasor estimator then extracts the base tone from the sampled (raw) data collected over the PMU sampling period (observe that sampling of raw data occurs with much shorter period) and provides an estimated waveform as an output. Obviously, the process introduces uncertainty and the accuracy of the estimated waveforms depends on the accuracy of the time source, sampling and conditioning unit and the synchronphasor. Details on PMU operation may be found in [26] and [28].

Having high sampling rate and being more accurate than the other monitoring devices, PMUs are the central part of Wide-Area Monitoring Systems (WAMS'es) and are the basis for accurate state estimation and real-time visibility of the grid and its parts [29]. Moreover, PMU data may be used for monitoring instabilities, and detecting and locating disturbances [4]. As relatively expensive devices there are mostly used at transmission level but their adoption is expected to take part also at distribution level in particular in ADNs. For example, [30] proposes a solution for integrating PMUs into monitoring system of distribution networks that have high share of renewable generation.

Smart Grids fully rely on a secure and reliable two-way communication infrastructure for the delivery of real-time information to processing centers and for sending the control signals back to actuating devices. Smart Grid communication infrastructure itself is a highly complex system that allows communication between all system elements while incorporating business and home area networks to connect to office and household appliances [31]. In order to fulfill these requirements, different communication technologies have to be combined including power-line communication (PLC), fiber optics and numerous wireless solutions from Wireless Sensor Networks (WSNs), GSM, GPRS to satellite communication [4]. Standardization and interoperability of existing technologies represent one of the most challenging tasks when it comes to implementation of this aspect of Smart Grid concept. In that respect, remarkable efforts have been invested and considerable progress has been made with wider implementation of substations automation as well as communication standard synchronization through a number of widely promoted and accepted standards and protocols such as CIM and IEC61850 [3].

1.1.2.2 Data Analytics

The process of power grid digitalization is still ongoing but, according to [32], the data volume in Smart Grids will be still greater by multiple orders of magnitude than in traditional grids. This includes operational data (e.g. voltage, phase, and active and reactive power measurements), non-operational data (e.g. power quality and reliability data), meter usage data (e.g. statistics on average and peak consumption), event message data (e.g. fault detection messages), and metadata (e.g. grid topology). In fact, a vast amount of data is already being collected from various devices in different formats and their unification is necessary in order to process such a data volume. This is the main motivation for the development of a standard data exchange framework known as the Common Information Model (CIM) [33]. CIM defines interoperability semantics, syntax for data exchange and proposes implementation technologies. This data may be further analyzed in a centralized or distributed fashion as a part of the existing SCADA, Energy Management Systems (EMS'es) and Distribution Management Systems (DMS'es). Enablers for the data analysis are high performance and cloud computing systems [8] and the main goal is to provide more accurate and optimal grid management through better understanding of the grid state and the prediction of future state.

The current trend is mainly in statistical analysis and postmortem analysis but also in real-time state estimation [29], grid status visualization [34], and demand forecast. Future applications may include also prediction of grid disturbances (as, for example, planned in the scope of the GridEye/FNET project [34]) [8]. Also, IBM's vision on big data in Smart Grids [35] is to use predictive analytics not only for a wide range of predictions including demand and generation forecasts and anticipation of customers' behavior but also for the prediction of equipment downtime and grid failures.

1.1.2.3 Simulation Tools

Considering safety-critical aspects of the grid, simulation provides an affordable and safe alternative to analyze and to validate new control strategies before their deployment. A challenge for Smart Grid simulation tools is that, besides simulating state of the electric infrastructure with high sampling rate and incorporating advanced components such as PMUs [36], they must also consider ICT infrastructure as well as their interdependency. We give a brief overview of only a few commonly used commercial and academic tools whereas a more comprehensive and an up-to-date list may be found at OpenElectrical's page[2].

- Power System Planning and Data Management (PSS)[3] is a set of tools provided by Siemens for system planning and operation with accurate and efficient system analyses. This includes tools for contingency and fault analysis as well as modeling of corrective actions, analysis during changing network conditions, simulation of distributed generation, dynamic simulations in time and frequency domains and CIM-compliant data representation. It is also important to point out that PSS is frequently used as a benchmark for validation of open-source tools.

[2] OpenElectrical's list of power systems analysis software, http://www.openelectrical.org/wiki/index.php?title=Power_Systems_Analysis_Software

[3] Siemens Power System Planning and Data Management suite, http://w3.siemens.com/smartgrid/global/en/products-systems-solutions/software-solutions/planning-data-management-software/Pages/overview.aspx

- Neplan[4] is Swiss-based company modular software based on MATLAB and C/C++ that provides a number of features ranging from load flow, dynamic simulation, and optimal power flow to reliability analysis and load forecasting.
- DIgSILENT PowerFactory[5], is a Windows-based integrated power systems modeling and analysis package that, besides standard power system analysis, includes high-end applications in new technologies such as wind power and distributed generation as well as algorithms for handling very large power systems.
- OPAL-RT HYPERSIM[6] is a real-time simulation platform that supports integration of DERs and incorporates real-time monitoring, control and protection features. It also supports hardware-in-the-loop testing.
- MatDyn[7] [37] is an open-source software for dynamic analysis of power systems developed as an extension of MATPOWER[8] [38] simulation tool for load flow and optimal power flow analysis.
- Power Systems Analysis Toolbox (PSAT)[9] [39] is an open-source MATLAB-based tool for analysis and design of small to medium size electric power systems that supports power flow, continuation power flow, optimal power flow, small-signal stability analysis, and time-domain simulation. It also includes several static and dynamic models of distribution and transmission grids.

1.1.2.4 Real-Time Actuators and Controllers

In order to be able to apply management strategies based on data analytics for grid control and to quickly react to detected or anticipated problems, new types of intelligent devices and actuators must be widely deployed. Among those that are expected to significantly contribute to the system's reliability are Intelligent Embedded Devices (IEDs), Flexible AC Transmission Systems (FACTS) and Solid-State Transformers (SSTs). IEDs may be controlled not only remotely but also provide the capability of local data processing and decision making. Different types of IEDs may also be used to control household appliances. FACTS provide control of AC transmission parameters. For example, they can alter power flow through transmission lines. Equipped with real-time control algorithms these

[4] Neplan, www.neplan.ch

[5] DIgSILENT PowerFactory, http://www.digsilent.de/index.php/products-powerfactory.html

[6] OPAL-RT HYPERSIM, http://www.opal-rt.com/systems-hypersim/

[7] MatDyn, http://www.esat.kuleuven.be/electa/teaching/matdyn/

[8] MATPOWER, http://www.pserc.cornell.edu/matpower/

[9] Power Systems Analysis Toolbox (PSAT), http://faraday1.ucd.ie/psat.html

devices may help improving system's reliability [40]. SSTs are lighter and more efficient than traditional transformers. Moreover, they come with new functions such as power flow control, voltage sag compensation, and fault current limitation [41]. As such, they are also seen as an efficient way of coping with intermittent nature of renewable energy generators.

1.1.3 Active Distribution Networks

A traditional distribution network is designed as a passive one with radial topology and unidirectional power flow. It accepts bulk power from the transmission grid and distributes it to customers [42]. Typically, it has a very limited monitoring infrastructure as most of the power flow problems are resolved at planning and design time [43, 44].

With exponential growth of distributed resources (DG) that include both DERs and ES'es, distribution network is significantly changing and becoming an active part of the power system. As defined by International Council on Large Electric Systems (CIGRE)[10] an Active Distribution Network (ADN) is a distribution network that has systems in place to control a combination of distributed energy source (generators, loads and storage) [45].

With topology changes, bidirectional energy flow and intermittent resources, more faults and disturbances may be expected at this level of the grid. This includes greater voltage variations, power imbalance and supply quality decrease in general [44]. Redundancy, as a way to ensuring network's reliability may be too expensive and also not sufficiently efficient solution from the perspective of generated power exploitation. Therefore, novel control algorithms must be developed and put in place. This includes methods for fast online state estimation with PMUs [29, 46] as well as methods for DERs [47] and ES'es [48] integration and control for voltage and frequency regulation.

1.2 Rising Smart Grid Dependability Concerns

Compromising power system's dependability may cause a high financial loss or may even have more severe consequences related to grid's safety-critical aspects. According to US Department of Energy report from 2009, an annual cost of power outages and interruptions in US is at least 150 billion dollars [49]. The same report reveals that grid's availability in US equals to 0.9997 that corresponds to more than 2.5 h of complete power interruption per year. Considering the

[10] International Council on Large Electric Systems (CIGRE), http://www.cigre.org/

challenges that were reviewed in the previous subsections, we may only expect a decrease in availability if new approaches to maintain grid's dependability are not imposed.

1.2.1 Dependability Challenges

With (r)evolution of electric power systems and foreseen changes, the risk of compromising its dependability is increasing. Here we identify, categorize and describe the main Smart Grid dependability challenges:

- **Complexity and interdependency** With evolution towards Smart Grid, complexity of the system is rapidly increasing, and ensuring its dependability becomes more difficult, especially when taken into account that Smart Grid closely combines electric power and ICT infrastructures. As such, it is a good example of a system-of-systems whose operation and management requires to combine knowledge and best practice from various domains. Since a systematic approach to Smart Grid dependability does not exist at this time, the danger is that dependability may be compromised due to generally different approaches in the two communities (ICT and electric power), including diverse terminology for dependability attributes and frequently interchangeable use of terms [50], different understanding of basic concepts such as fault, error and failure, and different figures of merit for the quantification of dependability attributes. Moreover, with growing interdependency between ICT (cyber) and electric power (physical) infrastructures, components' failures may propagate from one infrastructure to another [51], causing errors and failures.

- **Structural changes.** Generation paradigm shift and bidirectional energy flow significantly change grid's structure, especially at the distribution level. Protection systems are designed with unidirectional power flow in mind [14] and they may fail to ensure reliability when the flow is bidirectional. Moreover, due to deregulation and economic reasons, the system is becoming more congested, has fewer redundant components and frequently operates at its full capacity.

- **Increasing number of faults.** Due to growing complexity, we may expect an increase in the number of faults and also new types of power grid faults in the near future [52]. Aging infrastructure and smaller energy margins may also cause an increase in the number of faults. Finally, as the complexity increases, more human faults may also be expected.

- **Uncertainties**. With the market deregulation, the level of uncertainty in power systems has increased significantly [53]. This includes uncertainty on availability of generators, transmission and distribution lines, electricity prices as well as uncertainty on load and (renewable) generation forecasts that are frequently weakly correlated with real production and consumption [5]. Volatile nature of RES'es particularly contributes to the uncertainty of electric power generation. Moreover, with growing demand and introduction of EVs as power-demanding loads, predicting customers behavior is ever-more important and, at the same time, more challenging.

- **Cyber security.** As the Smart Grid expands and incorporates more ICT elements, from servers to embedded devices and even novel technologies as Internet-of-Things (IoT), it opens a door to a plethora of attack opportunities that bring its security concerns to an extreme [54]. Practically, every connected device (e.g. a smart meter or a household appliance) is a potential gateway for a cyber-attack [4]. Some of these attacks may also cause malfunction of power equipment, disturbances and outages, thus reducing power delivery service availability. In fact, a few recent outages are believed to have been caused by cyber-attacks (e.g. those reported in [55]).

All of this is threatening to increase the number and severity of disturbances, including voltage and frequency deviations as well as brownouts (long-term voltage drops) and blackouts. For example, one such blackout occurred in Amsterdam (that is considered to be an advanced-stage Smart City with a Smart Microgrid), in March 2015 affecting almost 3 million people [56]. In [57] we have reviewed almost 40 recent events such as this one. Many of these events have left tens and hundreds of millions of people without electricity for tens of hours. This obviously demonstrates the need for new methods for grid management.

1.2.2 A Need for Revision of Power Grid Reliability Standards

Power grid reliability standards, that are still in use, have been defined in 1950s and need to be fundamentally reviewed in the light of the grid evolution [15]. In brief, these standards mainly require sufficient redundancy to ensure reliability and do not consider Smart Grid solutions based on, for example, real-time monitoring and control and energy storage. As such they are not economically efficient as the result is low utilization of assets and higher operational cost. Moreover, the current standard is deterministic in the sense that the system is considered as free of risks of disturbances as long as no operational limits are violated (e.g.

all voltages and frequencies are in a defined range), and in an unacceptable level of risk if operating limits are violated. A more realistic approach would be to estimate the risk of a disturbance at runtime considering current system state and previously observed disturbances.

On the other hand, understanding dependability of Smart Grid and proposing new standards is difficult as the concept is still under development and the grid is evolving gradually. There is still not sufficient data to model and to evaluate dependability and to design new methods for its enhancement. Even when disturbance data is available, for example in industrial pilots, due to high sensitivity and security concerns, such data is rarely, if at all, available for public use.

1.2.3 Dependability of Active Distribution Networks

As distribution grids are rapidly evolving towards Active Distribution Networks, there is a concern that dependability of this part of the grid may also be affected to a greater extent. In fact, Distribution System Operators (DSOs) are already faced with disturbances such as voltage variations and congestions [44]. Overvoltage is a common problem that occurs at a DER connection point and the surrounding area. Reverse power flow may also occur due to a DER generation that compromises dependability as protection schemes in a radial distribution network that are designed with unidirectional flow in mind.

Voltage sags and swells are particularly dangerous as they may damage voltage sensitive equipment [58]. As defined by IEEE standard 1159 [59], a voltage sag (dip) is a decrease in voltage root mean square (rms) value to between 0.1 and 0.9 from the nominal value for a duration of 0.5 cycle to 1 minute. Similarly, a swell (surge) is an increase of rms to between 1.1 and 1.8 of the nominal value for 0.5 cycle to 1 minute. Voltage problems in general are identified as the most severe ones in distribution grids as frequency is regulated at higher levels. In fact, voltage disturbances are also observed in some mid-size ADNs such as the one in Rheinfelden, Switzerland and that has been analyzed in the scope of the VEiN[11] project.

A congestion is a type of power imbalance that occurs when a distributed generator pushes the system beyond its physical limits causing a power interruption. An imbalance may also occur when local generation is not sufficient to meet the demand. In principal, DERs may help ADN's dependability if well controlled and if sufficiently close to points of consumption. Unfortunately, this is not always the case as DERs, in most cases, cannot be dispatched [44].

[11] VEiN: Verteilte Einspeisung in Niederspannungsnetze, http://www.vein-grid.ch

There are several ways of coping with disturbances in ADNs for improved dependability. Redundancy is the most obvious one but it is not always efficient. More sophisticated solutions include different protection algorithms as the ones proposed in [14], coordinated distributed generation and load control schemes as in [47] and [60] as well as methods based on distributed storage control such as, for example, the one proposed in [61]. Finally, methods based on online prediction of oscillations as well as voltage sags and swell as in [62] and [63] are also being proposed.

1.3 Proactive Management Concept

Traditional approaches to electric power grid operation are essentially reactive and based on detect-localize-repair paradigm. If a disturbing event is detected (e.g. a voltage sag, a frequency deviation or a line trip), a fault location is first determined and corrective actions are taken (e.g. load shedding) to bring the system back to a stable state and to prevent disturbance propagation. As previously pointed out, this traditional reactive approach becomes insufficient in the face of increasing complexity, uncertainties, interdependencies as well as higher number and new types of faults and disturbances.

With the ability to collect and to analyze increasing amount of data in real time, we observe a rapid paradigm shift in all industrial spheres from analyzing the past and monitoring the present, to predicting and prescribing the future (as on the Figure 1.2 from Gartner[12]). For systems such as Smart Grids and others, we used to ask questions like what happened and why it happened but now with the ability to collect more data and with rapid processing capability, we are or will be able to ask questions about the future: what will happen and what can we do about it? Obviously, the level of difficulty increases but so does the expected level of dependability, as knowing about future problems gives a possibility to prepare for them and to handle them more efficiently.

Managing the grid proactively, in the sense of predicting future problems by using data analytics and acting in their anticipation to prevent or to mitigate them, may significantly improve grid's dependability and is identified as one of the pillars of Smart Grid's dependability [8]. In this sense, we may differentiate between two types of proactive management based on prediction of failures [64]: predict-prevent and predict-mitigate. In the first case, after a prediction of a failure (or a disturbance) actions are taken to avoid it. For example, in

[12] Gartner IT Glossary, Predictive Analytics:
http://www.gartner.com/it-glossary/predictive-analytics, December 2015

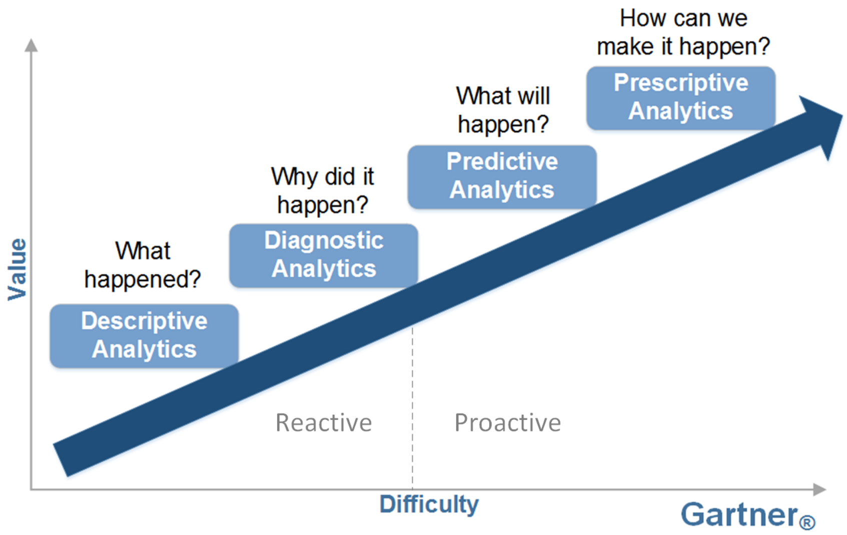


Figure 1.2. System management paradigm evolution.

a cloud environment, virtual machines may be migrated to another host when a failure of the main host is predicted. In power systems, for example, we may use controllable distributed generators when a voltage sag at a specific location is predicted. This boosts availability when compared to typical reactive approaches as downtime is, in an ideal case, fully avoided.

Still, not all problems may be anticipated in advance and prevented. In cases when the prediction lead time is not sufficient for prevention, or when prevention is not possible, we may still prepare the system for the recovery so as to decrease recovery-associated downtime. For example, a cold spare might be started when a failure of a server is predicted. In power systems, a maintenance team may be directed to a specific location when a failure of a component is predicted. Also, the cost of false positive predictions must be taken into account as a mitigation triggered by a false alarm may cause unnecessary downtime.

Proactive management and predictive analytics have already found their way to the power grid and are used for predictive maintenance [65] and mainly load forecasting [66]. However, to the best of our knowledge, a comprehensive proactive approach that specifically aims at improving availability of Smart Grids, by predicting disturbances, has not been proposed to date.

1.4 Problem Statement

Massive introduction of renewable generation, market deregulation, new operation paradigms and other Smart Grid trends drive numerous changes in electric power systems, especially at the distribution level, that result in new challenges to Smart Grids dependability, and dependability of ADNs in particular. Hence, with power grid's digitalization we also need new dependability management approaches and methods.

Efforts towards more dependable grid, that in great part rely on ICT infrastructure, include, among others, new demand side management programs [8] for better power balance and real-time state estimation [29] for improved grid status awareness, and faster and better detection of disturbances and other problems. Complementing and extending these efforts, the goal of this thesis is to develop a methodology and to design methods for prediction of disturbances in ADNs using voltage sags as a case study. With this we want to pave the way for proactive management methods to additionally enhance availability of electric power delivery service, and thus to contribute to a more trustworthy grid. The motivation for adopting such an approach are numerous research results in failure prediction developed for industrial systems, such as, for example telecommunication systems [67], high-performance computing systems [68], as well as commercial computer systems that exploit predictions for enhancing availability including IBM XIV storage system [69], IBM predictive management [70] and HP Backup Navigator [71]. More examples of systems with proactive fault management may be found in [72].

The backbone of the approach is in the use of accurate grid-status data (e.g. from PMU devices whose number is steadily increasing) to predict a disturbance and to take proactive corrective actions (e.g. with a distributed storage, a controllable generator or a tap changer) to prevent it or to mitigate it.

1.4.1 Challenges

Even though it is appealing to assume that, with more ICT elements in power grids, proactive management methods, as used in computer systems, may be simply transferred to the new field of application, their adaptation comes with numerous challenges:

- *Identification and classification of Smart Grid faults and failures.* With growing complexity and interdependency between cyber and physical Smart Grid infrastructures and higher penetration of RES'es, the number of faults and disturbances is expected to increase but also new types of faults are expected to appear. This requires to identify and to classify faults and disturbances as a part of a comprehensive approach to Smart Grid dependability.

- *Determination of the efficiency of proactive approach for enhancing system's availability.* Despite encouraging implementations of proactive approaches in computer systems, there are still no models and figures of merit that would allow to evaluate to what extent, for which failure prediction quality, and under what conditions availability of a system may be enhanced with proactive (predictive) approaches. Such models and metrics must be developed to determine the efficiency of proactive approaches on system's availability in general before proposing their application for enhancing Smart Grid's availability.

- *Development of proactive methodology and methods for fault management in Smart Grids.* A proactive fault-management methodology has to be defined for Smart Grids and appropriate methods for prediction and mitigation of disturbances identified. This also includes acquisition of data for prediction algorithm training and evaluation.

- *Quantification of power delivery service availability and evaluation of proactive methods.* Comparing the existing and the proposed Smart Grid fault-management methods requires their evaluation with respect to availability enhancement that has to be quantified.

In addition to these challenges, the fact that the Smart Grid as a concept is constantly developing and improving brings new problems due to lack of dependability standards in Smart Grids and standardized models for system simulation that consider its cyber and physical aspects to allow evaluation of different management approaches. Another difficulty is that the data on disturbances in power grids are rarely, if at all, publicly available.

1.4.2 Goals and Objectives

To tackle these challenges and problems for the sake of achieving the goal of the thesis, we aim at meeting the following objectives that also drive the dissertation work-flow:

1. *Propose definitions for Smart Grid dependability attributes, develop taxonomy of faults and identify figures of merit for availability quantification.*

This should provide unambiguous communication platform between various Smart Grid communities and, along the lines of the thesis goal, it should also help in identifying new types of faults and proper figures of merit of grid dependability and availability in particular.

2. *Develop a model and metrics to evaluate to what extent availability of a system may be enhanced with a proactive approach and how this depends on the quality of prediction.* The model should be sufficiently general, easily applicable and understandable so as to provide better insights into the effect of a proactive failure management on availability of a system.

3. *Define a methodology for proactive management of disturbances in Smart Grid (and Active Distribution Networks in particular) and identify methods for the implementation.* As accurate and effective disturbance prediction is the core of the approach it deserves particular attention. This includes identification of requirements for monitoring infrastructure, identification and adaptation of methods for the selection of features and finally methods for the prediction (e.g. by adopting the ones reviewed in [72]).

4. *Develop and implement a simulation environment to synthesize disturbance-related data that can be used for disturbance analysis and prediction.* In the absence of field data, generation of disturbance-related data through simulation is an alternative. In this respect, fault injection, that is used in computer systems for the evaluation of fault-tolerance policies and prediction methods [73] may be employed. Simulations must be performed for relevant models (that also need to be defined) and for different settings including different load and generation profiles that simulate system dynamics, and behavior of the systems in the presence of different types of faults. In this regard, faults identified in the scope of the Objective 1 should be used.

5. *Case Study: Implement disturbance predictor and evaluate the approach for management of voltage sags in Active Distribution Networks.* Evaluate the approach defined in the scope of the Objective 3 for the case of prediction of voltage sags in distribution networks in the presence of distributed generation and variable generation and load. Use the framework implemented in the scope of the Objective 4 for the generation of voltage sag data. Identify an appropriate mitigation mean and quantify availability enhancement using the model from the Objective 2.

1.5 Contents and Contributions

The rest of the manuscript is organized as follows:

- In Chapter 2 first basic dependability concepts as well as modeling and evaluation methods and tools are introduced. This includes definition of dependability attributes and figures of merit, means to attain dependability, the fault-error-failure concept as well as a brief description of commonly used modeling and evaluation tools. Also, considering recent approaches based on prediction of disturbances, we present our extension of a taxonomy of fault-tolerance policies. Then, modeling Smart Grid as a cyber-physical system (CPS), a unified approach is taken to define its dependability by combining approaches and definitions from computer and power systems communities. A large set of previous blackouts (close to 40) has been analyzed to identify most common root causes, to identify their main properties and to conduct their classification in a form of a developed taxonomy. Finally, appropriate figures of merit for quantification of Smart Grid availability are identified.

- Related work is reviewed in Chapter 3. First a traditional approach to power grid dependability is explained followed by an overview of recent research in Smart Grid dependability modeling and evaluation. Proactive management approaches and prediction-based methods are then reviewed. The section concludes with a description of relevant Smart Grid pilot projects.

- In Chapter 4 the impact of predictive (proactive) management on availability is analyzed. First a generic model of a predictive management is defined considering the quality of failure prediction and the cost of disturbance mitigation actions. Then, a metric for optimizing failure prediction for enhanced availability is proposed and a sensitivity analysis of the approach with respect to model parameters is conducted. The main contribution of this part of the work is that the derived model and the metric may be used to identify if a proactive approach may be applied to the specific system and what the minimum prediction quality requirements are. Also, derived model and equations may be used to maximize availability when a proactive approach is used.

- A methodology and methods for online disturbance prediction are described in Chapter 5. Also a methodology, methods and tools for the design of a disturbance predictor are proposed.

- Developed framework for disturbance-related data generation is described in Chapter 6. The framework is based on simulation and fault injection following successful examples of implementing a similar approach for synthesizing failure data in computer systems for failure predictor training and evaluation. The framework is developed as fully modular so that system simulator, monitor and disturbance detector may be clearly distinguished. The final output of the framework is a set of time tagged and classified instances. An instance represents a set of values of selected variables (voltages, phasors, etc.). Each instance may be classified as disturbance-free or related with a specific disturbance. The generated data set must be further conditioned before being used for training a prediction algorithm. The same data set maybe used for training and evaluating online disturbance detectors.

- The case study is presented in Chapter 7. We evaluate if and to what extent voltage sags may be predicted in an Active Distribution Network (ADN). Using the developed simulation framework we first simulate behavior of an ADN in the presence of short-circuit balanced faults that cause voltage sags in different parts of the network. We record and condition the data so that it may be used with the classification-based machine learning algorithms. We then perform feature selection to identify the most indicative features and evaluate with what quality of prediction sags may be predicted. We also evaluate how the prediction quality varies with monitoring sampling period, lead time and the size of the prediction window. Finally, we optimize the prediction to maximize availability of an ADN and compare availability gain measured with downtime decrease.

- Chapter 8 concludes the work, reviews the contributions and gives directions for the future work.

Chapter 2

Terminology, Concepts and Taxonomies

As dependability is a mature field in computer systems with well-defined and widely-accepted terms and terminology, basic dependability concepts, as used in computer systems, are first introduced. This includes definitions of dependability attributes and figures of merit, description of means to attain dependability, the fault-error-failure concepts as well as a brief overview of commonly used modeling and evaluation tools. Also, considering recent approaches based on prediction of disturbances, we present our extension of a taxonomy of fault-tolerance policies.

Then, we present our unified approach to Smart Grid dependability that we modeled as a cyber-physical system (CPS). In fact, our unified model is built by combining approaches and definitions from computer and power systems communities. A large set of reported blackouts (close to 40) has been analyzed to identify the most common root causes, to identify their main properties and to conduct their classification by proposing a fault taxonomy. The proposed set of definitions and the taxonomy contribute to better communication between the two communities as a mutual understanding platform. Also, it gives a unique overview of the range of faults that may occur in Smart Grids and help to better understand Smart Grid dependability threats. Moreover, the taxonomy facilitates the application of dependability evaluation methods, such as fault injection.

Finally, appropriate figures of merit for quantification of Smart Grid availability are identified.

A part of the work presented in this section has been published [57].

2.1 Overview of Basic Dependability Concepts

Dependability is the ability of a system to perform a required service under stated conditions for a specified period of time. A dependable system is the one which delivers a required service during its lifetime.

As an umbrella term, dependability integrates the following attributes [74]:

- Availability: readiness for correct service,
- Reliability: continuity of correct service,
- Safety: absence of catastrophic consequences on the user(s) and the environment,
- Integrity: absence of improper system alterations, and
- Maintainability: ability to undergo modifications and repairs.

Security, that is mostly addressed separately, integrates availability, integrity and confidentiality (the absence of unauthorized disclosure of information) [74].

An important concept in dependability is the fault-error-failure one. If a failure occurs, delivered service deviates from the correct one. An error is a deviation of a system's state from the correct state, and a fault is a root cause of an error [74]. In addition, *Salfner*, *Lenk* and *Malek* distinguish between detected undetected and detected errors (an error is undetected as long as a detector does not identify an incorrect state) [72]. A failure of one component may also propagate to another components or cause a fault at higher levels of system hierarchy. A failure of a component may be a fault from the system's point of view as long as it does not cause deviation of the system's state.Once activated, the fault will cause an error that, if affecting the service provided to the user, propagates to a failure.

According to [74], means to attain dependability include:

- Fault prevention (also called fault avoidance or fault intolerance): prevention of occurrence or introduction of faults,
- Fault tolerance (FT): avoidance of service failures in the presence of faults. Also, capability to continue the correct operation in the presence of faults,
- Fault removal: reduction of the number and severity of faults, and
- Fault forecasting: estimation of the present number, the future incidence, and the likely consequences of faults as well as prediction of failures.

In this work, we focus mainly on fault tolerance. Considering novel techniques based on online prediction of failures, we extend taxonomy of fault-tolerance techniques, originally developed by Avizienis *et. al* [74]. The taxonomy is presented in Figure 2.1. Parts that are present in the taxonomy from [74] are written in italics, whereas classes of techniques are presented in rectangles.

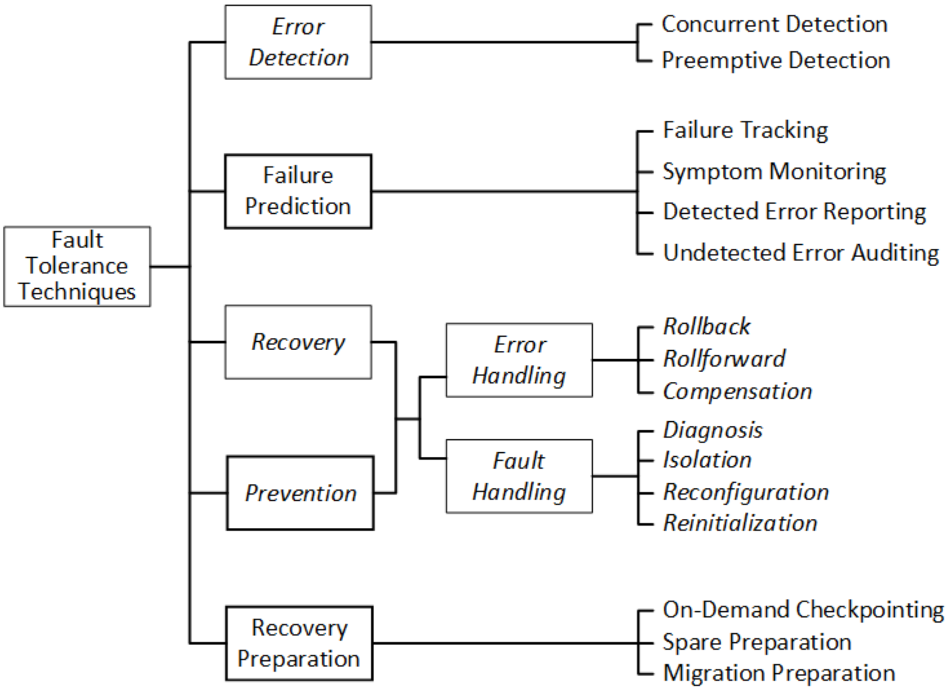


Figure 2.1. Taxonomy of fault-tolerance techniques.

Error detection may be performed concurrently or by interrupting the main process to check the system for errors. In concurrent error detection techniques, for example, health-status (as in [75]) or the system load (as in [76]) may be estimated by online monitoring system parameters (also called features or variables). For a computer system and ICT components these parameters may be: memory usage, CPU activity and temperature, disk activity, number of exceptions or fan speed. For a power systems and its components these parameters may be voltages, phasors, and component temperatures. Failure prediction may be performed by: failure tracking, symptom monitoring, detected error reporting, and undetected error auditing [72]. The same or a similar set of error- and fault-handling techniques that are identified in [74] as system recovery techniques, may be used to prevent errors and failures when triggered based on the results of the state estimation or failure prediction. When a failure is anticipated, the system can be prepared for a recovery. In computer systems this may be done by creating a checkpoint on-demand, preparing a spare components, performing a failover or by applying similar techniques. In power systems, for example, a spare dispatchable distributed generator may be preventively started in anticipation of a voltage drop.

By combining FT techniques different FT techniques may be defined. In general, fault-tolerance policies may be reactive, proactive-reactive, or proactive depending if an action is taken after a failure has been detected, before a failure (preventively) or in anticipation of an upcoming failure. Considering only techniques originally presented in [74] two types of reactive fault-tolerance policies may be defined: detection and recovery and masking and recovery [1]. In the first one, recovery actions are triggered on demand, when a failure of a component is detected. In the second policy, errors (component failures) are systematically masked (for example with redundancy) and recovery is performed on demand when an error is detected. Note that different implementations of these basic policies may also include additional, proactive actions that are performed systematically. For example, periodic checkpoints are created as a part of the rollback and recovery policy that is one possible implementation of a detection and recovery policy type.

Proactive-reactive policies assume taking actions to prepare the system for more efficient recovery if a failure occurs. A typical example of preparation and recovery policy is checkpointing. Even though a failure will not be fully avoided, recovery time may be shortened and availability improved.

Taking into account all FT techniques from Figure 2.1, we identify additional types of FT policies that we presented in a form of a taxonomy in Figure 2.2. Proactive policies can further be categorized as systematic or predictive. Systematic policies include those where actions are taken periodically, with a period defined statically or adjusted dynamically as in [76], depending on the system status indicators as, for example, system's health and load distribution (e.g. as in [77]), or after specific events (e.g. a completion of a task or a preemptive error detection). For example, a migration to a spare server (failover) may be performed periodically to rejuvenate the original server. Predictive policies are based on online failure prediction and the use of prevention or preparation techniques to avoid a failure or to prepare a system for recovery. In the first case, failure-related downtime may be eliminated by failure avoidance. In the second

[1] A system with a masking-and-recovery policy reacts on errors (and from this perspective is reactive) to prevent them from propagating into failures (from this perspective it is proactive). Failure prevention in this case is based on redundancy and is implicit by the system design. During system life, there is no need to take any actions before a failure to prevent it. This is the main difference with respect to proactive policies as defined here and the reason for classifying masking and recovery as a reactive policy. Another view, by which masking and recovery would be classified as passive-proactive and other proactive policies considered here as active-proactive, is also possible.

case, error detection and recovery techniques still have to be used but the downtime caused by the failure and the system recovery will usually be minimized by the preparation. Based on this, we identify two types of predictive policies: (a) prediction and avoidance, and (b) prediction and preparation.

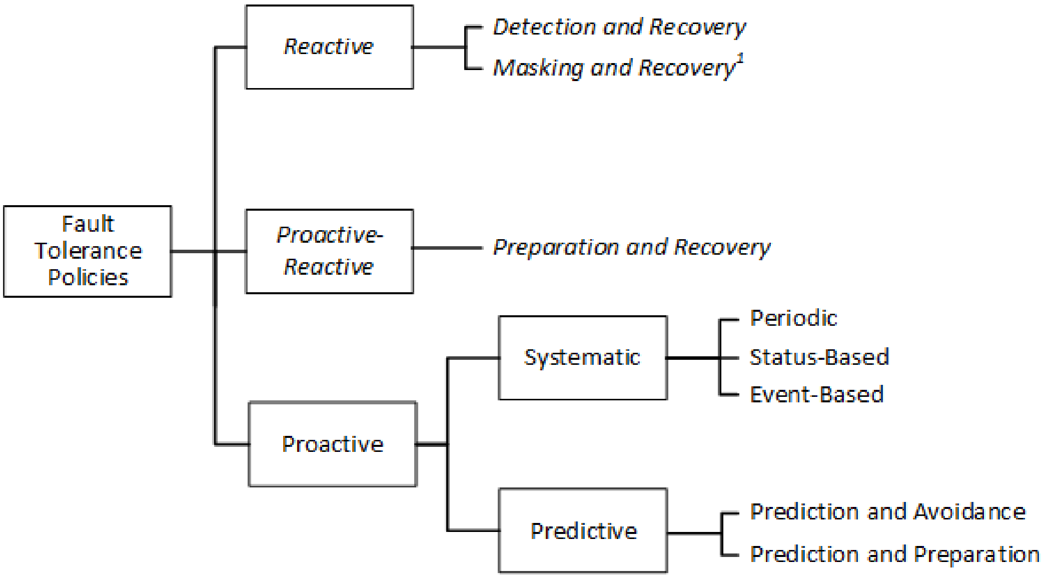


Figure 2.2. Taxonomy of fault-tolerance policies.

2.2 Overview of Modeling and Evaluation Methods and Tools

A brief overview of dependability modeling and evaluation methods and tools is given. Interested readers are referred to [77, 78, 79] for more details. In general, dependability models may be classified as component based and state based. The first ones focus on system building blocks and assume that failures and repairs of components are stochastically independent. The most commonly used models of this type are Reliability Block Diagrams (RBDs) and fault trees. An RBD reflects the structure of the system and each component is described with its mean-time-to-failure (MTTF) and mean-time-to-repair (MTTR). MTTF is a mean time between two consecutive failures and MTTR is a mean time needed for the component repair. Failure and repair rates, that are usually denoted by λ and μ, are defined as inverses of MTTF and MTTR, respectively. A fault tree is a top-down graphical representation of the system's structure formula so that the probability of higher level events can be calculated by combining probabilities of the lower level events.

In state-based models system states are modeled, thus interdependencies between component failures may be incorporated as well. The most commonly used ones are Markov Chain Models and Stochastic Petri Nets. Markov Chains consist of states, transitions between the states, transition rates and initial state probabilities. Each state may be either operational or a failure state. It is assumed that the transition rates are exponentially distributed. The main problem with Markov Chains is that the size of the model grows exponentially with the number of components. In Petri nets, that may handle larger systems better, a system is modeled as a directed bipartite graph.

Both types of models may be solved analytically or with commercial and academic tools. ReliaSoft[2] offers a suite of dependability modeling tools that are widely used for commercial purposes. Two most commonly used tools in academia are Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) and Stochastic Petri Net Package (SPNP). SHARPE [80] is a general hierarchical modeling tool that analyzes stochastic models of reliability, availability, performance, and performability. SPNP [81], allows modeling of complex system behaviors with advanced constructs.

In this work, we mainly use Markov chains that are solved in SHARPE that also incorporates plugins for model sensitivity analysis with respect to different parameters. More details are given in appropriate sections.

2.3 Dependability of Smart Grids

From the infrastructure point of view, Smart Grid is the advanced electric power grid with increased usage of cyber instruments, such as intelligent sensors, smart meters, communication infrastructures, control algorithms, and applications, for better management, higher efficiency, and increased dependability. Thus, it may be interpreted as a cyber-physical system (CPS) that combines information and communication technology (ICT) with physical elements and processes underneath the electric power delivery service. Electric components (conductors, capacitors, transformers, circuit breakers, etc.) that compose the power grid react on various events in the grid, by the rules of physics, and impose changes that also affect cyber infrastructure. Moreover, some modern components (e.g. smart meters) may also contain logic to autonomously react to grid events. Thus, the

[2] ReliaSoft, http://www.reliasoft.com/products.htm

control of the system and changes in the system are not unidirectional and governed only by the cyber infrastructure, but rather bidirectional, due to influences and interdependencies between physical and cyber elements [51]. As for that, Smart Grid cannot be modeled and analyzed by separately considering cyber and physical infrastructures. Many researchers tackle this problem by focusing exclusively on interdependencies (see, for example [51, 82, 83, 84]) but there is still a need for a unified approach that considers all aspects of Smart Grids as cyber-physical systems.

The main challenge for developing a unified approach is that power engineering and ICT communities are both well established and have their own terminologies, methodologies, models and approaches applied at different stages of the design and implementation of a system. Even though similar models, such as RBDs, faults trees and Markov chains are used for reliability modeling and evaluation in both communities, approaches to dependability are essentially different. This also applies to different definitions of dependability attributes, the way that faults, errors and failures are distinguished, different qualitative and quantitative measures and fault taxonomies. While significant effort has been made to unify definitions of dependability attributes in electric power systems (see, for example, [85, 86, 87]) cyber aspects of the system are still not sufficiently addressed.

The looming danger of merging ICT and power systems is that system's dependability may be compromised due to these differences.

2.3.1 Terminology of Smart Grid Dependability Threats

Faults, errors and failures are clearly distinguishable in ICT community as defined in [74] and reviewed in Section 2.1. On the other hand, a number of terms, such as incidents, perturbations, disturbances, disruptions, events, losses, adversities, emergencies, changes, anomalies, threats, shocks, and hazards are used in power systems to denote "disturbing events", such as voltage sags and swells, line trips, frequency variations, etc. [50]. Most of these disturbing events that affect power quality, may be categorized as errors but, depending on the category of the user and the level of a disturbance, they may, from the user perspective, be observed as failures as well. For instance, a relatively minor voltage variation can stop a highly sensitive industrial processes, while on the other side it may not even be noticed as a disturbance in a household. Thus, the same disturbance will be classified as a failure in the former case and as an error in the later one. In power systems, a failure is mainly used to describe a complete outage, when no service is delivered to one or more users. Blackouts represent the most severe system failures and the total collapse of the system that causes interruption of the service that affects all the users in an entire area [88]. It is also important to notice that system failures are events that are observable by the customer. Thus, if one infrastructure fails (for example, cyber infrastructure), but power is still delivered to the customer, there is no failure of the entire system.

2.3.2 Smart Grid Dependability Attributes

To better understand dependability of Smart Girds, as cyber-physical systems, in Table 2.1 we compare the most widely accepted definitions of major dependability attributes and means as used in computer systems dependability (CSD) and electric power systems (EPS) communities. Still, it should be kept in mind that these terms are sometimes interpreted differently within the two communities and that it may be difficult to clearly distinguish between some dependability attributes or that their definitions overlap to some extent (for example, robustness and resilience in CSD). In addition, some attributes are frequently defined differently over the power systems community. For example, reliability and security are defined in different ways in different fields within EPS community. Resilience is sometimes mixed with robustness which characterizes a system's ability to cope with a specific class of faults [50], and availability is addressed as reliability when defining power systems reliability figures of merit in [85].

Still, having in mind definitions in Table 2.1, availability and reliability are generally accepted similarly in both communities and those definitions could be used when addressing Smart Grid dependability. Understanding of maintainability in the two communities is also similar. Interestingly, fault tolerance is not a common term in the EPS community but fault-tolerance aspects of the system are addressed as the ability of the transmission network to keep operational after a failure of k out of N lines is known as the N-k problem [90, 91].

In the CSD community security is defined as the system property that integrates confidentiality, integrity, and availability. Considering cases of grid failures due to cyber-attacks as, for example, those reported in [55] and [95], implies that security as defined in CSD community should be included as a dependability attribute in Smart Grids as well. On the other hand, term security in EPS community is used to describe system's alternative configurations (that is closer to the meaning of fault tolerance or redundancy in CSD) but also system's ability to smoothly make a transition to these alternative configurations in case of errors [10]. This second aspect is considered as a part of fault-tolerance (resilience) in CSD community. Thus, a security as defined in EPS community is covered by resilience and redundancy. Security aspects of ICT Smart Grid infrastructure

Table 2.1. Major dependability attributes and means as defined in computer systems dependability (CSD) and electric power systems (EPS) communities.

Relibility	CSD	Continuity of the correct service. The probability that the system will perform satisfactorily from time zero to time t, given that operation commences successfully at time zero [74].
	EPS	Continuity of electric service to customers, which depends both on the availability of sufficient generation resources to meet demand and on the ability of the transmission and distribution system to deliver the power [10]. In a bulk power electric system: The degree to which the performance of the elements of that system results in power being delivered to consumers within accepted standards and in the amount desired. The degree of reliability may be measured by the frequency, duration, and magnitude of adverse effects on consumer service [86].
Availability	CSD	Readiness for the correct service. The probability that a system is performing correctly at time t [74]. Steady-state availability is the fraction of time a system is operational during its expected lifetime.
	EPS	The probability that a system is performing its required function at a given point in time used under stated operating conditions [89].
Fault Tolerance	CSD	Avoiding service failures in the presence of faults [74].
	EPS	The term is usually not used directly. The N-k problem [90] that is considered as a part of reliability standards is the closest to the understanding of fault tolerance in CSD. It is usually applies to the transmission grid and represents the ability of an N-line transmission grid to tolerate a failure of k lines [91].
Resilience	CSD	Persistence of service delivery that can justifiably be trusted, when facing changes (unexpected failures, intrusions, accidents or increased load) [92, 93].
	EPS	System's ability to reduce the magnitude and duration of the disruption [50]. The ability of the system to recover from catastrophic failures and return to a state where it is considered functional [94].
Security	CSD	Integrates confidentiality (absence of unauthorized disclosure of information), integrity (absence of improper system alterations), and availability and requires their concurrent existence [74].
	EPS	The ability of the system to withstand disturbances [89]. A measure of the width of the operating envelope, or set of immediately available operating configurations that will result in a successful outcome (no load interruption and no equipment is damage) [10].
Maintenance	CSD	Ability to undergo modifications and repairs [74].
	EPS	Activity wherein a fault-free device has, from time to time, its deterioration arrested, reduced or eliminated [87].

are still addressed in EPS community particularly after September 11, 2001 and are referred to as cyber-security. In fact, numerous organizations have proposed guidelines and standards to address cyber-security of electricity, telecommunications, transportation and other critical infrastructures [16]. However, these are mainly addressing confidentiality, data redundancy, and protection of critical functions from cyber-attacks but do not include availability and integrity to a sufficient extent.

Finally, the aspects of resilience as defined in EPS community are considered as a part of fault tolerance and maintenance in CSD.

In addition, stability is an important dependability attribute used in power systems community. In a nutshell, it describes the ability of a power system to maintain a synchronous and balanced operating state [10]. A more comprehensive definition is given in [86] where power system stability is described as "the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact." In the same paper, a classification of power system stability, that is presented in Figure 2.3, is performed. Definitions of power stability classes, given in the rest of the subsection, are also adopted from [86].

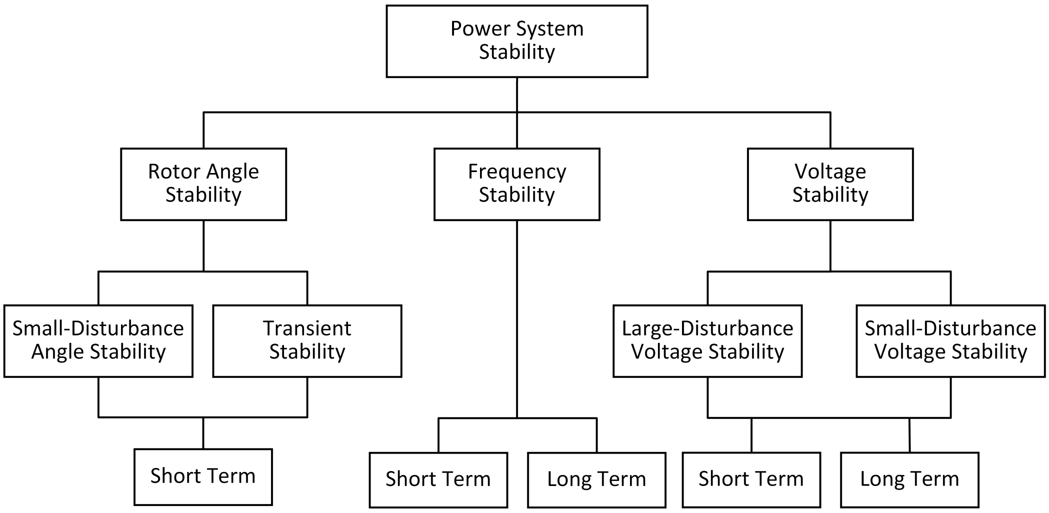


Figure 2.3. Power system stability classification.

Rotor angle stability is the ability of synchronous machines of an interconnected power system to, after being subjected to a disturbance, maintain/restore equilibrium between electromagnetic and mechanical torque of each synchronous machine in the system and thus to remain in synchrony. Small-disturbance rotor angle stability is usually associated with insufficient damping of oscillations, whereas transient (also called large-disturbance) rotor angle stability is associated with severe disturbances, such as a short circuit on a transmission line. Both types of instabilities are typically short-term events that may last for a few seconds to up to 20 seconds for the case of a transient instability in large systems.

Frequency stability is the ability of the system to maintain steady frequency following a severe system disturbance that results in a significant imbalance between generation and load. Frequency instability is usually related to insufficient generation reserve. Short frequency instabilities last for a few seconds and may be related, for example, to islanding where one of the islands does not have sufficient generation reserve. Long-term frequency instabilities may be related to a steam turbine overspeed control or a boiler/reactor protection. They typically last from tens of seconds up to a few minutes.

Voltage stability is the system's ability to maintain steady voltages at all system buses after being subjected to a disturbance, that depends on the ability to maintain/restore equilibrium between load demand and supply. Voltage instability may occur as a progressive fall or rise of voltages of some buses (voltage sags and swells) that may also lead to a loss of load in an area or to a voltage collapse in a significant part of the grid. Voltage drops, which are more frequent, may be associated with a load increase, a short on a distribution bus or a rotor angle instability. Voltage swells are usually related to system's inability to operate below a certain load level so that generation remains significantly higher than demand. Large-disturbance voltage stability is related to system faults, loss of generation, or circuit contingencies. On the other hand, small-disturbance voltage stability is related to small perturbations such as incremental changes in system load. Short-term events last for a few seconds and involve dynamics of fast acting load components such as induction motors, electronically controlled loads, and high-voltage-direct-current converters. Long-term instabilities may last for several or many minutes and, typically, involve slower acting equipment such as tap-changing transformers, thermostatically controlled loads, and generator current limiters.

Thus, having in mind these definitions, for defining main dependability attributes of the Smart Grid we propose to use definitions of these attributes as defined in computer systems' dependability while adding stability as an additional attribute.

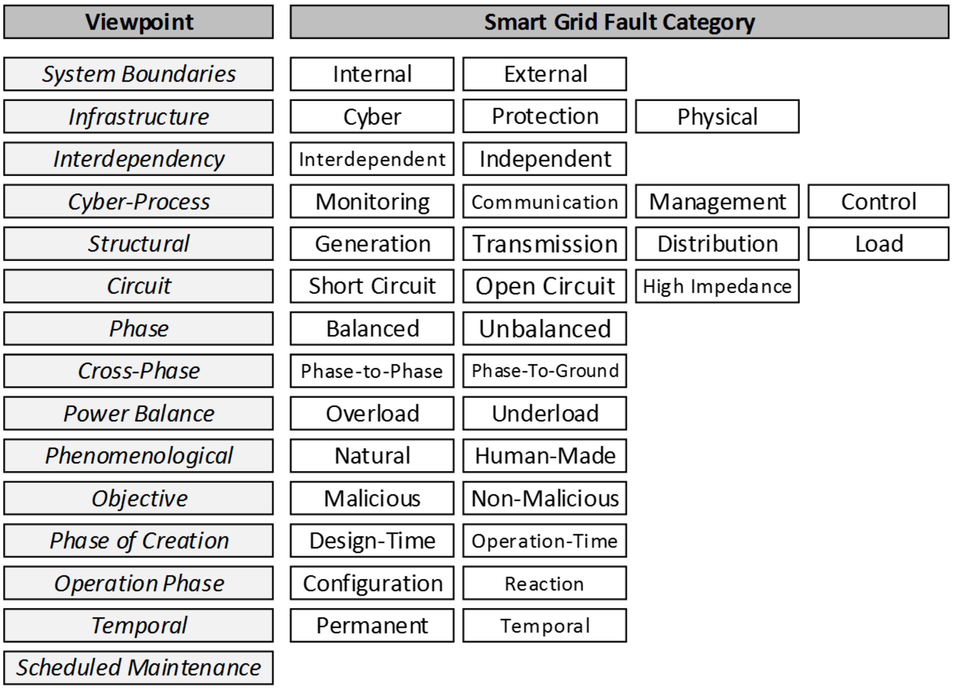
2.3.3 Taxonomy of Smart Grid Faults

Electric power system outages have been extensively analyzed, in both academia and industry, in order to identify root causes and sequences of events that lead to system failures (see, for example, [96] and [97]). We provide a holistic overview of these root causes (faults) by identifying their main properties and conducting fault classification. The taxonomy is intended to help better understanding of the Smart Grid dependability and defining which classes of faults should be included in system's dependability specification. Moreover, a taxonomy may be used when modeling and evaluating system's fault-tolerance with, for example, fault-injection methods. As a starting point, a taxonomy of faults for computing systems defined in [74] is used. We extend this taxonomy, include power system specific and interdependency faults while focusing on fault classes that were root causes of previous blackouts. Some classes of faults that are purely related to the computer (cyber) infrastructure and are included in [74] are omitted in our taxonomy for the simplification reasons.

We have analyzed a number of power system failure reports and publications related to Smart Grid dependability including [96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113] to identify faults responsible for the past outages and possible causes of future failures, and to define a minimum set of properties for the classification of faults. Faults are classified with respect to 14 different viewpoints in total, where each viewpoint differentiates among up to four fault categories. Each combination of categories from different viewpoints defines one class of faults, but not all combinations of categories are possible. An overview of all viewpoints and categories is presented in Figure 2.4.

Taking a *System Boundaries Viewpoint*, faults are categorized as internal or external. Internal faults are those of system components and other faults that originate within the system boundary (e.g. an operator fault). External faults are faults that originate outside the system boundary. In power systems, a vast number of components' failures are due to bad weather conditions, electrification of animals, vandalism or other external faults [97]. Cyber-attacks are another type of external faults that is expected to be more common in the future, as the power grid is advancing towards the Smart Grid [55].

Infrastructure Viewpoint considers infrastructures of the Smart Grid, namely cyber and physical. In addition, we consider protection part of the physical infrastructure as a separate one to distinguish between faults that occur in the part of the system that is purely in charge of power delivery and the part related to system dependability. Once activated, these faults will result in errors and, possi-



Viewpoint	Smart Grid Fault Category			
System Boundaries	Internal	External		
Infrastructure	Cyber	Protection	Physical	
Interdependency	Interdependent	Independent		
Cyber-Process	Monitoring	Communication	Management	Control
Structural	Generation	Transmission	Distribution	Load
Circuit	Short Circuit	Open Circuit	High Impedance	
Phase	Balanced	Unbalanced		
Cross-Phase	Phase-to-Phase	Phase-To-Ground		
Power Balance	Overload	Underload		
Phenomenological	Natural	Human-Made		
Objective	Malicious	Non-Malicious		
Phase of Creation	Design-Time	Operation-Time		
Operation Phase	Configuration	Reaction		
Temporal	Permanent	Temporal		
Scheduled Maintenance				

Figure 2.4. Viewpoints and categories for faults classification.

bly, failures in one of the infrastructures. As described, for example in [51], these failures may propagate and cause errors and failures in other infrastructures finally driving the system to fail. For example, during a power system blackout in the Northeast of the United States on November 9, 1965 incorrectly set relay has tripped a transmission line during the load increase before the maximum capacity of the line was reached [96, 97]. An incorrect setting of the relay that belonged to the protection infrastructure, was a fault that, after activation, resulted in an error and a failure of the relay. This failure propagated to an error and a failure of the line that belonged to the physical infrastructure. Thus, in this case, there was no fault in the physical infrastructure responsible for the power delivery but a failure propagated from the protection infrastructure. It is important to notice that this, initial series of events, does not necessarily cause a system failure.

In addition, events in one infrastructure, that are not necessarily failures or errors, may cause or activate faults in another infrastructure. We refer to these faults as interdependent faults and define *Interdependency Viewpoint* to classify faults as interdependent or independent. For example, going back to the same Northeast USA blackout [96], the event in the physical infrastructure, namely an increase of the line load has activated a dormant fault in the protection infrastructure, in fact an incorrectly set relay. Similarly, a cyber-attack may cause a failure in ICT infrastructure that may propagate to protection or physical infrastructures. In more generic terms, if a system is composed of two infrastructures, A and B, from the perspective of an infrastructure A and not the entire system, events that originate in another infrastructure B and affect infrastructure A by creating or activating faults are external events. From the perspective of the entire system, they are interdependent events and the caused or activated faults are interdependent faults. Further classification of interdependent faults would be also possible following a classification of interdependent failures from [51] and [114].

Four main processes performed by the cyber infrastructure are monitoring, communication, grid management (i.e. data processing and decision making) and control. Each of these processes are supported by a proper hardware and software elements that may contain a fault. *Cyber-Process Viewpoint* considers this aspect to categorize faults. Further classification of faults as hardware and software faults, that is a part of taxonomy presented in [74], is omitted for the simplification reasons.

Electric power system structure is typically composed of four main elements: electrical power generation, transmission and distribution and loads [10]. A fault may originate in any of these parts and also propagate to others. In *Structural Viewpoint*, faults are categorized considering this aspect. For example, a fault of a relay in Northeast USA blackout [96] would be categorized as a transmission fault as the relay in question physically belonged to the transmission network. Loads are usually not under the control of a network operator but are still a part of the system and thus, load faults that may create errors and failures of the power system (e.g. making the grid unbalanced and causing voltage or frequency fluctuations) are considered as internal faults and also included in this viewpoint.

In traditional reliability analysis of power systems mainly three types of faults are considered: short circuit faults, open circuit faults and high impedance faults [115, 116]. An open circuit fault occurs when the current flow is interrupted. A short circuit fault (short) occurs when a connection is established between the phases, or a phase and a ground. If these faults cause variations of voltage or current, they will also result in errors. A high impedance fault occurs when an energized conductor gets in contact with the ground or other environmental objects like a tree branch. These faults are not detectable by the protection infrastructure as shorts, due to high impedance and relatively small current flow, but represent a serious safety issue. Furthermore, short circuit and high impedance faults may occur between a phase and a ground or between the phases. If all three phases are affected symmetrically, a fault is balanced, otherwise it is unbalanced [115]. These aspects of faults' categorization are considered in *Circuit,* *Phase and Cross-Phase Viewpoints*.

A sudden increase or decrease of the load demand or a loss of a number of loads will make the system unbalanced and, in some cases, may propagate into a failure. Similarly, a sudden uncontrollable decrease of generation, for example in the case of renewable resources, may cause the same effect. Finally, insufficient spinning reserve might also get the system in unbalanced, overload, state when the demand increases [96]. *Power Balance Viewpoint* classifies faults as overload and underload faults. An overload fault occurs when the generation or reserve cannot meet the load demand and an underload fault occurs when generation is much higher than demand. Furthermore, an overload or underload may affect active or reactive power.

A large fraction of failures in power systems is caused by external natural events such as severe weather conditions, lightening, tree contacts, and animals' electrification [97]. Other natural causes include aging of the equipment, electromagnetic phenomena and cosmic radiation that may affect computer and other electronic infrastructure [74, 115]. Opposite to this, many faults may be caused by humans during any phase of system's lifetime including design, installation, assembly, inspection, operation and maintenance [117]. From the *Phenomenological Viewpoint*, faults are categorized as natural or human-made faults. Human-made faults may be further categorized depending on the phase of creation, objective, intention and capability.

For the sake of this taxonomy we take a simplified approach, and distinguish only between malicious and non-malicious human-made faults in the *Objective Viewpoint*. Malicious faults include acts of vandalism, terrorism and sabotages [117] that mainly affect physical infrastructure, as well as cyber-attacks [55]. Non-malicious faults include designers' or operators' faults or external human-made faults. Further categorization of human-made faults considering intent and capability is conducted in [74].

Faults may be introduced during any phase of system's life time, namely design and production time, runtime or maintenance time. In the *Phase of Creation Viewpoint*, we differentiate between faults that were introduced during the design/implementation phase and operation phase. Design phase faults include faults due to bad decisions, bad design, implementation bugs (e.g. software bugs), etc. but also faults introduced during the production and implementation phase. Operation faults include all faults introduced at runtime, scheduled maintenance or recovery period.

Moreover, human-made faults made at operation phase are further categorized as configuration faults and reaction faults in the *Operation Phase Viewpoint*. Configuration faults are introduced by an operator or a software and include incorrect setting of device parameters during normal operation or maintenance. Reaction faults are introduced at runtime when an operator or a software reacts improperly in a given situation or on an occurring failure and does not prevent its propagation. For example, an incorrectly configured relay is a configuration fault. If an operator misses the opportunity to rebalance the grid when a transmission line trips, a reaction fault is activated.

Finally, faults may be permanent or temporal. Permanent faults are continuous in time. For example, a natural fault during the production process that creates a short within an electric device is a permanent fault. Temporal faults are bounded in time. For example, a branch that connects to a transmission line during a stormy weather is a temporal fault. This aspect is considered with *Temporal Viewpoint*. Furthermore, in CSD, temporary faults are split between intermittent (caused within a component) and transient (caused by environment). Both categories might be aperiodic (occur once) or periodic.

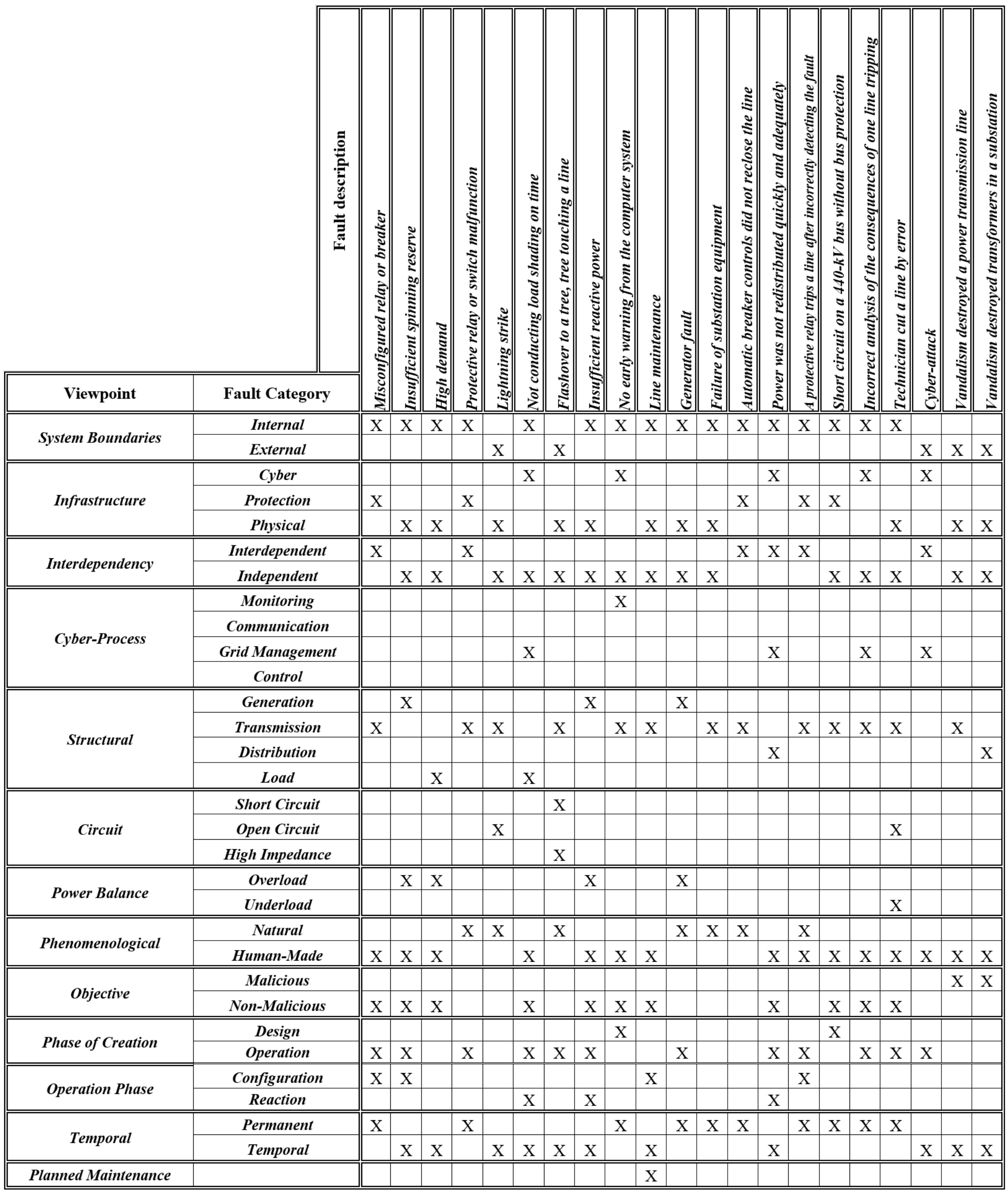
In addition to this, it is important to notice that planned (scheduled) maintenance frequently weakens the system. In combination with activation of other faults, this may lead to the system failure. For example, in 2003 a high-voltage direct current (HVDC) link between the northern Europe and the rest of Europe was out of service for a scheduled maintenance when a 1200 MW nuclear unit in southern Sweden went off due to a failure of a steam valve. In combination with other, unrelated events that occurred shortly after, this resulted in a total loss of 6550 MW of load affecting 4 million people in Sweden and Denmark [97, 99]. A series of events that occurred during blackouts in India in July 2012 also involved scheduled maintenance [98]. These blackouts might have been prevented, if the system was not under the maintenance when faults become active. Keeping in mind the impact that scheduled maintenance may have on system's dependability, and its role in previous blackouts, we include it in the fault taxonomy as well, but acknowledge that scheduled maintenance is not a fault.

We demonstrate how events during a blackout should be categorized, following the taxonomy, on the example of the North American blackout that occurred on August 14, 2003, and that affected nearly 50 million people over USA and Canada with the total load capacity loss of 63 GW. The initiating event was the insufficient reactive power. This fault is classified as an internal fault of physical infrastructure at generation level, human-made (due to the lack of reaction), temporal, reaction fault at operation phase. In this period, state estimator, real-time contingency analysis, energy management and control center software failed due to activation of independent faults. All these faults are categorized as internal, cyber faults of monitoring (or management) process at transmission level. Moreover, they are human-made, non-malicious introduced at design phase and permanent. Generator outage that followed shortly after is an internal fault of physical infrastructure at generation level and so on. Line trips due to tree contact that also occurred during this blackout, are an example of an external fault. Observed voltage variations before the outage would be categorized as errors, while the failure of the system occurred when the first group of users was cut off from the power supply.

Following the same procedure we have analyzed a large set (close to 40) of major power outages to identify and classify faults. A subset of major blackouts is presented in Table 2.2. The table is sorted according to the total number of people affected by a blackout. In the analysis, we rely on relevant publications or the official post-mortem reports and official announcements of electrical power supply companies. A classification of the most representative and frequent faults is presented in Figure 2.5.

Table 2.2. The most severe recent blackouts that were analyzed.

Date and place of the outage	**People affected**	**Duration**	**Capacity loss**	**References**
July 31, 2012, North and East India	700 M	12 h	-	[98]
August 14, 2003, USA and Canada	50 M	-	63 GW	[96, 97, 99, 100]
September 28, 2003, Italy and Switzerland	60 M			[96, 97, 99]
November 4, 2006, Europe	15 M	2 h	-	[101]
March 27, 2015, Netherlands (Amsterdam)	3 M	<5 h		[56]
September 23, 2003, Sweden and Denmark	4 M	-	6.5 GW	[96, 97, 99]
September 8, 2011, USA	2.7 M	12 h		[105]
August 14, 2006, Tokyo, Japan	1.4 M	-		[107]
January 16, 2007, Victoria, Australia	480 K		2.2 GW	[96, 110]
October 6, 2013 Arkansas, USA	10 K	-	-	[111]



	Fault description																					
Viewpoint	**Fault Category**	*Misconfigured relay or breaker*	*Insufficient spinning reserve*	*High demand*	*Protective relay or switch malfunction*	*Lightning strike*	*Not conducting load shading on time*	*Flashover to a tree, tree touching a line*	*Insufficient reactive power*	*No early warning from the computer system*	*Line maintenance*	*Generator fault*	*Failure of substation equipment*	*Automatic breaker controls did not reclose the line*	*Power was not redistributed quickly and adequately*	*A protective relay trips a line after incorrectly detecting the fault*	*Short circuit on a 440-kV bus without bus protection*	*Incorrect analysis of the consequences of one line tripping*	*Technician cut a line by error*	*Cyber-attack*	*Vandalism destroyed a power transmission line*	*Vandalism destroyed transformers in a substation*
System Boundaries	*Internal*	X	X	X	X		X		X	X	X	X	X	X	X	X	X	X	X			
	External					X		X												X	X	X
Infrastructure	*Cyber*						X			X					X			X		X		
	Protection	X			X									X		X	X					
	Physical		X	X		X		X	X		X	X	X						X		X	X
Interdependency	*Interdependent*	X			X									X	X	X				X		
	Independent		X	X		X	X	X	X	X	X	X	X				X	X	X		X	X
Cyber-Process	*Monitoring*									X												
	Communication																					
	Grid Management						X								X			X		X		
	Control																					
Structural	*Generation*		X						X			X										
	Transmission	X			X	X		X		X	X		X	X		X	X	X	X		X	
	Distribution														X							X
	Load			X			X															
Circuit	*Short Circuit*							X														
	Open Circuit					X													X			
	High Impedance							X														
Power Balance	*Overload*		X	X					X			X										
	Underload																		X			
Phenomenological	*Natural*				X	X		X				X	X	X		X						
	Human-Made	X	X	X			X		X	X	X				X	X	X	X	X	X	X	X
Objective	*Malicious*																				X	X
	Non-Malicious	X	X	X			X		X	X	X				X		X	X	X			
Phase of Creation	*Design*									X							X					
	Operation	X	X		X		X	X	X			X			X	X		X	X	X		
Operation Phase	*Configuration*	X	X								X					X						
	Reaction						X		X						X							
Temporal	*Permanent*	X			X					X		X	X	X		X	X	X	X			
	Temporal		X	X		X	X	X	X		X				X					X	X	X
Planned Maintenance											X											

Figure 2.5. Classification of common Smart Grid faults.

A particular challenge for the analysis was that not all the events are reported in a uniform way and in a sufficient level of detail. As for that, a number of different sources were combined when analyzing some of the events. Still, in many cases some types of categorizations were not possible to conduct due to lack of relevant information and reliable sources. For example, most of the reports do not indicate if a fault on the transmission or distribution level was balanced or not. Also, most of the time, it is not reported if a fault has occurred between the phases or between a phase and the ground. For that reason, Phase and Cross-Phase viewpoints are omitted in the table.

2.3.4 Figures of Merit of Smart Grid Availability

Metrics for quantifiable dependability attributes as used in computer systems are given in [74]. Our interest is mainly in availability of power delivery service and thus we focus on availability metrics. More metrics may be found in [57].

Steady-state availability is the most commonly used to evaluate service availability in computer systems. It is the ratio of system's uptime over the lifetime. As expressed in Equation 2.1, this may be translated in a more frequently used form with system's Mean-Time-To-Failure (MTTF) and Mean-Time-To-Repair (MTTR).

$$Availability = \frac{uptime}{lifetime} = \frac{uptime}{uptime + downtime} = \frac{MTTF}{MTTF + MTTR} \tag{2.1}$$

Even if the notion of user-perceived availability is sometimes used in computer systems as different users may perceive service differently [118], to the best of our knowledge, no user-perceived availability metrics are proposed to the date for electric power systems.

For evaluating availability of Smart Grids, metrics proposed in [85] (even though reliability and availability are interchangeably used in the source) seem to be more appropriate. The most commonly used ones are System Average Interruption Frequency Index (SAIFI), System Average Interruption Duration Index (SAIDI), Customer Average Interruption Duration Index (CAIDI) and Average Service Availability Index (AIDI). These are the metrics that may be used when evaluating the effect of the proactive or other approaches on the electric power delivery service availability. Expressions for these metrics are given in Equations 2.2 to 2.5. In addition, to evaluate availability of the entire Smart Grid, downtime may be used a metric.

$$SAIFI = \frac{TotalNumberofCustomersInterrupted}{TotalNumberofCustomersServed} \tag{2.2}$$

$$SAIDI = \frac{\sum CustomerMinutesofInteruption}{TotalNumberofCustomersServed} \tag{2.3}$$

$$CAIDI = \frac{\sum CustomerMinutesofInteruption}{TotalNumberofCustomersInterrupted} \tag{2.4}$$

$$ASAI = \frac{CustomerHoursServiceAvailability}{CustomerHoursServiceDemand} \tag{2.5}$$

For individual users we may also adopt and use the above metrics from the user perspective leading to a set of user-perceived availability metrics.

Finally, downtime may be adopted to effectively evaluate availability of a part of the power grid from the system's or from a user's perspective. In that sense a system may be considered as unavailable (down) whenever not providing a proper power quality to a user. It should also be kept in mind that the minimum required quality may depend on the type of a user (e.g. industrial vs. a private household).

Chapter 3

Related Work

Management of Smart Grids for enhanced dependability and better efficiency attracts considerable attention in both, industry and academia. Relevant papers and project reports are reviewed in this section. First a traditional approach to power grid dependability is explained followed by an overview of recent research in Smart Grid dependability modeling and evaluation. Proactive management approaches and prediction-based methods are then addressed. The section concludes with a description of relevant Smart Grid pilot projects.

3.1 Traditional Approach to Power Grid Dependability

Power system's dependability is traditionally addressed mainly through power grid protection (e.g. special protection schemes) [88, 115]. A protection system is composed of relays and circuit breakers that should detect electric (physical) failures and isolate a failed section to prevent failure propagation. These components may also incorporate logic and make local decisions that will affect the network topology. For example, a relay may detect a short circuit (a fault) but also a voltage sag. Depending on the settings it may only store or send the information for further analysis or automatically isolate a part of the grid to prevent disturbance propagation.

When addressing dependability (in fact mostly stability as one of its attributes) of power systems, power engineers mainly have in mind different operating states of the system that reflect its current stability. These states are depicted in Figure 3.1 [119].

The system is in the Normal state when it operates inside the defined stability margins in transmission and generation so that the system can withstand a single contingency. Alert state is characterized by insufficient stability margins and

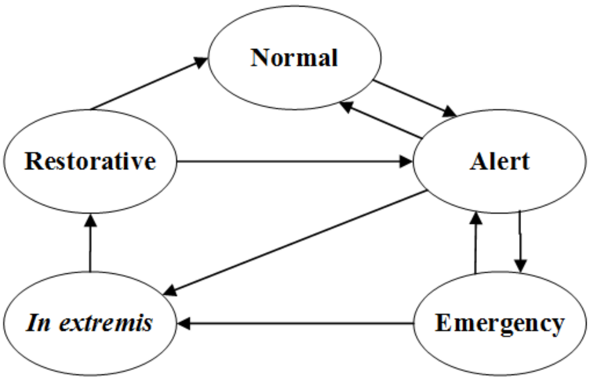


Figure 3.1. Power system operating states and transitions.

higher vulnerability. The system is in the Emergency state if at least one system inequality constraint is violated. When both, equality and inequality constraints are violated, a transition to *In extremis* (that is a non-operational state) occurs. Transitions between the states occur due to component failures and restorations. A system is in the Restorative state during a recovery and maintenance.

3.2 Smart Grid Dependability Modeling and Evaluation

With increased utilization and importance of cyber-physical systems in many complex, safety critical, and cross-disciplinary fields, Smart Grid modeling from CPS'es perspective emerged recently as an important research topic resulting in a number of relevant scientific works.

Modeling of the electric power grid as a cyber-based physical system has been proposed in [120]. The work has introduced a novel cyber-based dynamical model whose mathematical description depends on cyber technologies supporting the physical system. The paper discusses how such a model could be used to enable full observability through a cooperative information exchange among its components. The authors also show how the proposed cyber-physical model could be used to develop interactive protocols between the intelligent electronic devices embedded within the system layers and the network operator. Nevertheless, due to a tremendous amount of information that will be produced and collected, novel modeling methods for efficient integration of advanced monitoring and control instruments for such cyber-physical systems are needed. These methods should be built as compatible and scalable with the existing SCADA systems and they should also support future industry needs.

In [121] the authors take a different approach and do not consider the entire

Smart Grid as a cyber-physical system but, instead, CPS'es, as individual devices, are modeled as mediators between the physical world and the business aspects of electric power grids. Continuous evolution of embedded and ubiquitous computing technologies is perceived as a driver for decentralization of business decisions by transferring them to computing nodes that are closer to customers.

Utilization of cyber-physical energy systems (CPES'es) for key Smart Grid related challenges such as modeling power systems, energy efficiency, energy resource management, and energy control are studied in [122]. The application of CPES'es for optimal power flow management has been described for a specific use case of a microgrid model. In line with this, some customized solutions such as cyber-physical SCADA for the distributed energy resources management are proposed in [123].

Considering Smart Grid as a cyber-physical system, in [51] interdependencies between electrical and information infrastructure are qualitatively analyzed. A model has been developed to capture the effects of failure propagation from one infrastructure to another, distinguishing between cascading, escalating, common-cause and unrelated failures. Each infrastructure is modeled with a set of states that includes working state, weakened state, partial outage state, failure state, and restauration state. A unified state model of the system is generated combining the states of individual infrastructures. Transitions between the states may be triggered by events from one or another infrastructure and are used to model the interdependencies. Failures of information infrastructure are classified as masked and signaled. Errors of the information infrastructure are classified as active and passive. Active errors are those that directly affect the electric infrastructure. Passive errors are those that do not affect the electric infrastructure but make electric infrastructure errors undetectable. Malicious attacks are also briefly addressed.

A generic guideline for developing a unified tool for the analysis of reliability of the electric power system, having in mind the interdependencies between the infrastructures, has been proposed in [82]. In the framework both, static and dynamic aspects of the system are considered. A method for structural modeling of the electric system capable to capture high-level elements, such as topology of the system, as well as low-level ones that are associated with basic components is proposed. The authors differentiate between two types of disruptive events (failures), namely transient or permanent disconnection of a component and transient or permanent overloads. Cyber infrastructure failures are summarized as omission failures, time failures, value failures and byzantine failures. Faults are not addressed. The framework also implements a few basic modeling mechanisms.

In [124], a quantitative modeling and analysis of reliability of Smart Grids has been conducted with focus on transmission network. The method relies on the analysis of previous cascading failure scenarios to group grid components into subsystems and reduce the size of the model. The method has been extended in [40] and [94] to analyze reliability of the grid in the presence of interdependent failures. The focus in [94] is on the application of FACTS (Flexible Alternating Current Transmission System) devices that may increase reliability of the grid by controlling power flow in transmission lines. The authors evaluate the effect that failures of FACTS devices have on the grid. Software failures of FACTS devices are analyzed with fault injection and possible failure propagation scenarios are identified. Faults, errors, and failures are not clearly distinguished and terms are occasionally used interchangeably. Four types of software faults with respect to their effect on the power flow are identified and a generic model of the grid reliability is derived.

A few papers focus specifically on dependability of ADNs. Voltage stability in ADNs is studied in [14]. The main conclusion of the work is that current methods are not sufficient to cope with higher penetration of renewable resources. The author also proposes a few novel algorithms for managing a protection system in the presence of distributed generation for improved voltage stability. These schemes are based on proper timing for triggering different protection means that include shunt capacitors, DERs and online tap changers (OLTCs). The author has demonstrated that, with a proper scheme, DERs may generally contribute to grid's stability.

In [5], the impact of renewable resources, storages, and demand response programs on ADNs, and Smart Grids reliability in general, has been reviewed. The conclusion is that, an ideal mix of Smart Grid resources may lead to its better stability. An architectural blueprint to facilitate design, development and integration of Smart Grid components for ensuring reliability has also been proposed.

With proliferation of ICT elements, power grids are becoming more prone to cyber-attacks and cyber-security of Smart Grids is getting more importance and therefore being increasingly researched. The most frequent topics are intrusion prevention, privacy, and confidentiality. A good overview of Smart Grid cyber-security aspects including identification of security requirements, modeling of network vulnerabilities, attack countermeasures, secure communication protocols and architectures is given in [125]. The problem of propagation of faults caused by attacks from cyber to physical infrastructure is addressed in [51].

3.3 Disturbance Prediction and Proactive Management

In the sense used in this thesis, proactive operation means predicting problems online and taking preventive actions in order to avoid the anticipated problems. According to [8], this concept, which is identified as one of the most important ones for enduring dependability of future power systems, considers two main aspects - disturbance prevention (i.e. proactive disturbance management) and asset management (i.e. predictive maintenance). In this regard, the following areas of proactive grid management solutions are presented in [8]: 1) Decision Support Systems (DSS'es); 2) synchrophasor solutions; 3) symbiotic integration of synchrophasors with fast-acting controls. The proposed solutions rely on the analysis of a combination of PMU and SCADA/EMS collected and processed grid-related information. The usage of historical (i.e. pre-event) data is also foreseen.

The concept of prediction-based control has already found its way to power systems through the application of a model-predictive control (MPC). The main idea of a MPC is to predict changes in dependent variables of a dynamic system by observing changes in independent ones and to optimize the system based on predictions. However, this approach is not being used to predict disturbances online but mainly to predict power imbalance. For example, in [126] a two-stage scheduling has been proposed to optimize control of distributed resources in ADNs. A day-ahead scheduler optimizes renewable production based on a generation forecast and load profiles, and a 15-minutes scheduler adapts the optimization points based on observed changes and a more accurate production and load anticipation that further triggers control of DERs and OLTCs. A similar approach has been taken in [127] to optimize voltage control.

Considerable research efforts have been invested in the improvement of grid's reliability from the perspective of efficient maintenance, coupled with maximization of assets utilization. In particular, this concerns a shift from scheduled to predictive grid maintenance [87]. The predictive maintenance instruments include a group of programs named Reliability-Centered Maintenance (RCM). In an RCM approach, various alternative maintenance policies can be compared to select the most cost-effective one for sustaining equipment reliability. A relevant work on using historical data for predicting failures of aged components and systems followed by a case study on maintaining New York City's electric grid has been presented in [65]. The most important properties of the authors' approach are: that machine learning features are meaningful to the domain experts, that the processing of data is transparent, and that prediction results are accurate enough to support sound decision-making.

In [128], an approach to predict power grid weak points, and specifically to efficiently identify the most probable failure modes assuming static load distribution for a given power network has been developed. The approach is applied to two examples. The algorithm represents a power network adaptation of the heuristic originally developed to study low probability events in physics. One finding is that, if the normal operational mode of the grid is sufficiently healthy, failures are relatively sparse, i.e. failures are caused by load fluctuations at only a few buses.

Power grid disturbances are extensively analyzed from the statistical point. For example, in [129] two methods (a method of fault position and a method of critical distance) are proposed for stochastic prediction of voltage sags in transmission networks. The methods are used to generate a voltage-sag map that visualizes a variation of power quality through the system. For each substation, the expected number of voltage sags per year with a magnitude of less than 85% is calculated. When monitoring data are not available, the method of critical distances is identified as an acceptable alternative for estimating the expected number of sags based on the system topology. However, this and similar methods are not applicable for online prediction of disturbances but are mainly used to identify failure-prone parts of the grid as a part of planning grid upgrades and maintenance or to support failure localization.

With renewable generation and ADNs on one side and advances in machine learning algorithms on the other, online predictions in power grids are becoming more popular as means for optimal grid control. Most of the work is focused on predicting generation from renewable resources or on prediction of demand (load). For example, a heuristic for predicting active AC power generation of a PV has been proposed in [130] for optimal control of microgrids. The heuristic is based on an observed correlation between the time derivative of the active power output and the errors caused by a generic point forecast technique. A comprehensive survey of neural network methods for short-term load forecasting is presented in [66]. However, it is not clear if these approaches may be used for predicting disturbances online.

An exception is the work presented in [63]. The authors propose three methods (based on PMU voltage measurements, phase angle, and the single machine equivalent (SIME) method) for online prediction of transient voltage sags caused by rotor swings. Even though the quality of prediction is not quantified in appropriate metrics, the results are valuable as they clearly indicate that voltage sags of the specific type may be predicted with good lead time. Specifically, the SIME-based methods allows early and consistent identification of critical buses and the prediction of the voltage sag minimum value with low computational load.

3.4 Relevant Smart Grid Projects

With gradual digitalization of the grid, the main focus of most of the research projects is on monitoring, data collection, statistical analysis and visualization whereas advance data analytics and grid control based on these analytics are still not addressed to a sufficient extent and are usually considered as a future work. Probably the best known academic Smart Grid project worldwide is the frequency monitoring network FNET/GridEye project [34, 131], in the scope of which a wide area monitoring and data collection system have been implemented. The system is based on Frequency Disturbance Records (FDRs) that measure frequencies, phase angles and voltages at ordinary customer outlets and provide GPS synchronization. The main advantage of FDRs is the user-friendly installation as devices may be simply plugged into ordinary household outlets. Still, their accuracy and the sampling rate are below of those of PMUs. Moreover, FDRs measure power quality indices only on one phase. In the scope of the project, FDRs are mainly distributed over the US but a considerable number of devices is installed in Europe as well. The system also implements an online visualization and allows detection of such incidents as oscillations, frequency and angle perturbations, system failures and islanding. Prediction of instabilities is also foreseen as a part of future enhancements. Due to the currently limited number of FDRs, only major events may be analyzed and there is still not sufficient data to perform a comprehensive analysis at the level of a single distribution system.

Major industrial players such as Alstom and PG&E (Pacific Gas & Electric) announced to advance Synchrophasor Grid Monitoring into Proactive Grid Stability Management [132]. In 2013, Alstom Grid collaborated with PG&E project team to deliver enhanced e-terra integrated real-time synchrophasor and EMS applications as the first stage of the Production Grade Synchrophasor Project. This enables PG&E to monitor power system behavior from a new class of GPS time-synchronized, high resolution PMUs. These devices take grid measurements with the rate of up to 120 times per second versus the traditional rate of one measurement per four to six seconds with unsynchronized SCADA sensors. The increased observability will allow PG&E to identify and to analyze system vulnerabilities in real-time, assess available transfer margins across transmission corridors and provide corrective actions to prevent potential blackouts. In the future, Alstom's Grid Stability Package will help to integrate existing measurement-based PMU analytics, model-based EMS and dynamic stability analytics to enable proactive management of grid stability.

Tollgade, in partnership with DTE Energy, implemented a pilot to prove the concept of "predictive grid" across DTE Energy's service territory by installing advanced monitoring equipment and predictive grid analytics software to find tell-tale signs of faults and asset health symptoms before outages occur [133]. The preliminary results have shown that up to 86% of all outages might have been preceded by line disturbances which is a strong motivation to apply a similar approach to predict not only outages but also other types of disturbances such as voltage sags.

In Switzerland, the Swiss Competence Center for Energy Research (SCCER) has been established in 2014 with the aim of ensuring seamless transition from centralized (that is to high extent nuclear-based) power generation to decentralized generation based on renewable resources and to meet Swiss Energy Policy 2050. It is the major Swiss Smart Grid project, that brings together a number of academic and industrial partners and is the umbrella for eight projects in the SCCER family that address specific aspects of the future Swiss grid from energy efficient buildings and industrial processes, to energy supply and future distribution and transmission infrastructure. For example, SCCER-FURIES[1] tackles the problem of the future Swiss electrical infrastructure with focus on topics such as: future grid monitoring and control infrastructure, control of a large number of DGs and distributed storages, demand side management, and standardization.

As a part of SCCER-FURIES, a grid operator Arbon Energie and Siemens (as a grid monitoring software provider) joined their forces in Arbon Smart Grid project[2] to establish a novel Smart Grid platform for an improved utilization of renewables coupled with DSM strategies while taking care of high standards of power quality. Development of tools for visualization of monitored data and provision of ancillary services based on advanced grid control are foreseen as a part of the project extension.

Swiss Energypark has been created to support testing and validation of grid management strategies for ADNs based on advanced monitoring and control instruments. A real-time monitoring infrastructure has been implemented for an ADN at the EPFL campus [27]. A reliable monitoring system comprises 6 PMUs, a dedicated communication infrastructure and novel algorithms for state estimation. All data records are available online for research purposes. Records are mainly used for testing and improving PMU-based state estimation algorithms. Since the campus grid is designed as highly reliable and is also well controlled, disturbances are very rare events and records do not include sufficient disturbance-related data that could be used to design algorithms for their

[1] Future Swiss Electrical Infrastructure (SCCER FURIES), http://sccer-furies.epfl.ch/

[2] ArbonEnergie, http://www.arbonenergie.ch/

prediction.

GridBox[3] is an on-going industrial project that aims at providing more information on the current state of distribution grids at medium- and low-voltage levels as well as solutions for gird monitoring and control. The core of the proposed solution is a highly distributed network of GPS-synchronized devices for real-time measurements. Foreseen applications include grid monitoring and control, real-time state estimation, real-time optimization, as well as automated topology detection. Two operational pilots will be implemented as final deliverables.

VEiN[4] pilot project has been realized by AEW Energie AG in Kreuzmatt in Rheinfelden with the aim of determining the effect of distributed generation on the electricity quality and the operation of low-voltage networks. The distribution system incorporates two combined heat and power units and four photovoltaics (PVs). The grid quality is obtained from online measurements of voltages and currents using PQ (power quality) devices that are similar to PMUs. Initial evaluation of recorded data indicates several voltage sags and swells mostly near PV generation.

[3] GridBox - A holistic Smart Grid approach, http://www.gridbox.ch

[4] VEiN - Verteilte Einspeisung in Niederspannungsnetze, http://www.vein-grid.ch/projekt.html

Chapter 4

Proactive Management and Its Impact on Availability

Proactive management may be adopted in different industries and for various purposes from improving overall efficiency and performance of a system to decreasing operational cost. For example, proactive approach may be used to predict aging of Uninterrupted Power Supplies (UPS'es) so as to decrease maintenance cost and to endure reliability at the same time [134]. Our main goal is to use proactive management to enhance dependability and, in particular, availability of ADNs by predicting events such as disturbances and failures so that proactive actions may be taken to mitigate them. In that sense, proactive management may be seen as proactive (predictive) fault-tolerance. As for that, without a loss of generality, when talking about proactive management in the scope of this chapter, terms *predictive management* and *predictive fault tolerance* are also used, referring to the two predictive FT policies that have been defined in Section 2.1 (prediction and avoidance and prediction and preparation). In this chapter, we take a generic approach and propose a model of proactive management that may be adopted for any type of system in order to evaluate availability gain.

Intuitively, if a failure or a disturbance is predicted, and proactive actions are performed successfully, availability will be improved in comparison to a reactive approach based on failure (disturbance) detection and system repair. On the other hand, a proactive action may also introduce a downtime that we refer to as proactive action overhead. Typically, the overhead is much less than the downtime caused by a failure and the system recovery. Thus, assuming that a failure is predicted correctly and mitigated successfully, downtime with predictive policy will be shorter than the reactive policy downtime and availability will improve. We refer to this decrease of downtime as a reward. Nevertheless, as there is no perfect oracle, not every failure can be predicted. Also, there is no guarantee of a successful mitigation. For example, if a system is already in a contaminated state when a failure is predicted, the failure might be inevitable. Furthermore, a failure might be (incorrectly) predicted even when it is not imminent. Such a false alarm will still trigger a proactive action that will introduce unnecessary overhead that we call penalty. For example, in power systems, if a voltage drop is predicted, a shunt capacitor, a solid-state transformer or a DER may be used to boost the voltage just before the expected drop. This will keep the voltage within defined limits in the cause of a correct prediction. But, if the prediction was incorrect, the proactive action will case overvoltage at a specific point that may propagate through the network and cause additional problems.

In conclusion, on one hand, predictive management may enhance availability but, on the other hand, unsuccessful mitigation and low quality of failure prediction may even result in decrease of availability. For example, it has been observed in [68] that a predictive policy for a high performance computing system may increase a task execution time by up to 10% when failure prediction quality is low. Despite a number of implementations of predictive fault tolerance, mostly in computer systems, there is still a need to further investigate a tradeoff between reactive and proactive methods and to provide an analytical model for availability with proactive management. In particular, it is not clear to what extent availability may be enhanced with a proactive FT, how this effect can be modeled, what the minimum requirements for the prediction quality are (in terms of precision and recall), and how failure prediction can be optimized so that availability enhancement is maximized.

To address these problems and to confirm if proactive approach may indeed be used for management of disturbances and failures in Smart Grids and other systems, we create a generic analytical model of predictive FT that may be used to quantify availability gain for various types of systems. This helps designers to quickly estimate availability with predictive and reactive policy and to select the optimal one for the improving availability of a specific system with available failure prediction mechanisms.

To model availability in a comprehensive way, we create a unified Markov chain that incorporates both types of predictive FT policies. Using the model we analyze the effect of a predictive FT on availability and extend the steady-state availability equation so that it includes the model parameters. As, in some cases, it may be possible to change the quality of failure prediction by manipulating prediction parameters, we also derive an A-measure to optimize failure prediction for maximizing availability. We provide guidelines for availability equation and the A-measure application. To evaluate the approach we analyze availability improvement of a virtualized server with predictive FT and conduct sensitivity analysis of the equation and the Markov model to identify the variables that affect availability the most. We then perform Matlab simulations to validate the model accuracy.

In the analysis we focus on computer systems due to higher availability of component failure rates and failure prediction quality data but the generated model and the guidelines are universal and may be used for other types of systems such as Smart Grid and its parts.

A part of the work presented in this section has been published in [135].

4.1 Model of Proactive Management

We first derive a model of failure (disturbance) prediction, identify prediction quality metrics and derive a model of proactive (mitigative) actions. We then generate Markov models for the two types of predictive (proactive) management policies. Starting from these models we derive a generic Markov model of a predictive management that we use to extend steady-state availability equation.

4.1.1 Failure Prediction Model and Quality Metrics

The goal of failure (disturbance) prediction is to predict, with sufficient lead time, whether a failure will occur in a certain time period that is referred to as a prediction window. A predictor should predict as many failures as possible while keeping the number of incorrect predictions to the minimum. The output of a failure prediction can be categorical (Boolean) if it forecasts whether a failure will occur or not, or numerical if a probability of a failure imminence is estimated. By setting a prediction threshold, so that a failure is predicted when the probability is above the threshold, a probabilistic output can be translated into categorical. Depending on the result of failure prediction and its actual occurrence in the prediction window, a prediction may be true-positive (when a failure is predicted and it also occurs), false-positive (when a failure is predicted but does not occur), true-negative (when no failure is predicted and none occurs) or false-negative (when no failure is predicted but it occurs), as summarized in Figure 4.1.

Different metrics may be used to evaluate a failure predictor but, as failures are relatively rare events, precision and recall are identified as the most appropriate ones [72]. If the total number of true-positive, false-positive, true-negative, and false-negative predictions is denoted by n_{tp}, n_{fp}, n_{tn}, and n_{fn} respectively, the

Table 4.1. Failure prediction contingency table.

		Observation	
		Failure	No failure
Prediction	Failure	True positive	False positive
	No failure	False negative	True negative

total number of failures is n_f, and the total number of alarms n_a, then precision (P) and recall (R) may be defined as in Equations 4.1 and 4.2.

$$Precision = \frac{n_{tp}}{n_a} = \frac{n_{tp}}{n_{tp} + n_{fp}} \tag{4.1}$$

$$Recall = \frac{n_{tp}}{n_f} = \frac{n_{tp}}{n_{tp} + n_{fn}} \tag{4.2}$$

If n_{tp} equals zero, both precision and recall are zero. If the number of alarms is zero, precision is set to one by convention. If the number of failures is zero, recall is undefined. This last case will not be considered as a hypothetical system that never fails is fault-free and needs no fault tolerance.

In an ideal case, both precision and recall are equal to 1. In practice, they are related and one measure may frequently be improved at the expense of the other. In fact, for different predictor configurations, different precision-recall pairs can be obtained but, as also observed in [136], prediction with high precision usually has low recall, and vice versa. The relation may be depicted in a precision-recall curve as the one in Figure 4.1. When the output of a prediction is numerical, the relation between precision and recall can be tuned by changing the prediction threshold. Typically, higher threshold causes higher precision and lower threshold causes higher recall. In Figure 4.1, Threshold 1 is greater than Threshold 2. In classification and prediction algorithms, finding an optimal precision-recall trade-off is frequently done with F-measure, which is a harmonic mean of precision and recall. As it will be demonstrated later, F-measure might not be the most appropriate one when the objective is availability maximization. In fact, as also pointed out by [137], abstract metrics such as F-measure are good for benchmarking and comparing prediction (machine-learning) algorithms but they unfortunately ignore problem-specific details. As such, these metrics say nothing about a practical impact of the algorithm and how efficiently the problem in hand is solved. Thus, there is a need for metrics that also incorporate problem-specific attributes such as the cost of an incorrect prediction.

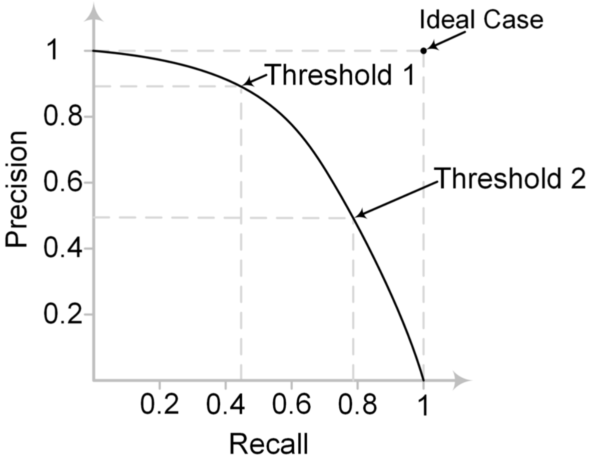


Figure 4.1. An example of a precision-recall curve.

If a predictor is running on the system for which it is predicting failures, then it may also introduce additional load that depends on the computational complexity of the prediction algorithm. For example, we may expect a load increase of about few percents with a relatively simple algorithm as the one from [67] or a significant increase when a more complex prediction (e.g. the one proposed in [138]) is used. To capture failure rate increase caused by higher load we introduce failure rate increase factor as a parameter of failure prediction.

4.1.2 Proactive Action Model

Proactive actions are taken either to avoid failures or to decrease the failure recovery time by preparing for it in advance. A proactive action may be modeled with action latency, action overhead, and success probability. The latency is the total time needed to perform an action and the overhead is the time during which the execution of the main process (and more generally a delivery of a service to a user) may be interrupted. In other words, the overhead may cause a downtime. Note that prediction lead time should not be shorter than the action latency.

The case of live migration of virtual machines (VMs) studied in [139] may serve as an example of a proactive action in computer systems. Two stages of live VM migration are identified: preparatory and blackout. In the preparatory stage, parts of VM are migrated while keeping the VM running on the original node, and in the blackout stage, dirty pages are copied from the source to the target node while interrupting VM execution. In this case, latency is the sum of the preparatory and the blackout stage time, and overhead (downtime) corresponds to the blackout stage. Success probability describes a probability that a failure is avoided or that the system is prepared for the recovery with a proactive action.

In power systems, for example when a DER is used to increase a voltage at the connection point in an anticipation of a voltage sag, the latency is the time needed to start a DER whereas, for this particular case, the overhead is zero.

4.1.3 Models of Predictive Policies

A continuous-time Markov chain (CTMC) model of a prediction and avoidance fault-tolerance policy is presented in Figure 4.2a. Parameters λ, μ, P, R, a, and c stand for: failure rate, repair rate, precision, recall, failure rate increase factor, and proactive action success probability, respectively. Failure and repair rates are those of the system with only reactive policy and equal 1/MTTF and 1/MTTR, respectively. Parameter α is the alarm rate and can be expressed as (R/P)(1+a)λ, whereas γ is a proactive action completion rate that equals 1/overhead.

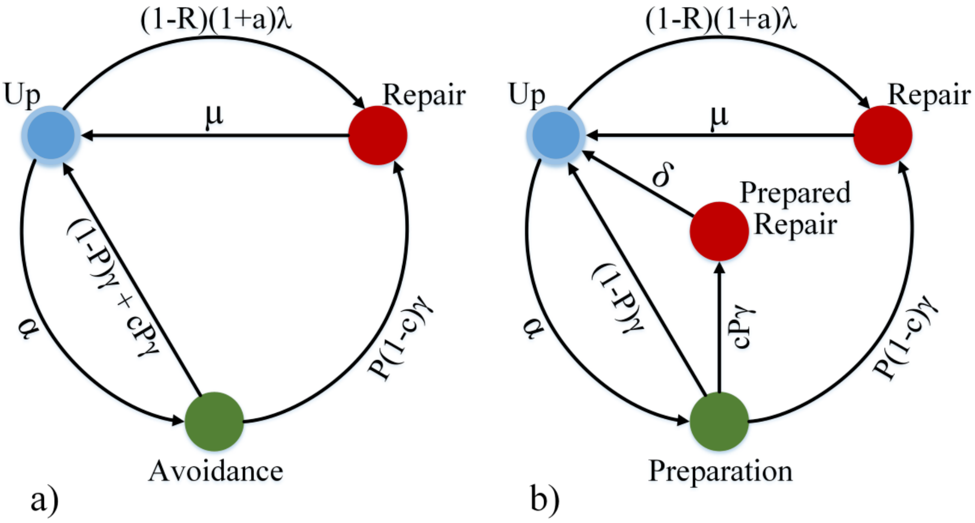


Figure 4.2. Markov models of two predictive fault-tolerance policies (a) prediction and avoidance and (b) prediction and preparation.

Unpredicted failures bring the system to the Repair state at the rate (1-R)(1+a)λ. Alarms (true-positive and false-positive predictions) bring the system to the Avoidance state at the rate α. A transition from the Avoidance state to the Up state occurs at the rate (1-P)γ + cPγ. This corresponds to the cases of false-positive predictions and successful failure avoidances when a prediction is true-positive. When a failure is correctly predicted but unsuccessfully avoided, a transition from the Avoidance to the Repair state occurs at the rate P(1-c)γ. We neglect transitions from the Avoidance state to the Repair state for the cases when another failure occurs during the proactive action overhead time assuming γ»λ.

A CTMC model of a prediction and preparation fault-tolerance policy is presented in Figure 4.2b. The difference with respect to the model from Figure 4.2a, apart from the naming of the states, is that correct prediction and successful preparation brings the system to the Prepared Repair state at the rate c*P*γ. As the system is prepared for the recovery, mean time spent in the Prepared Repair state (MTTRp) is shorter than MTTR. The rate δ, that is a transition rate from the Prepared Repair state to the Up state, equals 1/MTTRp.

Solving the models from Figure 4.2 for the system steady state, one can derive two different steady-state availability equations for the two types of predictive FT. Deriving a generic availability equation requires introducing two additional parameters, penalty and reward. Penalty corresponds to introduced downtime when the action was needless or unsuccessful and reward is a downtime decrease in the case of a correct prediction and successful proactive action. To illustrate the impact of these parameters more precisely, we compare availability and unavailability time periods of systems with reactive and the two types of predictive fault tolerance policies considering the four cases of a prediction.

System availability for the case of a true-positive prediction assuming a successful proactive action is depicted in Figure 4.3. Lightning symbol indicates a failure occurrence time and an alarm (a bell symbol) indicates a failure prediction time. For simplicity, we assume that a proactive action is initiated as soon as a failure is predicted as the time needed to initiate the action may be considered as a part of the action latency. The system is unavailable when under a recovery or during a period that corresponds to the proactive action overhead. We define reward, for the case of a prediction and avoidance policy as MTTR - overhead, and for the case of a prediction and preparation policy as MTTR - (overhead + MTTR_p).

In the case of a false-positive alarm, preventive actions are still initiated when an alarm is raised even though the action is needless. Thus, for both types of predictive policies, system states will be identical to the ones depicted in Figure 4.3b. Penalty is equal to the overhead in this case. If a failure is not predicted, but still occurs (false-negative prediction), the same scenario as in the case of reactive policy applies. When prediction is true-positive but a proactive action is unsuccessful, the system unavailability period equals the sum of the penalty (that equals to the overhead) and MTTR.

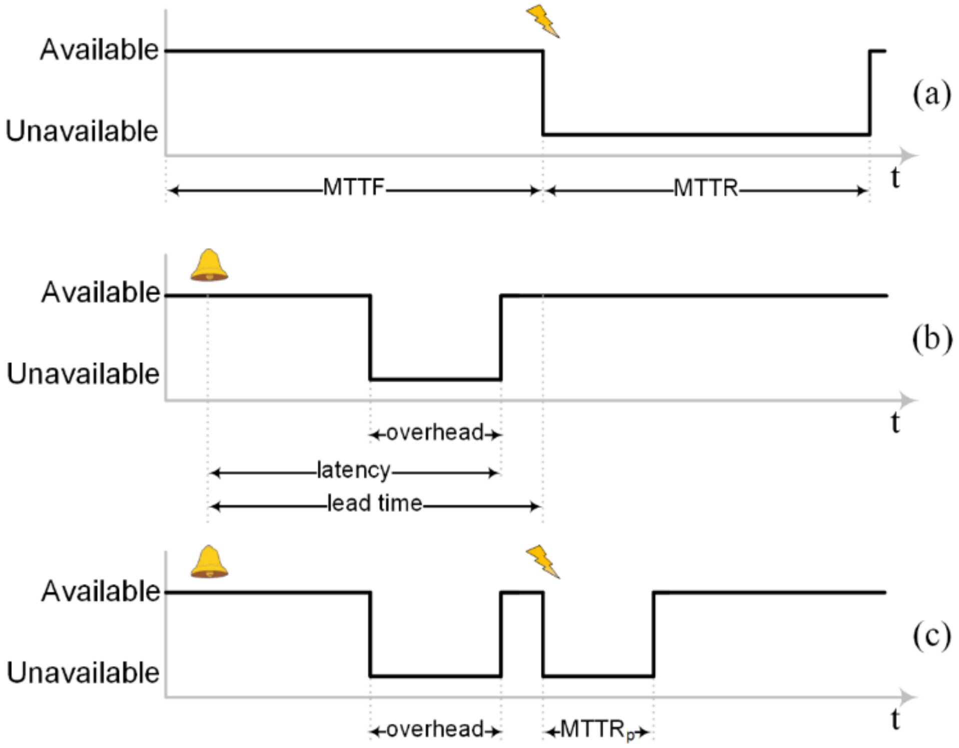


Figure 4.3. System availability for the case of a true-positive alarm and successful proactive action for (a) reactive, (b) prediction and avoidance, and (c) prediction and preparation fault-tolerance policies.

We include these parameters and derive a unified Markov model of proactive fault-tolerance policies that we also compare to the models from Figure 4.2 to ensure equivalence. Parameters γ and δ from Figure 4.2 may be expressed in function of penalty, reward and MTTR. As before, we assume transition rates from any other state to the Up state are much higher than 1/MTTF so that additional failures do not occur while the system is unavailable. The model is depicted in Figure 4.4 with states described in Table 4.2.

Thus, predictive FT policies may be generically modeled with prediction 1) precision (P) and 2) recall (R) of failure prediction, 3) penalty (p) and 4) reward (r), 5) proactive action success probability (c) and 6) failure rate increase (a). In addition, the availability model has to include MTTF and MTTR of the same system when only a reactive policy is implemented.

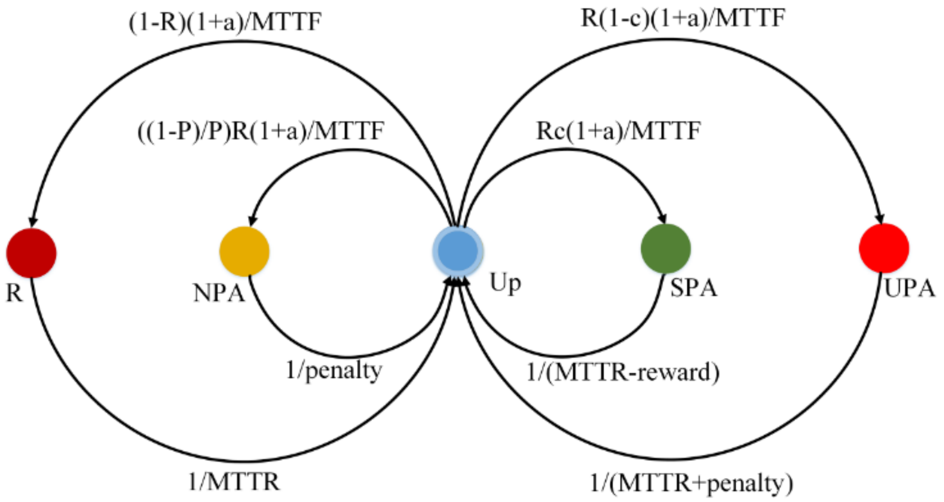


Figure 4.4. Unified Markov model of predictive fault-tolerance policies.

Table 4.2. Description of the unified Markov model states.

State ID	**Description**
R	Recovery state.
NPA	State caused by needless proactive actions due to a false-positive alarm.
SPA	State caused by a true-positive alarm and a successful proactive action.
UPA	State caused by a true-positive alarm and an unsuccessful proactive action.

4.2 Optimizing Prediction for Enhanced Availability

Steady-state availability, that is simply a ratio of system's uptime over the lifetime is the most frequently used to quantify availability of a system that implements a reactive FT policy. We fist extend this equation for the case of a predictive policy and then derive a measure and provide a guideline for maximizing availability by tuning failure prediction quality.

4.2.1 Steady-state Availability Equation

By analytically solving the generic model for the steady state, a steady-state availability equation for the case of a predictive FT policy is derived in Equation 4.3. Non-capital 'r' stands for reward, whereas 'p' stands for penalty. Other symbols are used as before. Availability of the same system with a reactive policy is simply the ratio between MTTF and (MTTF+MTTR). A less comprehensive version of Equation 4.3 can be derived without a Markov model by only considering cases from Figure 4.3 as we did in [135]. Equation 4.3 can be presented in a more compact and well-known form as in 4.4 with MTTFp defined by Equation 4.5 and MTTRp defined by Equation 4.6.

$$A_p = \frac{\frac{MTTF}{1+a}}{\frac{MTTF}{1+a} + MTTR - R * \Big(c * r - \big((1-c) + \frac{1-P}{P}\big) * p\Big)} \tag{4.3}$$

$$A_p = \frac{MTTF_p}{MTTF_p + MTTR_p} \tag{4.4}$$

$$MTTF_p = \frac{MTTF}{1+a} \tag{4.5}$$

$$MTTR_p = MTTR - R * \Big(c * r - \big((1-c) + \frac{1-P}{P}\big) * p\Big) \tag{4.6}$$

The equations apply when precision is non-zero. The precision is zero only when the number of true-positive predictions is zero, in which case recall is zero as well. With such a prediction no failures can be anticipated, and predictive FT that relies on it makes no sense. If recall tends to zero due to a high number of failures and a small number of true-positive predictions, availability improvement will be negligible. If precision tends to zero, the expression in parenthesis in Equation 4.3 becomes negative and availability decreases.

In a simplified case, when the failure rate increase factor and the success rate can be neglected (a=0 and c=1), the break-even point can easily be derived. It defines the minimum requirement for the failure prediction quality so that availability improves with respect to the system with a reactive policy, and is defined by Equation 4.7. It is interesting to notice that the break-even depends only on precision but not on recall. However, the scale at which availability improves depends on recall as well.

In a realistic case, when a>=0 and/or c<=1, the inequality 4.7 still has to hold but, in this case, it is only a necessary and not a sufficient condition for a predictive policy to improve availability over a reactive one.

$$P \geqslant \frac{penalty}{penalty + reward} \tag{4.7}$$

4.2.2 A-measure and Prediction Optimization

In practice, finding a trade-off between precision and recall is frequently done with F-measure [140] (also called F-score), which is a harmonic mean of precision and recall. However, this is appropriate only for the case of classification algorithms when the number of instances of two classes is balanced. This is not the case with failure prediction as there are typically significantly more non-failure than the failure instances. In fact, the optimization function should be driven by the application requirement that, in our case, is availability maximization. For the optimization of prediction to maximize availability, we introduce A-measure that is defined by Equation 4.8. The measure actually represents a relative MTTR decrease with respect to a reactive policy. The trade-off between precision and recall should be such that A-measure is maximized in order to maximize availability assuming the success factor is defined and that a prediction precision-recall curve is known.

$$A_{measure} = R * \Big(reward * c - \big((1-c) + \frac{1-P}{P}\big) * penalty\Big) \tag{4.8}$$

In order to apply the equations and to find an optimal precision-recall pair, model parameters and the precision-recall curve have to be estimated. In a more favorable case, the system is already operational with a reactive policy, and error logs and system parameters that are needed for the specific failure predictor are recorded and stored. In the other case, it must be possible either to run experiments on the same or a similar system in order to obtain the data or to simulate the system. By running a prediction on the recorded data set and changing prediction parameters (e.g. prediction threshold in the case of a numerical predictor), different precision-recall pairs can be obtained and a precision-recall curve derived. MTTF and MTTR can be obtained from error logs by considering time stamps. Penalty, that is essentially the action overhead, can be derived by running a proactive action, for different system configurations, and finding an average system unavailability period during the action execution.

Estimating reward, the rate increase factor, and the success probability, requires to run experiments on the system with failure prediction. The rate increase factor can be simply derived by comparing MTTF of the system without failure prediction and with failure prediction but without taking any proactive actions. Alternatively, for computer systems, when the load increase is known, a model from [141] may be used to estimate a failure rate increase. To estimate success probability and reward more accurately, a large number of true positives is required. To accelerate their estimation, failure prediction should be set such that recall is high. This will increase a probability of both true-positive and false-positive predictions.

We will first consider the case of a prediction and avoidance fault-tolerance policy. Let the total number of alarms during the experiment, which corresponds to the sum of true-positive and false-positive predictions, be n_a. Furthermore, let the total number of observed cases when a failure is predicted and it also occurs, that corresponds to a correct prediction and unsuccessful avoidance, be n_{pf}. If n_{tp} is the number of true-positive predictions during the experiment, then $n_{pf}=(1\text{-}c)n_{tp}$. As precision and recall are predefined for this set of experiments, n_{tp} can be expressed as a product of precision and n_a and the proactive action success probability derived as $c=1\text{-}(n_{pf}/(P{*}n_a))$. Reward is simply the difference between the MTTR without failure prediction and the penalty.

For the case of prediction and preparation policy, success probability can be derived in a similar manner with a difference that the case of a correct prediction and unsuccessful proactive action is recognized when the downtime associated with system repair equals MTTR. The cases when a failure is predicted and the repair time is shortened correspond to a correct prediction and successful preparation. If an average repair time for these cases is MTTRp, then reward is MTTR-(penalty+MTTRp).

Alternatively, when experiments and simulations cannot be performed, one may still analyze system's availability with or without predictive fault tolerance and identify optimal precision and recall. This can be done by estimating model parameters while considering similar systems described in the literature and different phases of recovery and proactive actions that contribute to MTTF, MTTR, penalty and reward.

Once the parameters are estimated, the procedure for finding an optimal trade-off, between precision and recall with respect to A-measure, is rather straightforward and similar to the one used to finding the maximum F-measure. In this case, for each point in the precision-recall curve, A-measure has to be calculated to find the maximum. When precision-recall curve can be approximated with a mathematical function, an optimal point may be found by expressing precision as a function of recall (or vice versa), inserting this relation into Equation 4.8, and finding the maximum of the A-measure using a derivative.

The availability equation can be applied to the entire system or only to one or more components depending on the type of failures that can be predicted. In the first case, Equation 4.3 may be applied directly. In the second case, availability model of the system has to be first derived (e.g. a Markov chain or a Reliability Block Diagram) and equivalent MTTF and MTTR of a component(s) calculated following Equations 4.5 and 4.6. A similar procedure applies when only one type of system or component failures can be treated proactively. For example, if only software failures of a server can be predicted, then a Markov model of the system with a reactive policy can be generated so that server software failures are modeled with a separate system state with appropriate transitions rates from the "Up" state to the "Down" state that correspond to $1/MTTF_{ssf}$ and $1/MTTR_{ssf}$ ("ssf" stands for "server software failure"). Then, to evaluate availability of the system with a predictive policy, $MTTF_{ssf}$ and $MTTR_{ssf}$ have to be substituted with appropriate measures following Equations 4.5 and 4.6 and the entire Markov model has to be reevaluated.

4.3 Model Validation

To validate the model and to demonstrate the application of the derived generic Markov model and the equations, as well as to analyze sensitivity of system's availability with respect to different model parameters for the two types of predictive fault-tolerance policies, we consider a simple virtualized server system.

4.3.1 System Structure and Parameters

As, in this type of computer systems, host (hardware) failures are identified to be among the most frequent and severe ones [142, 143] with host's MTTF and MTTR having a large effect on system's availability [67], we analyze availability of the server infrastructure to understand to what extent it may be improved with predictive FT and under what conditions. The system is composed of the main server, a cold spare and a shared external storage. It also supports live virtual machine migration. The structure of the system is depicted in Figure 4.5.

Initially, the system uses a reactive failover policy with checkpoint-recovery mechanism. A checkpoint is created periodically on the external storage that may be accessed from both hosts. A checkpoint saves an entire VM state. Another VM is started on to the spare host when a failure of the main host is detected, and the computation continues from the latest checkpoint. Once the failed host is repaired it takes the role of a spare. For the simplification we assume that MTTF » MTTR and that no host failures occur while the other one is being repaired. This

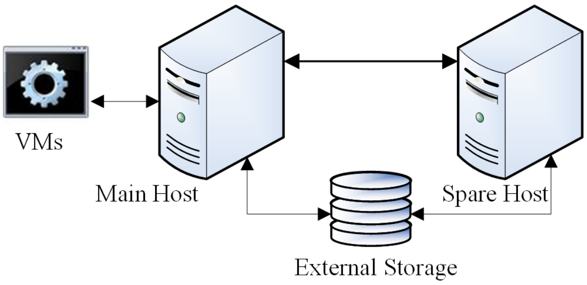


Figure 4.5. Structure of the analyzed virtualized server system.

is also a frequent practice in analysis of similar systems. In the prediction and avoidance policy a live migration of VM is initiated when a failure of the main host is predicted. In the prediction and preparation policy, a new checkpoint is created as soon as the host failure is predicted and offline migration of VM from the failed host to the spare one is conducted after a failure is detected. We assume that the lead time equals 5 minutes as in [67] that considers a commercial telecommunication server system. Also, we assume that a similar prediction method as in [67], which has low computational over-head and good prediction quality is implemented. A sufficient lead time improves proactive action success probability and low overhead ensures a small increase in failure rate. On the other hand, increased lead time may cause a decrease in the quality of prediction [67, 144].

To create as realistic and as generic model of a server system as possible, host failure parameters and the success probability are adopted from [145] and adapted for the selected system, time-aspect parameters of live VM migration from [80, 139, 146, 147] , and checkpointing parameters from [148, 149]. Other parameters estimation is based on empirical knowledge. A generic Markov model from Figure 4.4 is analyzed with the Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) [80] and results are compared against Equation 4.3 to ensure equivalence. Table 4.3 summarizes system parameters.

The first part of the table contains the data obtained from the literature and the calculated model parameters for the case of the reactive policy. MTTR (for the reactive policy) is the sum of mean values of failure detection time (MTTFD), migration latency (MML) and system recovery time with periodic checkpointing ($\mathrm{MRT_{pcp}}$). Failure rate increase factor is set to a rather high value (10%) in order to include even a case when failure prediction introduces great load to the system. In practice, we may expect failure rate increase of only a few percents. Predictive policies parameters are presented in the second part of the table and are calculated following the equations given in the second column.

Table 4.3. System parameters.

	Parameter	**Definition**	**Value**
Adopted from the literature	MTTF	Mean-Time-To-Failure	1000 h
	MTTFD	Mean-Time-To-Failure-Detection	30 s
	MML	Mean-Migration-Latency	20 s
	MMO	Mean-Migration-Overhead	5 s
	MRT_{pcp}	Mean-Recovery-Time (periodic checkpoint)	8 min
	MRT_{odcp}	Mean-Recovery-Time (on-demand checkpoint)	4 min
	MCPO	Mean-Checkpoint-Overhead	10 s
	MTTR	= MTTFD + MML+ MRT_{pcp}	530 s
	MTTR_{pp}	= MTTFD + MML + MTR_{odcp}	290 s
	c	Proactive action success probability	0.90
	a	Failure rate increase factor	0.10
Derived	$\text{penalty}_{\text{pa}}$	= MMO	5 s
	$\text{reward}_{\text{pa}}$	= MTTR - MMO	525 s
	$\text{P}_{\text{breakeven-pa}}$	Approx. break-even point (See 4.7)	0.0095
	$\text{penalty}_{\text{pp}}$	= MCPO	10 s
	$\text{reward}_{\text{pp}}$	= MTTR - (MTTR_{pp} + MCPO)	230 s
	$\text{P}_{\text{breakeven-pp}}$	Approx. break-even point (4.7)	0.0416

For the prediction and avoidance predictive policy (subscript "pa" is used to indicate policy parameters), assuming that prediction lead time is greater than the migration latency, penalty equals the migration overhead (that corresponds to the VM live migration blackout stage), and reward is the difference between the MTTR and the migration overhead. For the prediction and preparation policy (subscript "pp" used to indicate parameters), with the same assumption for the lead time with respect to checkpoint latency, penalty equals checkpoint overhead (MCPO). As checkpoint is created closer to the failure, recovery time is decreased to MRT_{odcp}, and mean-time-to-repair becomes MTTR_{pp}. It is interesting to observe that the minimum precision for which a proactive policy is superior over the reactive (the breakeven point) is very low for the case of both predictive policies. The reason is that the rewards are much higher than the penalties.

4.3.2 Availability with Reactive and Proactive Policies

Steady-state availability for the range of values of precision and recall between 0.001 and 1 is depicted for the prediction and avoidance, and for the prediction and preparation policy in Figure 4.6 and Figure 4.7 respectively. A solid line on the top surface in the figures applies to the precision-recall curve from Figure 4.1. For the comparison, availability of the system with a reactive policy is also presented. A breakeven precision value (see Equation 4.7) is where the two surfaces in figures intersect.

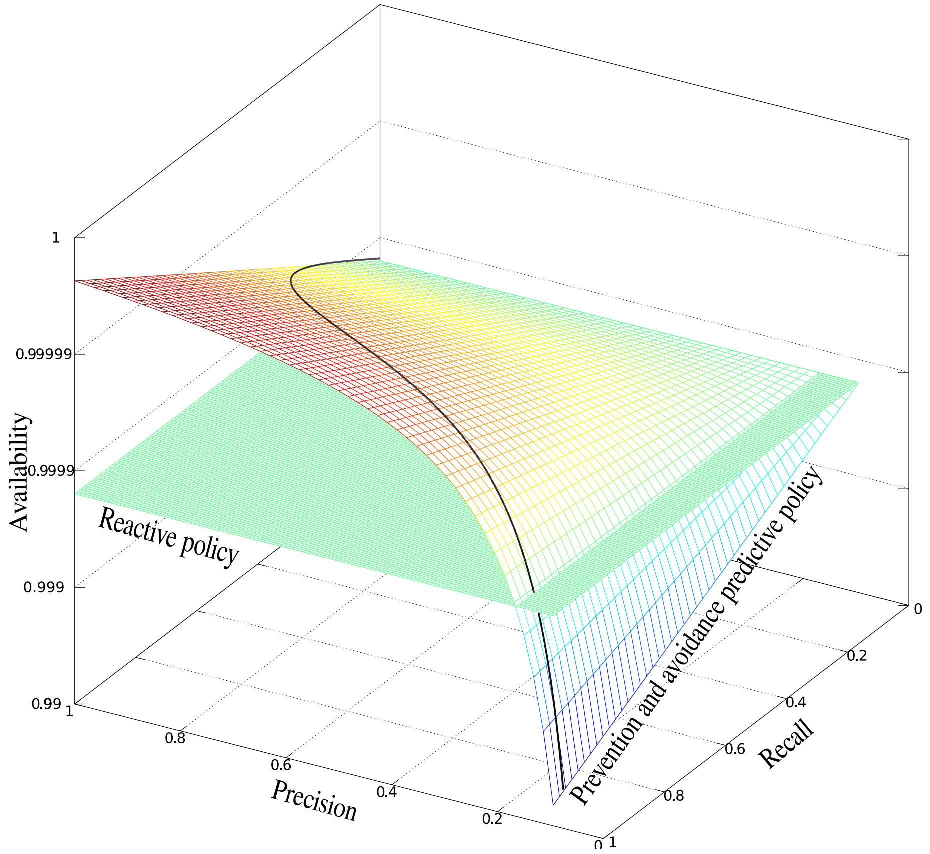


Figure 4.6. Availability of the server infrastructure with prediction and avoidance, and reactive policies as a function of prediction quality.

Thus, when the prediction quality is such that precision is above the breakeven point - proactive policy is superior over the reactive one, and vice versa. Note that two predictive policies have different penalty and reward values, so that the range of availability that can be achieved varies differently with changing precision and recall. It is interesting to observe that the availability of a proactive

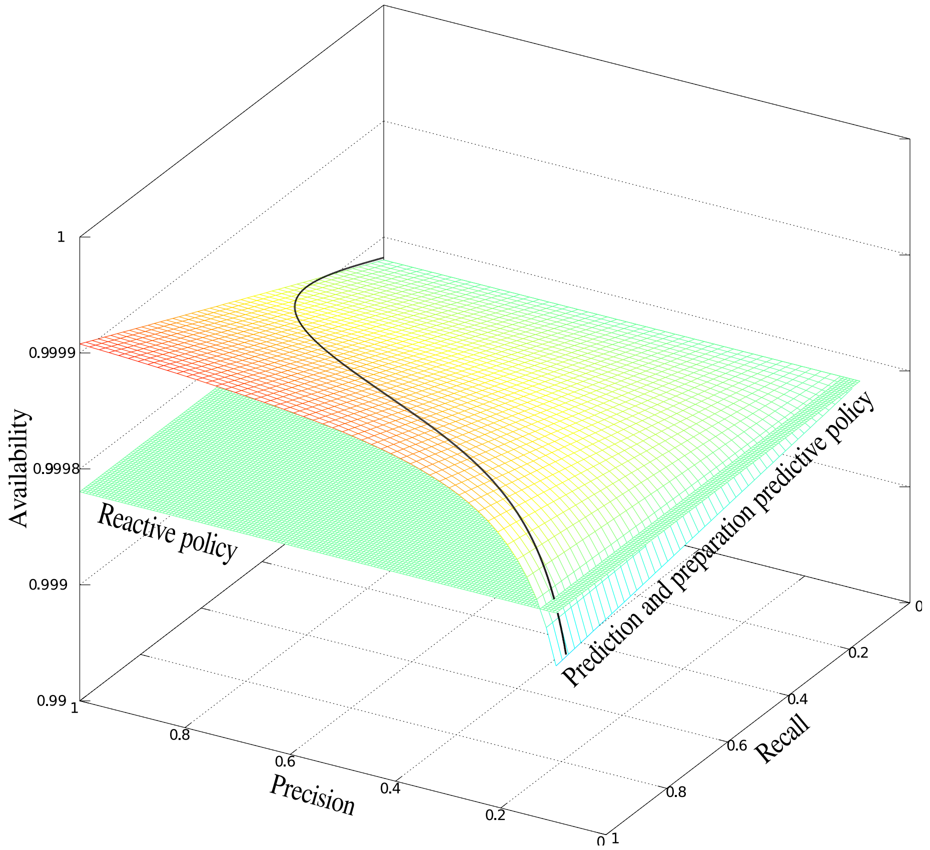


Figure 4.7. Availability of the server infrastructure with prediction and preparation, and reactive policies as a function of prediction quality.

system increases/decreases with the increase of recall, depending on whether precision is above/below breakeven line or not. For the selected parameters and a perfect failure prediction (P = R = 1), we compare steady-state availability, steady-state unavailability, and downtime per year for the server infrastructure with reactive and the two type of predictive policies. Results are presented in Table 4.4.

With the prediction and avoidance policy, unavailability reduces by a factor of almost 74 times, and with the prediction and preparation policy, unavailability is almost halved. Thus, even with highly reliable components, such as a server with MTTF of 1000 h, availability of the server system may still be additionally improved with a predictive FT policy if prediction quality is high.

The improvement effect is even more evident when reliability of the server is lower. In practice, a typical MTTF of a server is about 20 days that is, approximately, 500 h. In this case, availability with a reactive policy is 0.999705. With

Table 4.4. Availability of different policies with perfect failure prediction.

Policy type	Availability	Unavailability	Downtime per year
Reactive	0.999853	$147*10^{-6}$	1h 17 min
Prediction and avoidance	0.999998	$2*10^{-6}$	1 min
Prediction and preparation	0.999915	$85*10^{-6}$	44 min

predictive policies, availability reaches 0.999991 for the prediction and avoidance policy and 0.999830 for the prediction and preparation policy. This corresponds to reducing downtime over one year from 1 h and 17 min, to only 1 min with prediction and avoidance (unavailability reduced by a factor of 73) and to 44 min with prediction and preparation policy (unavailability reduced by a factor of 1.7).

As a more realistic scenario, when failure prediction is not perfect, we consider the case of a precision-recall curve from Figure 4.1. For the purpose of a simplified application of the A-measure, we approximate the curve with a function $P = 1 - R^3$. We depict, with solid lines on the higher surfaces in Figure 4.6 and Figure 4.7, how availability changes with respect to the model parameters from Table 4.3 and the approximated curve from Figure 4.1. In fact, the lines are projections of the approximated precision-recall curve on the availability surface. With the approximated curve, the optimal precision-recall trade-off may be derived by a substitution of precision in Equation 4.8 by $P = 1 - R^3$, and finding a derivative of Equation 4.8 with respect to recall with other parameters taken from Table 4.3.

Availability, unavailability, downtime per year, and downtime decrease with respect to the reactive policy, for optimal precision-recall point obtained by using A-measure are presented in Table 4.5. Unavailability and downtime decreases are not as high as in the case of a perfect failure prediction but are still significant. For example, with the prediction and avoidance policy unavailability reduces by a factor of almost 4.5 times, and downtime over one year is decreased by 1h when compared to the reactive policy (see Table 4.4 for availability with reactive policy). As a comparison, we also present the values of availability measures obtained with F-measure as well as identified precision and recall values. The results clearly demonstrate superiority of the proposed A-measure when it comes to availability improvement.

Table 4.5. Availability for different optimization metrics.

		Optimization Metric	
		A-measure	**F-measure**
Prediction and avoidance	**Availability**	0.999967	0.999936
	Unavailability	$33*10^{-6}$	$64*10^{-6}$
	Downtime per year [h]	17 min	34 min
	Downtime decrease [h]	1h	43 min
	Precision	0.1672	0.6856
	Recall	0.9408	0.6800
Prediction and preparation	**Availability**	0.999888	0.999879
	Unavailability	$112*10^{-6}$	$121*10^{-6}$
	Downtime per year [h]	58 min	63 min
	Downtime decrease [h]	17 min	14 min
	Precision	0.3292	0.6856
	Recall	0.8754	0.6800

To understand how the effect of predictive policies on availability changes with server's reliability, we consider a range of MTTF from about 10 days to 100 days (more precisely from 250 h to 2500 h). Steady-state availability and downtime decrease per year are depicted in Figure 4.8 and Figure 4.9 for the three types of policies. Precision and recall is optimized with A-measure.

Availability is always higher using either type of predictive policy than with the reactive policy. As one may expect, prediction and avoidance policy, that, for the analyzed case, has higher reward and lower penalty, also has higher availability. When MTTF is 2500 h, availability with the reactive policy is 0.999961. The same availability level may be reached with prediction and avoidance policy when MTTF is even 5 times smaller and equals to 500 h. Moreover, it may be observed that availability and downtime are affected less by changes in MTTF when one of the predictive policies is used. We have analyzed availability and downtime also for the case when success rate is only 0.6 and failure rate increase factor is 0.2. Even in such an extreme case, predictive policies still have higher availability and lower downtime than the reactive one.

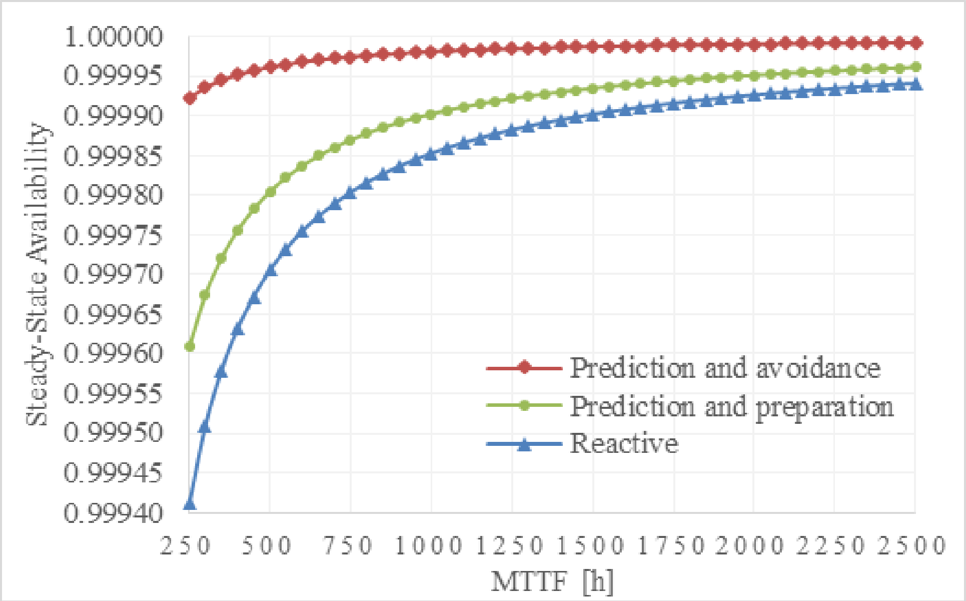


Figure 4.8. Impact of MTTF on steady-state availability.

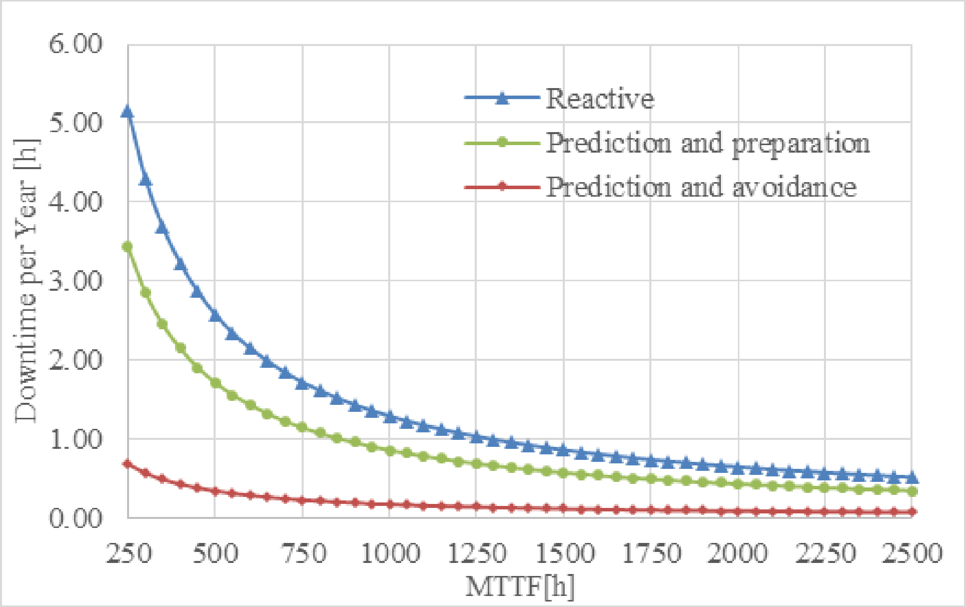


Figure 4.9. Impact of MTTF on downtime per year.

4.3.3 Sensitivity Analysis

We further analyze the effect that changing different system parameters has on availability by conducting sensitivity analysis. We use scaled differential sensitivity analysis as described in [145]. In the essence, considering a model of N input parameters x_i (i = 1..N) and an output Y, first a sensitivity of the output with respect to each parameter x_i, namely $S_{x_i}(Y)$, is calculated as a partial derivative of Y with respect to x_i. Then, scaled sensitivity is derived by scaling sensitivity with the ratio of the nominal value of the parameter to the nominal value of the output. Scaled sensitivity rank (SS_x) for each parameter is defined as the value of a scaled sensitivity with nominal values of parameters.

Sensitivity analysis is performed for the steady-state availability of the generic Markov model of a predictive fault-tolerance policy, using a SHARPE package for the sensitivity analysis developed in the scope of [145]. The analysis is conducted with respect to the parameters: MTTF, MTTR, precision, recall, penalty, reward, a, and c. Nominal values for the parameters are adopted from Table 4.3, and precision and recall nominal values are the ones from Table 4.5 when A-measure is used for optimization. Sensitivity analysis of system's availability with a reactive policy with respect to MTTF and MTTR is also conducted. Values of scaled sensitivity ranks are presented in Table 4.6. Negative value of sensitivity rank implies that the function decreases with an increase of the parameter. A zero value implies that the output is not sensitive to the parameter change.

Table 4.6. Sensitivity analysis results.

	Scaled Sensitivity Rank of System's Steady-state Availability		
Parameter (x)	**SSx(Ar)**	**SSx(Apa)**	**SSx(App)**
MTTF	$1.15*10^{-4}$	$1.94*10^{-5}$	$9.77*10^{-5}$
MTTR	$-1.15*10^{-4}$	$-1.49*10^{-4}$	$-1.49*10^{-4}$
precision	0	$8.24*10^{-6}$	$7.83*10^{-6}$
recall	0	$1.29*10^{-4}$	$5.09*10^{-5}$
penalty	0	$-6.93*10^{-6}$	$-5.38*10^{-6}$
reward	0	$1.36*10^{-4}$	$5.63*10^{-5}$
a	0	$-1.92*10^{-7}$	$-9.68*10^{-7}$
c	0	$1.37*10^{-4}$	$5.87*10^{-5}$

As previously observed, availability is becoming less sensitive with respect to the change of MTTF with predictive policies as the rank changes from the order of 10^{-4} with a reactive policy, to the order of 10^{-5} with a predictive one. In practice this means that reliability of a component will affect system's availability to a lesser extent. It is also interesting to observe that sensitivity is different for the two proactive policies. Namely, availability is less sensitive to the change of MTTF in the case of prediction and avoidance than in the case of prediction and preparation policy. This may be explained with a difference in penalties and rewards for the two policies (see Table 4.3). In particular, reward is more than two times higher in the case of prediction and avoidance policy when compared to the prediction and preparation one that makes it more resilient to the increase of MTTF as, with a good recall, the effect of MTTF change on availability is lower. Sensitivity with respect to MTTR remains almost the same regardless of the policy.

For the both predictive policies, availability is more sensitive to the change of recall than to the change of precision. As in the case of MTTF, it is interesting to observe how different sensitivity for the case of the two proactive policies to the change of recall is. Again, this may be explained with significantly different rewards. Namely, reward in the case of the prediction and avoidance policy is 525 s, whereas reward in the case of prediction and preparation policy is 230 s. A total reward over a period of time (that results in downtime decrease) may be calculated as a product of the reward for a single successful prediction and a number of successfully predicted failures in that time period (reflected in recall). Obviously, changing recall when reward is higher will have a more significant effect on the total reward. In other words, downtime decrease over a period of time is more sensitive on the change of recall when reward is higher. Thus, availability is also more sensitive on the change of recall for the case of prediction and avoidance policy.

Higher sensitivity to the change of recall than to the change of precision is also in line with some implementations of predictive policies (e.g. [68]) where, based on experiments and not on a formal analysis, the authors were suggesting that recall should be given priority over precision. However, one has to keep in mind that precision still has to be at least as required by Equation 4.7 for a predictive policy to improve availability with respect to the reactive one. It is even more important to observe that improving recall has a comparable effect on availability as changing server's MTTF. In fact, in the case of prediction and avoidance policy, improving recall is even more effective than improving server's MTTF.

Adding to this a previous conclusion, that a predictive policy makes a system less sensitive to the change of MTTF and considering results depicted in Figure 4.8 and Figure 4.9, we may say that improving system's availability may be more effective by implementing a predictive policy with high prediction quality (when reward is also high) than by investing into high reliability components with low failure rate. As it could be expected, one may also observe that availability is more sensitive on recall when reward is higher. In fact, sensitivity to reward is comparable to the one with respect to recall, whereas sensitivity with respect to penalty is comparable to the sensitivity with respect to precision. Availability is also very sensitive to the change of a proactive action success probability (parameter c). One of the ways to improve this parameter is to make sure that prediction lead time is always sufficient to perform a proactive action. Increasing lead time will also decrease a probability that the system is already contaminated when a failure is predicted. On the other hand, increased lead time deteriorates the quality of prediction as observed in [67, 144]. Failure rate increase due to a load introduced by failure prediction and other side effects, has a relatively low effect on availability. This means that even when failure prediction introduces significant load increase, availability may still be improved.

4.3.4 Simulation Results

The so far performed analyses assumes exponential distribution for the server failures, which is also a frequent practice in dependability analysis considering that models are simpler and easier to interpret when the failure rate is constant over the lifetime. For example, sensitivity analysis of the virtualized system in [145] has the same assumption for failure rate distributions. Nevertheless, numerous papers that consider real-life systems, including a study of the 9-year failure report of high-performance systems of the Los Alamos National Laboratory presented in [143], indicate that Weibull distribution is more appropriate for modeling failure rates in computer systems. Therefore, we perform additional analysis to validate the model with Weibull distribution as well. At the same time we also demonstrate how changing system parameters (proactive action overhead in this case) may be decisive on whether reactive or proactive FT is better from the availability point of view.

For the prediction and avoidance predictive policy the mitigation overhead is set from 5 s to 530 s with discrete steps of 5 s. For the prediction and preparation policy, checkpointing overhead is set from 5 s to 240 s. The upper limit for the ranges is decided so that reward remains greater than zero. When the reward is negative, reactive policy should, obviously, be used. Other parameters are as same as in the previous analysis (see Table 4.3) and failure prediction may be described with the PR-curve from Figure 4.1. In total, we perform more than 1000 simulations in MATLAB for the two types of predictive policies when the hazard rate is exponential and Weibull. The shape parameter for Weibull distribution is set to 0.7 following the results presented in [143]. For each simulation run, prediction quality is set to its optimal value that is selected following Equation 4.8, with penalty and reward obtained as in Section 4.2.2. In each run we simulate system's state for the lifetime of 100*MTTF (more than 11 years). Results are presented in Figure 4.10 and Figure 4.11. The horizontal line represents availability with reactive policy. Availability estimated with our model (Equation 4.3) is presented with a dashed line.

For the prediction and avoidance policy, simulation results indicate that, for the specific system and selected failure predictor, it is better to use reactive policy when the overhead is over 300 s. In fact, even with a perfect failure prediction it is still better to use reactive policy when the overhead is above 385 s. For the prediction and preparation the breakeven checkpoint overhead value is 115 s. Most importantly, selecting between the reactive and proactive policy may be done with the our analytical equation (model) with high confidence as minor difference between simulated and estimated availability at the breakeven point, for both predictive policies, may be contributed to the simulation variance. In fact, the maximum difference in simulated and estimated availability is at the order of 10^{-5} for the whole range of the overhead.

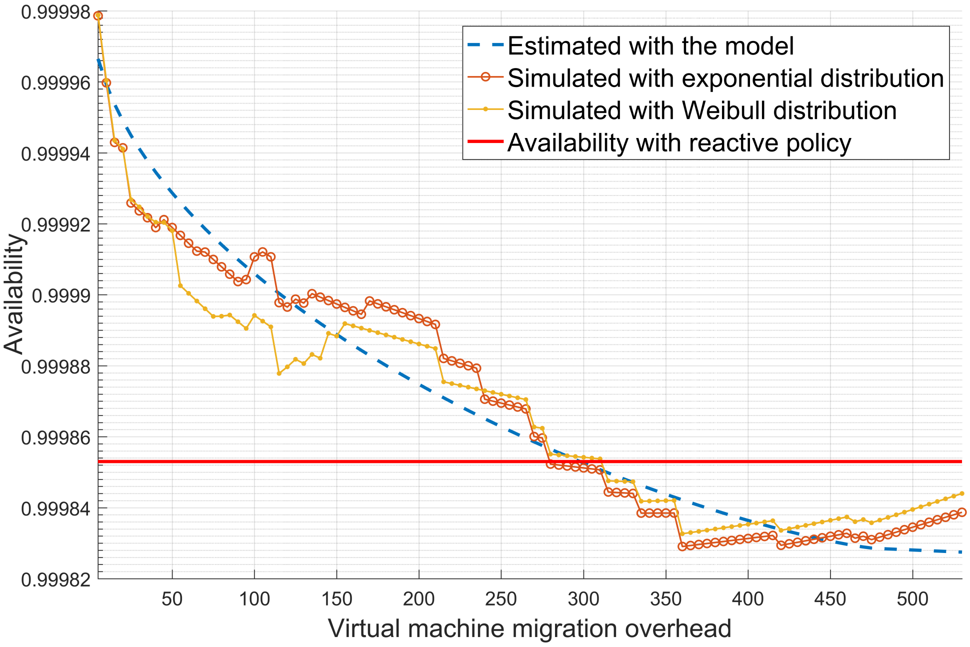


Figure 4.10. System's availability with different virtual machine migration overhead for prediction and avoidance policy.

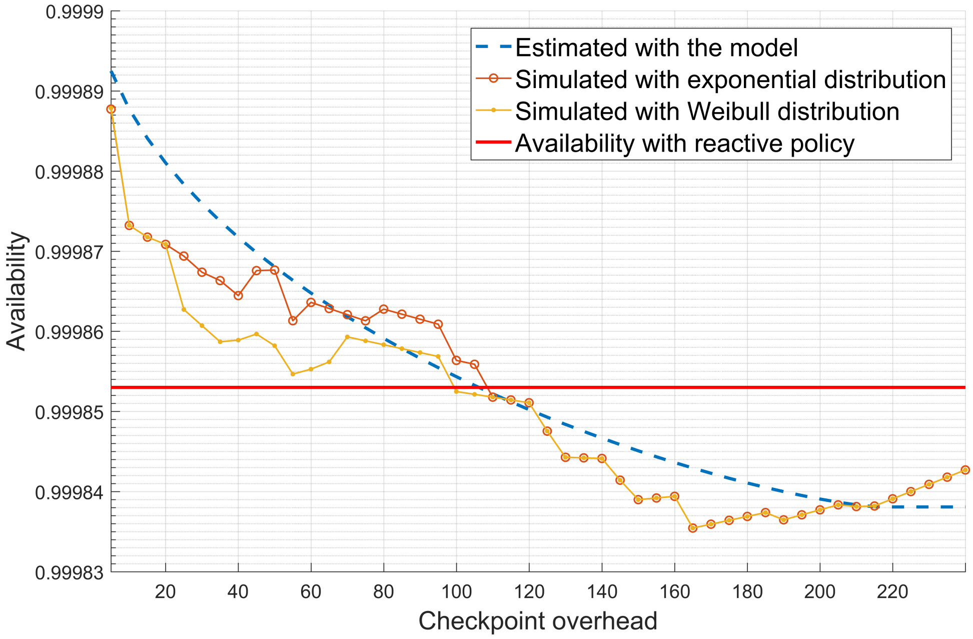


Figure 4.11. System's availability with different checkpoint overhead for the prediction and preparation policy.

Chapter 5

Methodology for Online Disturbance Prediction

An overview of the methodology for proactive management disturbances is depicted in Figure 5.1. We describe the main elements and identify methods for their implementation with focus on design of disturbance prediction algorithms.

Work presented in this section has also been described in part by Kaitovic et al. in [150].

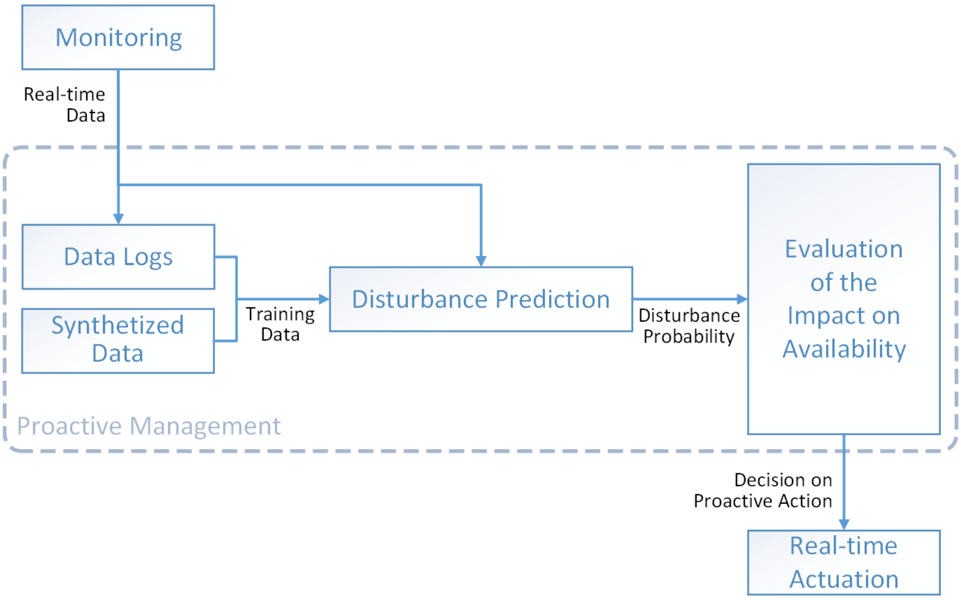


Figure 5.1. Proactive disturbance management methodology overview.

5.1 Monitoring and Data Acquisition

Monitoring infrastructure provides data to a prediction algorithm including smart meter and PMU measurements. Additionally, ambient measurements (e.g. environment and wire temperature, wind speed and direction) including weather predictions may also be a valuable input. The set of data, its quality and monitoring frequency have a decisive impact on the quality of prediction. Due to a large amount of data and still limited communication and computing resources, not all the data may be processed at runtime. Thus, monitoring must be adaptive in terms of monitoring frequency (sampling rate) and the number of features sent to the predictor must be adjusted based on the current state of the grid and the estimated probability of near-future problems. For example, if the system is in an Alert or Emergency state (see Figure 3.1), it may be needed to acquire the data with higher sampling rate and from additional sources (e.g. from a PMU in a different part of the network) in order to obtain a more accurate prediction in the light of increased probability of disturbances.

Online data that are used for prediction may also be stored to further refine the predictor. Moreover system may be simulated in order to collect additional data for the specific type of disturbances that rarely occur or for which there are no live records available.

5.2 Disturbance Prediction

Disturbance prediction identifies, at runtime, whether a disturbance will occur in the near future based on an assessment of the monitored current system state and the analysis of past events. The output of a predictor is the type of a disturbance and the probability of its imminence in the near future. Depending on the failure probability, the predefined threshold (that is set as described in Section 4.2), and available mitigation actions, the effect on availability is evaluated. The evaluation is conducted according to Equation 4.3.

5.3 Proactive Mitigation

Once the prediction mechanisms anticipate a disturbance, corrective actions to prevent it or to mitigate it should be scheduled and activated. A decision on taking a proactive action has to be performed with the impact on availability in mind. A great opportunity for the mitigation of disturbances in Smart Grid lies in the employment of FACTS devices that provide a sub-second response, the usage of distributed resources and solid state transformers as well as the employment of traditional elements such as circuit breakers and shunt capacitors that may be particularly useful at the distribution level. Moreover, at the distribution level, controllable DGs and solid-state transformers are considered to be the most efficient methods for the grid control.

5.4 Disturbance Predictor Design

The design of a predictor should be conducted in three phases as depicted in Figure 5.2.

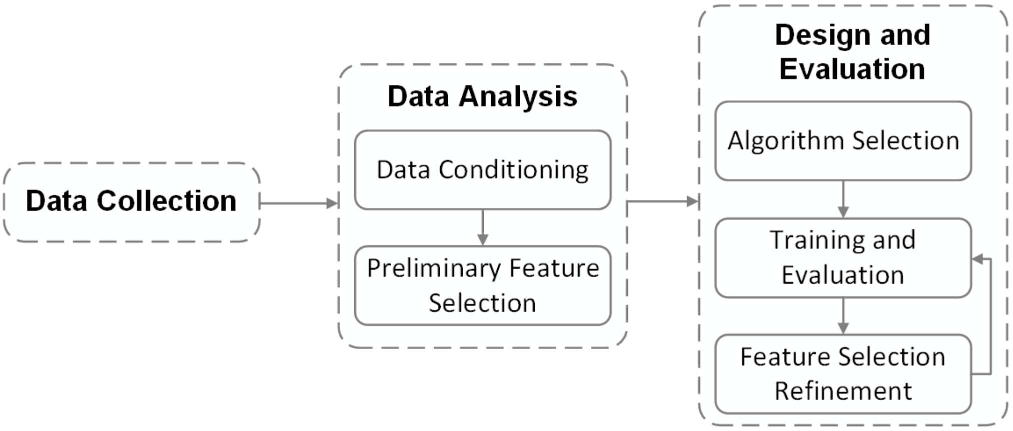


Figure 5.2. Disturbance predictor design phases.

5.4.1 Data Collection

In the first phase disturbance-related data are collected, preferably, from an existing system. Preliminary analysis is performed to identify the most frequent and the most sever disturbances. As disturbances are still relatively rare events, especially in the initial phase of a predictor design it may be necessary to simulate system's behavior in the presence of faults in order to obtain sufficient number of examples for the algorithm training.

5.4.2 Data Analysis

In the second phase, the obtained dataset should be analyzed. Data conditioning includes extraction of the features (also called events, variables or parameters by different research communities) and structuring the data in a form that may be used as an input for the prediction algorithm. In particular, each data set in the stream, that describes one system state, should be associated with a failure type or marked as failure-free.

A preliminary feature selection should be conducted while taking into account a system model. Feature selection is the process of selecting the most relevant features (and instances for algorithm training) and combining them in order to maximize predictors' performance; discard redundant and noisy data; obtain faster and more cost-effective algorithm training and online prediction; and to better interpret the data relations (data simplification for better human understanding). Feature selection methods may be classified as filters, wrappers and embedded methods [72]. A widely used filter method is Principal Component Analysis (PCA). PCA converts a set of correlated features into a set of linearly uncorrelated features (principal components) using orthogonal transformation. The procedure is independent with respect to the type of the prediction algorithm that will be used and thus very appropriate for preliminary selection of features. Numerous packages are available for feature selection, including those that are a part of popular tools for statistical analysis and machine learning (e.g. Matlab/Octave, Weka, Python and R). Good overviews of feature selection methods are given in [151] and [152].

5.4.3 Prediction Design and Evaluation

In the final stage, prediction algorithm is designed and evaluated. In fact, an ensemble of predictors may be used to improve quality of prediction. Having in mind a large number of existing prediction algorithms, the most viable solution is to select and to adopt one of them. A comprehensive survey of failure prediction algorithms is given in [72] that also groups prediction methods as those based on: failure tracking, symptom monitoring and detected error reporting.

Failure tracking draws conclusions about upcoming failures from the occurrence of the previous ones. These methods either aim at predicting the time of the next occurrence of a failure or at estimating the probability of failures co-occurrence.

Symptoms are defined as side effects of looming faults that not necessarily manifest themselves as errors. Symptom-monitoring based predictions analyze the system features in order to identify those that indicate an upcoming failure. Several methods for the estimation were proposed in the past, including function approximation, machine-learning techniques, system models, graph models, and time series analysis.

Finally, the methods based on detected error reporting, such as the rule-based, the co-concurrence-based and the pattern recognition methods, analyze the error reports to predict if a new failure is about to happen. Current trends in predicting disturbances in Smart Grids, such as, for example, cascading failures are mainly based on the application of machine-learning approaches, such as neural networks, support vector machines and anomalies' detection (see, for example, [65]). Nevertheless it might be more appropriate to use simpler algorithms for online prediction of near-future failures and disturbances to minimize prediction latency.

After selecting and training the algorithm, evaluation should be performed with respect to availability enhancement following Equation 4.3. A wrapper method may be applied at this stage to refine feature selection. A typical approach would be to rank the features according to availability improvement and to select a set of highly ranked features as the final one. In general, the goal is to set the number of features to a minimum while, at the same time, maximizing expected availability enhancement.

5.4.4 Tools

For the implementation of the methodology and design of a disturbance predictor a set of different tools has been used. A brief description of these tools is given here.

- Power System Analysis Toolbox (PSAT) [39, 153] was used to simulate system's behavior in the presence of faults and to generate disturbance-related data. PSAT is a Matlab/Octave toolbox for electric power systems simulation and analysis. Its functionalities include: power flow, continuation power flow, optimal power flow, small signal stability analysis, and time domain simulation. Its functions may be assessed through a graphical user interface and a Simulink-based library provides a user friendly tool for network design. It also supports a number of data formats and incorporates a number of static and dynamic models including: bus bars, slack buses, shunt capacitors, PV and wind generators, and voltage dependent loads. It is an open-source tool that may be adopted and extended.

- Matlab was be used for preliminary data conditioning and preparation for the feature selection phase. It is a high-level programing languages that come with a rich library for data manipulation.

- For feature selection, training and evaluation of prediction algorithms, Java and WEKA [154] were be used. Java was mainly used for scripting and invoking WEKA functions and customizing the format of the output. WEKA (Waikato Environment for Knowledge Analysis) is a suite for machine learning that also comes with a user-friendly GUI. It also includes a variety of tools for transforming datasets, such as the algorithms for discretization and sampling. The workbench has methods for the main data mining problems: regression, classification, clustering, association rule mining, and attribute selection. It also provides visualization facilities and methods for algorithms evaluation. It is fully expandable and may be used as a Java package. Unfortunately, it may not be used directly for time series classification but the existing functions may be adopted and combined for this purpose.

Chapter 6

Simulation and Fault Injection Framework

Analyzing the grid and understanding its behavior, when under a disturbance, is a prerequisite for designing methods for boosting grid's availability. This is the step that requires a vast amount of data to model grid's behavior not only when under a disturbance but also when in a fault-free state. The importance of data for design of novel grid control methods and strategies has been also emphasized in IBM's vision on big data in Smart Grids [35] where smart meter and other monitoring data are foreseen to be used for prediction of load and renewable generation as well as for anticipation of equipment and grid failures. In fact, methods based on data analytics and machine learning for predicting failures of aged grid components as a support for predictive maintenance are already being used. One such method and a relevant case study on maintaining New York City's electric grid has been presented in [65]. The study is based on data on equipment status collected by a local distribution company over a long period of time.

Unfortunately, data as the one used in [65] are rarely, if at all, available for public use mostly due to non-disclosure issues. Moreover, digitalization of distribution grids is still an undergoing process and only a few are equipped with devices such as PMUs that estimate measurement data with sufficient sampling rate for thorough analysis of disturbances. Even though there are many academic projects on Smart Grids (a comprehensive list of European Smart Grid projects that have started before 2014 may be found in [155]), not many include pilot implementations that also incorporate advanced measuring infrastructure. Those that do as, for example, the EPFL Smart Grid [27], are still relatively small and also well-controlled so that disturbances are very rare events. The lack of relevant data is a practical obstacle to disturbance analysis and design of methods for their prediction.

An alternative to obtaining disturbance-related data is to employ fault injection, which is a deliberate introduction of faults. Fault injection is used in computer systems to evaluate system's dependability and to compare different fault-tolerant mechanisms when it is not possible to wait for or to get field data [156]. Faults may be injected into an existing system or fault-injection approach may be combined with system simulation. For example, in [73] a simulation based fault-injection method to collect computer system failure data for failure prediction assessment is presented.

We adopt a similar approach to design and implement a modular simulation framework that allows injection of different types of faults to cause different types of system disturbances and to recored various types of grid measurements including voltages, currents and phasors with high sampling rate (sampling period of down to 20ms). For example, a voltage sag may be caused by a demand increase, by renewable generation variation, by creating a short between the lines or by their combination. The framework also allows to classify each measurement set as related with a different type of disturbance or with a normal operation state, following selected disturbance classification criteria. The framework is designed to be flexible and modular so that simulations may be performed at different grid levels. It is intended mainly to facilitate generation of power system disturbance data (with focus on ADNs) that may be used for design and assessment of detectors and predictors. It may also be used for evaluating the relation of different types of faults, errors and failures.

A part of the work presented in this section has been produced during a supervision of a master's thesis and thus is also included in [157]. Work has also resulted in a publication in Springer Journal on "Computer Science - Research and Development" in March 2017 [158].

6.1 Functionalities, Structure and Components

The framework supports disturbance analysis by simulating behavior of a modeled grid, for defined load and generation profiles, while dynamically injecting faults to cause disturbances. The output data is represented in a form of time-tagged variable vectors with each vector in one row. Each row is also classified as related to the normal system state or to a specific disturbance according to the selected disturbance detection (classification) criteria.

A structural model of the framework is presented in a UML-like fashion in Figure 6.1. Rounded rectangles represent instances of input and output modules, and rectangles represent instances of the framework engine. Solid connection lines are used to indicate direct association between the modules in the sense of one providing inputs to the other. Dashed lines indicate more generic dependency between the modules and will be explained case-by-case. Numbers next to association lines indicate a number of instances of one module that, in one simulation run, may be in a relation with n instances of another module where n stands for any positive integer (the scope of n is local in the sense that its value is related with an association line and that, for one simulation run, this value may be different for each association between the elements). For example, in one simulation run one grid model is used, one or many load profiles, one or many generation profiles, etc. but the number of generation and load profiles used is not necessarily the same.

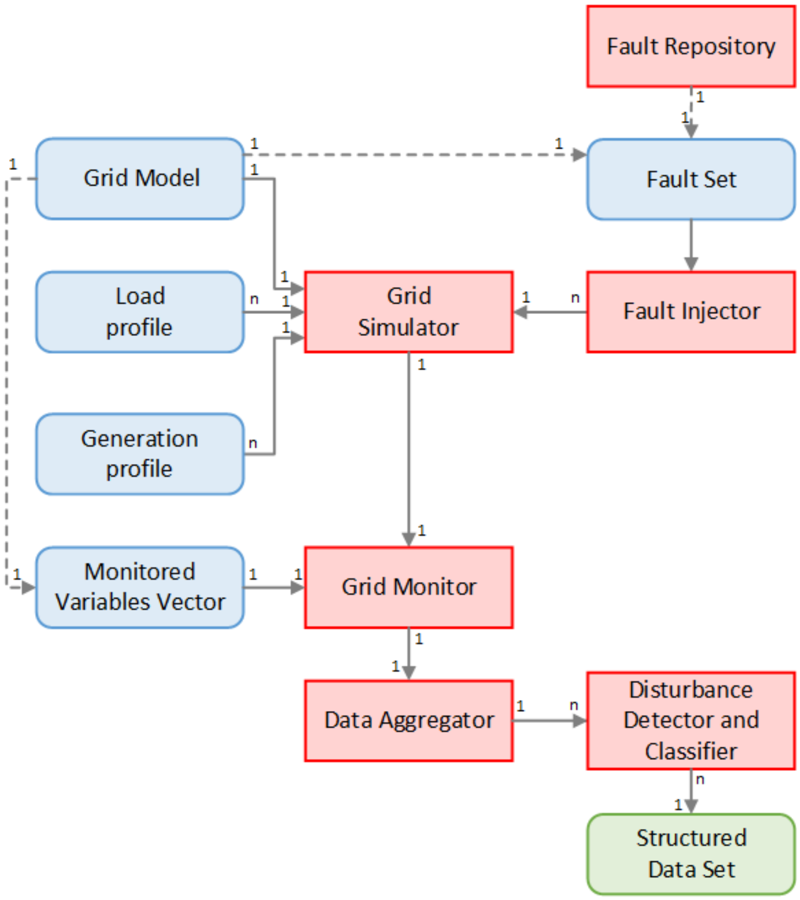


Figure 6.1. Structure of the simulation and fault injection framework.

The framework is modular so that additional instances (with different implementations) of the same module may be added or the existing modules may be extended. For example, to support injection of a new type of failure, a new instance of fault injector may be implemented and added.

6.1.1 Framework Frontend

Frontend modules include input ones that are: Grid Model, Load and Generation Profiles, Monitored Variables Vector, Fault Set and an output module that forms a Structured Data Set.

6.1.1.1 Grid Model

Grid Model defines the system whose behavior is to be analyzed. It may include a large network or only its part (e.g. a distribution grid). The level of detail should be sufficient to perform a standard power flow analysis. Grid models may be manually created or standardized and well-accepted models may be adopted. A list of standard IEEE test models may be found in [159] and [160].

6.1.1.2 Load and Generation Profiles

Load and Generation Profiles define variation of the load/generation of a typical consumer/generator over a period of time. The considered time period may vary from minutes to days, depending on the goal of the analysis. Load and generation profiles depend on the type of consumer/generator and are expressed in "per unit" (p.u.) so that the same profile may be applied to different load/-generator capacity. The framework allows to apply different profiles to different loads/generators so that a number of profiles may be loaded for one simulation run. Profiles must be defined in a simple textual format and each power change is defined as a pair of time (in seconds) and a new power value. For example, a profile of a load that initially (from time zero) consumes 1000 W of active power and 0 W of reactive power and increases active power consumption to 1200 W after 10 minutes (600 seconds) of operation will look like this: "0 1000 0 600 1200 0".

6.1.1.3 Monitored Variables Vector

A set of variables may be recorded for each simulation run including voltages, currents, active and reactive powers, and phasors. At this stage, a variable is defined by its type and a measurement location considering a model of the grid.

Monitored Variables Vector is used to predefine variables that will be monitored in a simulation run.

6.1.1.4 Fault Set

Fault Set contains a list of faults that are injected in one simulation run. Each fault must be related to a specific component of the grid (e.g. a bus, a DG, or a relay) while considering the Grid Model and available list of faults from the Fault Repository (that describes all types of faults that may be injected). The set must also define the exact time for each fault injection. A number of faults may be injected for the same component. It is important to note that injecting a fault does not necessarily imply occurrence of a disturbance (an error or a failure) but this also depends on system dynamics (load and generation profiles, topology of the grid and a specific simulation scenario). For example, if a fault is injected in a circuit breaker, but this component is not used in a specific simulation, the fault will not be activated and no disturbance related to this fault will be observed.

6.1.1.5 Structured Data Set

Output data is presented in a way that is suitable for further analysis, with proper classification of disturbances in CSV (comma separated value) format so that it may be easily read by widely used data analysis tools such as Weka, MATLAB/Octave, R and Python. Variables are places in columns whereas each row represents one data instance with a predefined sampling rate. Also, each column is tagged with a proper classification depending on the set of disturbances that are detected in a simulation run.

6.1.2 Framework Backend

Framework modules that take part in model simulation, dynamic manipulation, monitoring, data recording, conditioning and manipulation are Fault Repository, Fault Injector, Dynamic Simulator, Grid Monitor, Data Aggregator and Disturbance Detector.

6.1.2.1 Fault Repository

This module is a simple list that names all fault types that may be injected and for which a proper injector is implemented. Different types of faults may be related to the same module. For example, a short circuit or an open circuit fault may be injected into a bus. An extensive list of faults in Smart Grid is given in taxonomy in Figure 2.4 and it may be used to extend the current implementation of the Fault Repository.

6.1.2.2 Fault Injector

Fault Injector implements a mechanism to inject faults listed in Fault Repository at runtime while taking into consideration the grid model. One fault injector must be implemented for each fault type and a new fault injector may be added to support a new type of fault. This module is an important part of the framework and its implementation requires advanced knowledge of the simulation mechanism as well as sufficient expertise in power systems.

6.1.2.3 Dynamic Simulator

Dynamic Simulator performs time domain simulation of the grid model and allows monitoring and recording of grid parameters such as voltages, active and reactive power, phasors and currents. It also allows to make changes in the model at runtime so that faults may be injected.

6.1.2.4 Grid Monitor

Grid Monitor tracks and records the predefined set of parameters during the simulation following Monitored Variables Vector. It allows a user to set monitoring sampling rate at the begging of each simulation run. This is an important feature that allows to compare disturbance analysis (prediction) results depending on the sampling rate. It may be used, for example, to identify minimum sampling rate required for accurate disturbance detection or prediction.

6.1.2.5 Data Aggregator

Data Aggregator assembles the data from the Grid Monitor and presents it in one matrix. Each row of the matrix corresponds to one-time stamp and includes values of all monitored variables.

6.1.2.6 Disturbance Detector and Classifier

Disturbance Detector and Classifier implements a mechanism for automated detection of different types of disturbances by analyzing monitored and recorded grid variables. For example, a voltage sag detector may be implemented with a voltage threshold value so that all voltage drops under this value are considered as voltage sags. More complex rules may be defined for other types of disturbances. Detector is defined independently from the simulator and new detector instances may be implemented and added to the framework. If a disturbance is detected, appropriate data vector is classified accordingly. Detector labels each data set with a code to indicate normal (disturbance-free) operation of the system or to identify a type of detected disturbance and the associated component (location). At least one disturbance detector (rule) must be defined for each disturbance type and a number of detectors may be applied to the same data set at once. Disturbances detection and data classification should be done after simulations and generation of raw data records. In this way, different sets of structured data may be generated from the same raw data using different detectors.

6.2 Implementation Details

Following the abstract structure of the framework presented in the previous section, we implement a tool that we name a Dynamic Power System Fault Injector (DyPSyFI). Implementation details are given in this section.

6.2.1 Power System Simulator

For simulation and additional functionalities such as importing and saving grid model, we adapt PSAT that has been already introduced in Section 5.4.4 as an open source and widely used tool mainly in academia. A part of the PSAT structure relevant to the framework implementation is given in Figure 6.2. A complete structure and more details may be found in [39] and [153].

Most of the PSAT functions are not directly available to a DyPSyFI user but provided through a more user-friendly GUI so that the user is not required to be familiar with PSAT. In essence, we create a wrapper around PSAT functions and use only their subset while providing additional functions to simplify disturbance-related data generation process. The simulator already provides support for MATLAB Simulink models that we use directly for representing grid model. In fact, only Simulink GUI is used for representing the model whereas all the simulation mechanisms are implemented in PSAT. PSAT is also able to convert a variety of widely-used data formats in power systems such as IEEE common data format, WSCC and EPRY ETMSP. We also use PSAT plotting utilities and outputs directly.

Once loaded to the framework, the model is automatically initialized by calling a power flow routine from PSAT. Initialization includes setting of all the state and algebraic system variables.

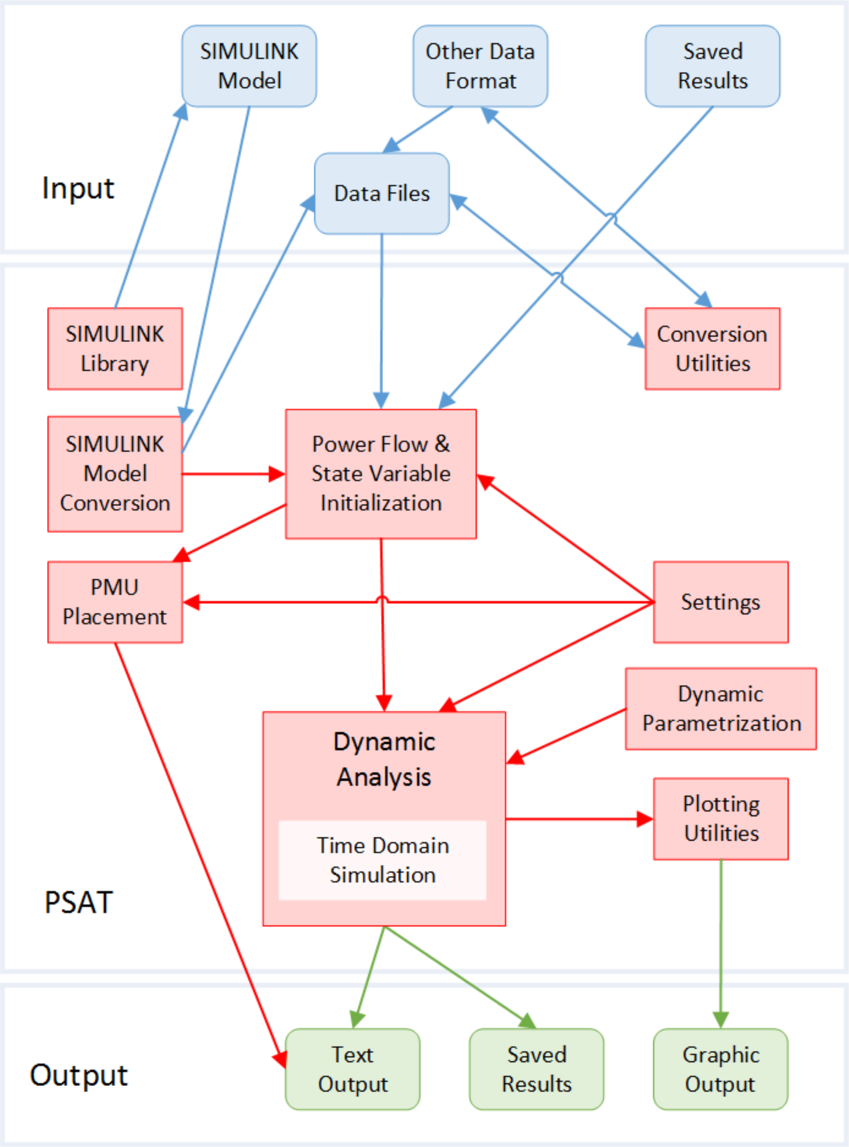


Figure 6.2. A part of the PSAT structure.

The most relevant part of the simulator is the Time Domain Simulation that also allows tracking and storing the selected set of parameters. We use this feature for the realization of the Grid Monitor framework module. Also, specific points in time may be computed and stored during the time domain simulation using a snapshot mechanism.

6.2.2 Dynamic Model Manipulation Mechanism

Dynamic changes of model components may be imposed through a PSAT mechanism that we name Dynamic Parametrization to avoid confusion due to different terminology used in this work and in the PSAT manual [153]. This mechanism allows to define exact value of a specified model parameter and the exact time when this value should be set. Theoretically, all dynamic parameters of any component may be changed which gives the possibility to simulate a wide range of different fault-injection scenarios. For example, this may be used to dynamically open or close a circuit breaker in the grid model or to change a generator's output power.

This is the exact mechanism we exploit for implementing injection on all types of phase faults (balanced, unbalanced, cross-phase and phase-to-ground). A breaker is placed between the lines where a fault needs to be injected, following the settings from the Fault Set, and closed at the simulation runtime, again following the description in the Fault Set. We use this mechanism also for implementing load and generation profiles by changing generated/demanded active and reactive power.

Fault injection may require multiple parameters to be changed at the same time. All changes are stored in one file that is parsed at runtime. After every change, a new power flow initialization is performed.

6.2.3 Data Generation and Conditioning

PSAT simulation output results are stored in one output file that is used by a Data Aggregator. In general, all voltage, power, phase and current variables are stored with a sampling rate defined by the user. Only a subset of these variables is presented to the user depending on the selection defined in Monitored Variables Vector. Still, the user may add other variables using a data manipulation environment. This is an important feature as data manipulation and addition of new variable does not require another simulation run for the same simulation setting. Disturbance Detector is implemented as a separate module independent from PSAT. Current implementation allows detection of voltage sags and swells. Sags are classified following the recommendations from the IEEE Recommended Practice for Monitoring Electric Power Quality (IEEE Std 1159-2009)[161].

The output is represented in a matrix following the format given in Table 6.1. Each row contains a time stamp (in seconds), variables' values and a disturbance classification code. Zero is used to represent a normal state of the system, namely when no disturbance is detected. A disturbance code is composed of a disturbance type and a code of the component where the disturbance is detected. For example, "D_VS01_B14" is used to identify a voltage sag on Bus 14.

Table 6.1. Output data format.

Time stamp [s]	Var_1	...	Var_n	Disturbance Classification

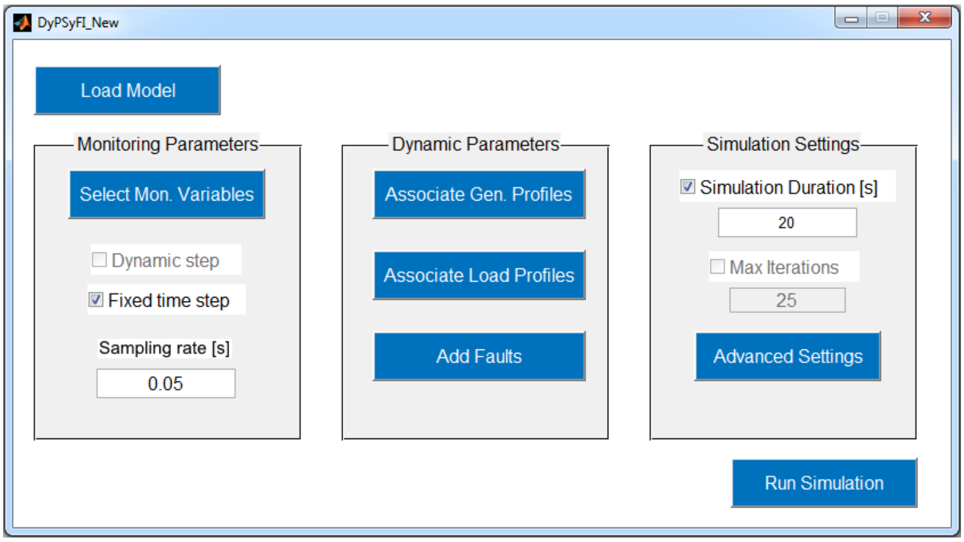
6.3 Graphical User Interface

A user-friendly GUI provides three main groups of functionalities: (i) dynamic simulation, (ii) fault injection and (iii) manipulation of previously generated and recorded data.

6.3.1 Simulation Environment

The snapshot of the window for parameterizing and initiating a new simulation is presented in Figure 6.3.

Loading a model automatically performs its initialization and allows other functions that are organized in three separate groups. Monitoring parameters groups allows to select variables to be monitored (this also opens a separate window) as well as defining monitoring sampling rate. Sampling period is defined as time between two consecutive measurements in seconds. Sampling rate is the



□ Dynamic step
☑ Fixed time step
Sampling rate [s]
0.05

☑ Simulation Duration [s]
20
□ Max Iterations
25

Figure 6.3. New simulation window snapshot.

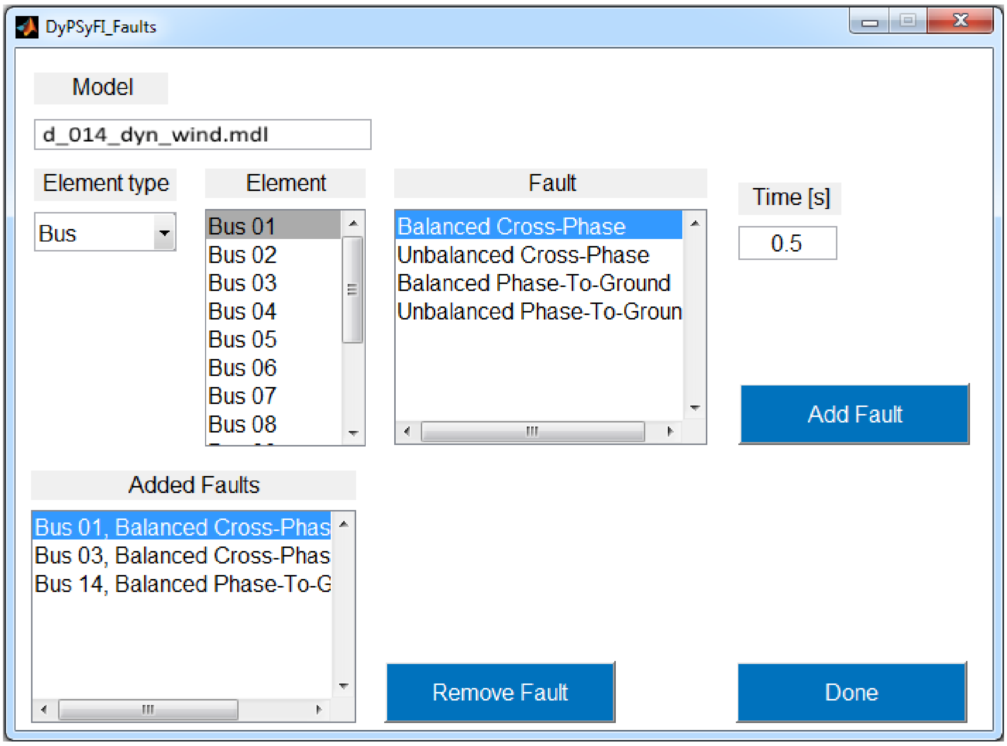
number of measurements per second. It can be set statically (to a fixed value for the entire simulation) or dynamically by using a native PSAT algorithm. In the latter case, the sampling rate is increased/decreased at runtime depending on the system's state so that simulation time is minimized.

Dynamic parameters group is used for assigning generation and load profiles to grid components and to add faults. In the case of generation/load settings, a separate window is used that contains a list of generators/loads on one side and possible generation profiles on the other side. Generation and load profiles are defined in separate .txt files that may be modified externally. Adding Faults function opens the Fault Injection Environment window.

Simulation settings are used to define wall-clock simulation duration in seconds or the maximum number of simulation iterations. Advanced settings button opens PSAT's simulation settings window.

6.3.2 Fault Injection Environment

A snapshot of the interface for adding faults is presented in Figure 6.4.



Model: d_014_dyn_wind.mdl

Element type: Bus

Element:
Bus 01
Bus 02
Bus 03
Bus 04
Bus 05
Bus 06
Bus 07
Bus 08

Fault:
Balanced Cross-Phase
Unbalanced Cross-Phase
Balanced Phase-To-Ground
Unbalanced Phase-To-Groun

Time [s]: 0.5

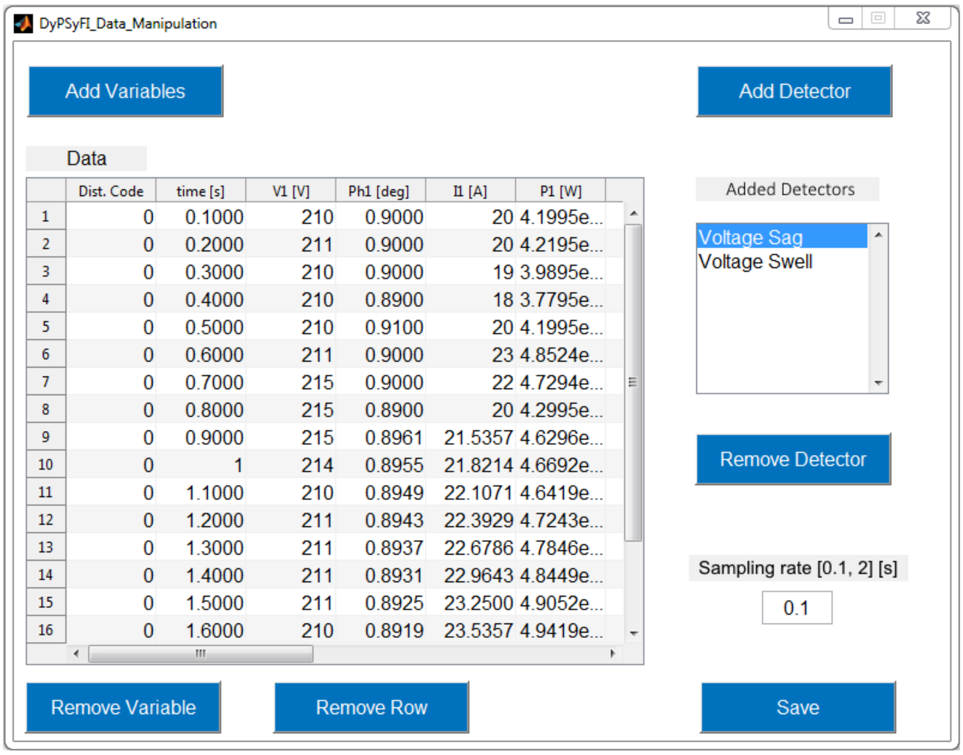
Added Faults:
Bus 01, Balanced Cross-Phas
Bus 03, Balanced Cross-Phas
Bus 14, Balanced Phase-To-G

Figure 6.4. Adding faults window snapshot.

Model name is displayed on the top of the window. The model may be opened in a separate window as a visualization guide when selecting components to inject faults into. For each component (element) of the grid a list of faults, that may be injected, is displayed when the component is selected. Components are grouped according to their types. A user has to select a fault and also to define the time of fault injection in seconds (assuming that simulation starts at time 0) before adding it to the list of faults to be injected. The list of added faults is displayed as ordered with respect to the time of fault injection. Added faults may also be removed from the list. Closing the window automatically saves the list of faults and brings back the simulation window in focus.

6.3.3 Data Manipulation Environment

Data manipulation environment allows basic manipulation of previously generated (and classified) data. This includes adding or removing features, changing sampling rate and adding or removing disturbance detectors and classifiers. A snapshot of the data manipulation window is presented in Figure 6.5.



Data

	Dist. Code	time [s]	V1 [V]	Ph1 [deg]	I1 [A]	P1 [W]
1	0	0.1000	210	0.9000	20	4.1995e...
2	0	0.2000	211	0.9000	20	4.2195e...
3	0	0.3000	210	0.9000	19	3.9895e...
4	0	0.4000	210	0.8900	18	3.7795e...
5	0	0.5000	210	0.9100	20	4.1995e...
6	0	0.6000	211	0.9000	23	4.8524e...
7	0	0.7000	215	0.9000	22	4.7294e...
8	0	0.8000	215	0.8900	20	4.2995e...
9	0	0.9000	215	0.8961	21.5357	4.6296e...
10	0	1	214	0.8955	21.8214	4.6692e...
11	0	1.1000	210	0.8949	22.1071	4.6419e...
12	0	1.2000	211	0.8943	22.3929	4.7243e...
13	0	1.3000	211	0.8937	22.6786	4.7846e...
14	0	1.4000	211	0.8931	22.9643	4.8449e...
15	0	1.5000	211	0.8925	23.2500	4.9052e...
16	0	1.6000	210	0.8919	23.5357	4.9419e...

Added Detectors

- Voltage Sag
- Voltage Swell

Sampling rate [0.1, 2] [s]: 0.1

Figure 6.5. Data manipulation window snapshot.

An important feature of DyPSyFI is that it records a complete list of features regardless of user's selection but presents only a subset of these variables following user's preferences. This allows adding new features at the later stage, namely from the recorded data without a need of repeating a simulation of the entire model with the same settings. Variables may also be removed as well as the entire data rows. Maximum sampling rate is limited by the one selected before the original simulation run. Disturbance detectors may be removed or new ones may also be added, through a separate window, and applied to the existing data. However, due to performance issues it is strongly suggested to have only one disturbance detector active. After manipulation, the new data set may be saved in a separate file.

Chapter 7

Case Study: Predicting Voltage Sags in an Active Distribution Network

The focus in this chapter is on predicting disturbances in Active Distribution Networks. We choose ADNs as they are heavily affected by massive penetration of non-dispatchable renewable generation and, at the same time, are being increasingly equipped with different types of monitoring devices that include smart meters and PMUs. Thus, on the one hand, a number of disturbances is expected to increase even more in the near future and, on the other hand, means to detect and to predict these disturbances are becoming available. Grid's frequency is regulated at the (centralized) generation and the transmission level but volatile generation from DERs may still cause voltage fluctuations at the distribution level [58, 162].

In fact, voltage sags are the most frequent power quality problems that customers may observe. They may damage voltage-sensitive power equipment and also cause substantial financial losses to distribution companies and customers [163]. Among the most affected industries and users are manufacturing industries (e.g. semiconductor and automotive), hospitals, air traffic control, and financial institutions. As the number of DERs increases, we may only expect more sags and similar disturbances.

As already pointed out, pilot projects that incorporate renewable generation and advanced monitoring equipment at the distribution level are mostly developed for well-designed and overall stable networks so that disturbances, if at all, occur only rarely (for example, the EPFL Smart Grid[1]). Some, on the other hand, do register disturbances but the monitoring infrastructure is not sufficient for

[1] EPFL Smart Grid, http://smartgrid.epfl.ch/

their prediction. For example, the VEiN[2] project documents a few over-voltages (swells) caused by higher generation from DERs. The implemented monitoring and the disturbance detection systems work with a one-minute sampling period. Unfortunately, the data are stored with a sampling period of 15 minutes only. Our preliminary analysis indicated that, with such sampling period, very low quality of voltage sag prediction is obtainable (both, precision and recall below 0.1). For that reason, we use DyPSyFI, the implemented simulation and fault-injection framework that has been described in the previous chapter and simulate a behavior of a standard IEEE distribution system that has been extended to include renewable generation.

For voltage sag prediction we adopt a set of machine-learning classification algorithms. Classification algorithms are widely used to train models that are then applied for classifying new elements in two or more classes. In general, machine learning may be described as a process of getting computers to act and to solve problems without programming them explicitly. In machine learning, training data are used to develop a model (rule) that is then used on new data instances. A data instance represents a vector of values of different features (features are also called properties or attributes in different communities).

For example, in classification problems training instances are first classified by experts. The training instances are used with a machine learning algorithm to develop a classification model. The model is then used to classify an unknown data instance. To obtain good accuracy of the classifier it is important that the training set is sufficiently large and that the classes are balanced (similar number of instances for all classes). A good example of classification is spam detection. Different features may be used to describe an email. This may include, for example, a number of persons to whom the email has been sent, frequency of specific words (e.g. bank, buy, money), if a receiver is addressed by the first name or not, and if the sender is in the receiver's address book or not. Training instances include emails that a user manually marks as spam or legitimate messages. Spam detection rules are then derived from the training instances. The rules (model) are then used to classify a new message. Clearly, the classification is not 100% accurate. Interestingly, the accuracy usually depends more on the training data and the selected features than on the classification algorithm [154].

[2] VEiN: Verteilte Einspeisung in Niederspannungsnetze, http://www.vein-grid.ch

7.1 Voltage Sags: Properties, Classification and Causes

According to the IEEE Standard 1159-1995 [59], a voltage sag is a reduction of a voltage rms value to below 90% of the nominal value for a duration of at least 0.5 cycle (we assume that the frequency is 50 Hz so that one cycle equals 20 ms). Following the same standard, voltage sags may further be classified depending on the duration of the under-voltage as instantaneous, momentary and temporary as summarized in Table 7.1. Column "magnitude" in the table referrers to the magnitude of the voltage reduction and not the retained voltage level whereas "p.u." stands for "per unit." If a voltage drops below 90% of the nominal value, a complete power interruption occurs.

Table 7.1. Classification of voltage sags and their characteristics.

Categories	Duration	Magnitude [p.u.]
Instantaneous Sag	0.5 - 30 cycles	0.1 - 0.9
Momentary Sag	30 cycles - 3s	0.1 - 0.9
Temporary Sag	3s - 60s	0.1 - 0.9

In this case study we focus on momentary sags but the same approach may be used on other types of sags as well.

Adopting terminology from computer systems dependability, we use downtime to evaluate availability. When focusing on voltage sags only, a (part of the) system is considered as being "down" (unavailable) during a voltage sag. It is important to observe that downtime evaluates availability from a user's perspective as a sag may affect only a few system buses and not the entire system (e.g. a distribution network).

Simulation of voltage sags requires advanced knowledge of the simulation environment and a high level of power systems knowledge [164]. This process is significantly simplified with the simulation environment that we have developed [158]. The most severe voltage sags are caused by short circuit balanced or non-balanced faults (see Figure 2.4), as well as significant load or (distributed) generation fluctuations, operation of circuit breakers and reclosers, and equipment failures [164]. If not prevented, a voltage sag will quickly propagate to other parts of the network. This generally depends on the grid topology, built-in protective mechanisms, position of voltage regulators and a distance from the fault location but also on the current state of the system (e.g. a power balance) and system dynamics (e.g. generation and consumption fluctuation).

7.2 Simulation of Voltage Sags and Data Generation

We simulate a behavior of an ADN, a model of which is presented in Figure 7.1. It is based on a standard IEEE 14-bus model [159, 160]. The model is relatively small to perform power flow simulations and data collection rather quickly (order of minutes per simulation) and, at the same time, large enough, so that faults may propagate from one part of the network to the other with a delay that is sufficient to perform prediction of voltage sags.

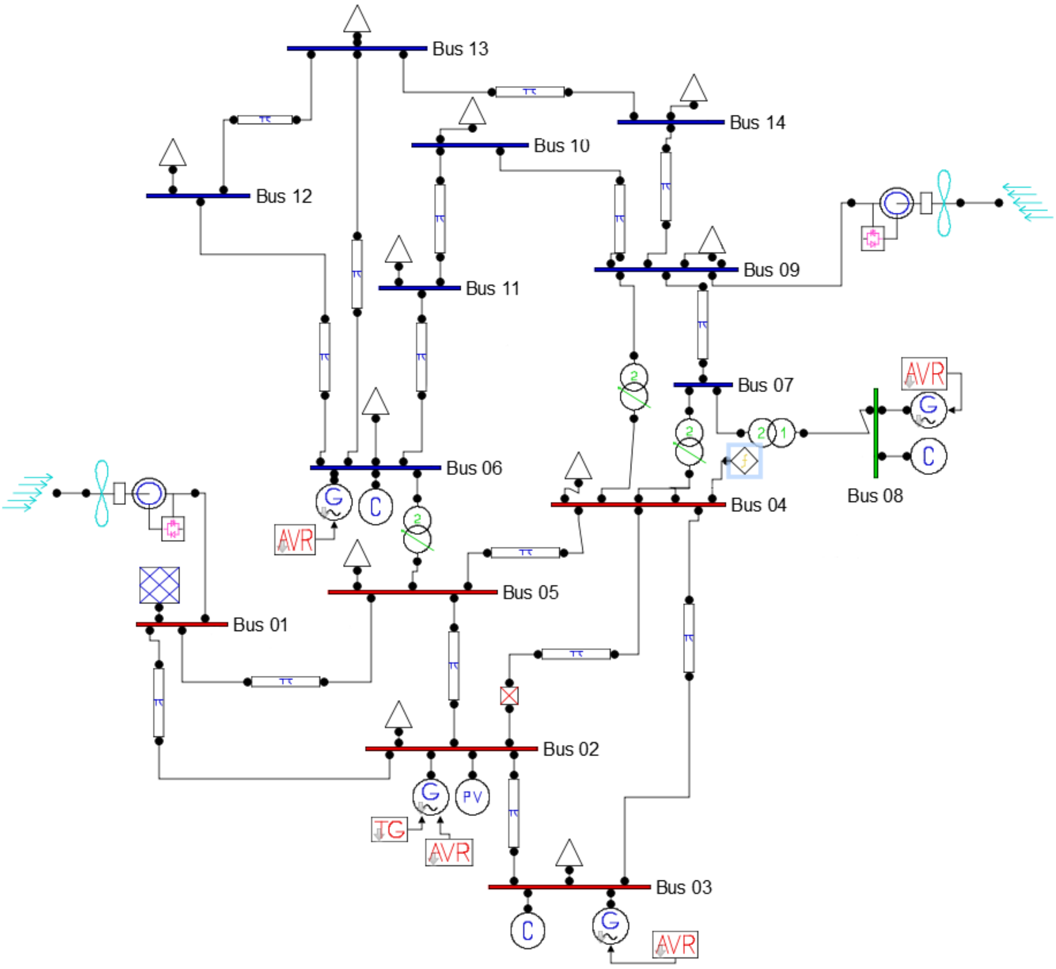


Figure 7.1. Active Distribution Network based on the IEEE 14-bus model.

The model has been further extended to include two wind turbines on Buses 1 and 9, and one photovoltaic generator on Bus 2. Also, Automatic Voltage Regulators (AVRs) and synchronous compensators are included on buses where the network connected to the rest of the grid so that voltage fluctuations that come from the outside of the network are, partially, compensated. As summarized in Table 7.2, the total generation power is 392 MW active and 204 Mvar reactive power, and the total load equals 362 MW active power and 114 Mvar reactive power. Total losses equal 30 MW active and 90 Mvar reactive power. Details on nominal load values on individual buses are presented in Table 7.3.

Table 7.2. Overview of the network power balance.

Total generation	Active power	392 MW
	Reactive power	204 Mvar
Total load	Active power	363 MW
	Reactive power	114 Mvar
Total losses	Active power	29 MW
	Reactive power	90 Mvar

Table 7.3. Nominal power demand on load buses.

Bus ID	Active power [MW]	Reactive power [Mvar]
2	38	18
3	130	23
4	65	6
5	9	2
6	14	10
9	39	23
10	11	8
11	3	2
12	6	8
13	30	7
14	17	7

To cause voltage sags we have injected short-circuit faults on different buses. Faults were injected at random time in first 15 cycles of the simulation and also removed at time that has been set randomly to between 1 and 15 cycles after the fault injection. We refer to the fault-removal time as clearance time.

An example of voltage profiles on Buses 1 to 7 after a fault has been injected on Bus 7 in cycle 7 and cleared in cycle 8, is presented in Figure 7.2. The horizontal dashed line in the figure indicates a sag threshold. Sags of different duration may be observed on Bus 1 and on Buses 4 to 7. As it may be observed, two sags occur on Buses 4, 6, and 7. The first sag is immediately after the fault injection whereas the second one comes with a delay of, about 10 cycles. The first sag cannot be predicted but the second one may if proper data are observed in the time window between the sags.

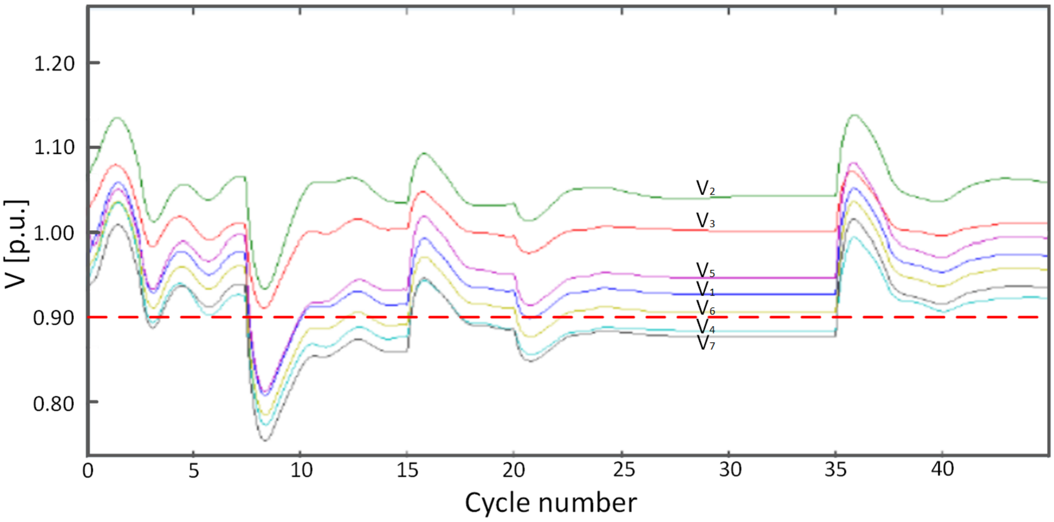


Figure 7.2. An example of voltage sags caused by a fault on Bus 7.

To simulate dynamic aspects of the system, load and renewable generation were also varied during the simulation for up to 20% of their nominal values. We observe that these variations heavily impact the number of voltage sags and their propagation. In each simulation run we measure bus voltages, phase angles and active and reactive powers. To reduce the number of features, currents are not included as they are anyway highly correlated with voltages and powers.

During the experiments we observe that the longest time for sag propagation through the network is obtained when faults are injected on Bus 2. For that reason we give an overview on results when injecting faults on all the buses and provide details for the case when faults are injected on Bus 2.

In total, we ran 100 simulations. Each simulation lasted between 20 s and 60 s. We performed 3×10^6 measurements of voltages, active and reactive powers and phase angles on all 14 buses. To total number of features is 56 as four features are measured per bus and there are 14 buses in total. Simulation parameters are summarized in Table 7.4.

In each simulation run one fault is injected but zero, one or more sags may occur on different buses depending on the topology and state of the grid.

Table 7.4. Simulation parameters summary.

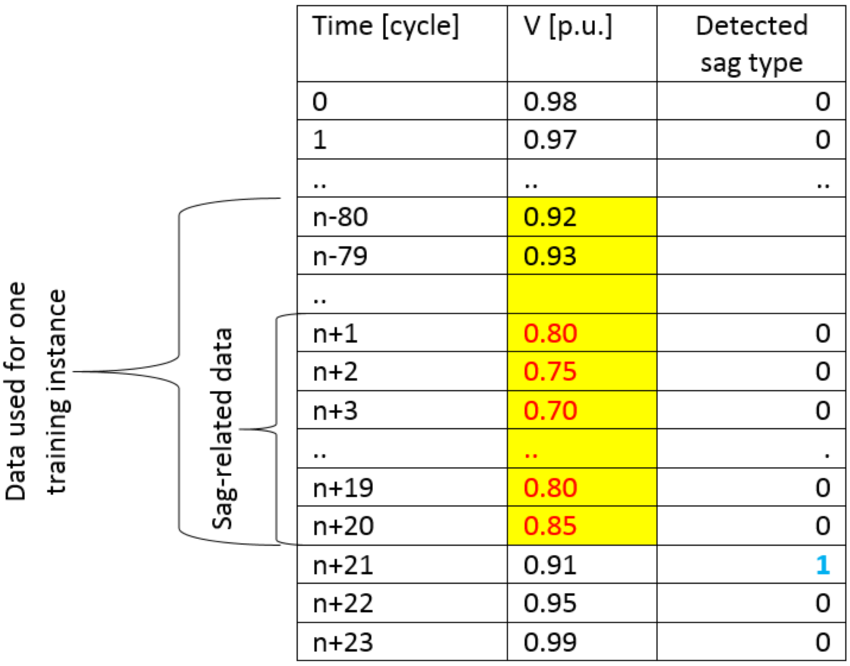
Total number of simulations	100
Duration of an individual simulation in seconds	from 20 to 60
Duration of an individual simulation in cycles	from 1000 to 3000
Time for execution of each simulation	$\approx$ 1 min
Fault injection time	random
Clearance time in cycles	random between 1 and 15
Load demand variation	20%
Renewable generation variation	20%
Total number of features	56 (14*4)
Sampling period	20 ms (1 cycle)
Total number of measurements	3 000 000

7.3 Data Conditioning

Data output from DyPSyFI is represented as a sequence of time-tagged values with sampling period of 20 ms (one cycle). This is also a typical sampling period period of PMUs that are expected to be installed in most of the ADNs in the future.

To use these data for classification algorithms' training and evaluation, they must be first presented in the matrix form and each data instance has to be classified. We process each data stream related to individual features separately. Sag detection on each bus is performed simply by comparing a current voltage value to the sag threshold (0.9 p.u.). Then, a subset of the data stream of a predefined length L backward from the point when the sag is detected, and that includes the entire set of measurements during the sag, is removed from the stream and placed in a table as one data instance with the class "sag". This process is also described in Figure 7.3 and an example of a data instance is given in Figure 7.4. Abbreviation "p.u." stands for "per unit".

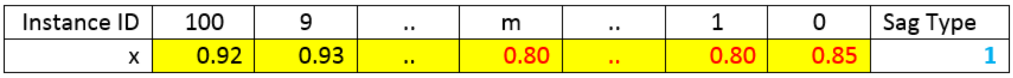
Once all the sag-related data are extracted from the stream, data related to sag-free system state are extracted in a similar way. Subsets of consequent data of



Time [cycle]	V [p.u.]	Detected sag type
0	0.98	0
1	0.97	0
..	..	..
n-80	0.92	
n-79	0.93	
..		
n+1	0.80	0
n+2	0.75	0
n+3	0.70	0
..	..	.
n+19	0.80	0
n+20	0.85	0
n+21	0.91	1
n+22	0.95	0
n+23	0.99	0

Data used for one training instance
Sag-related data

Figure 7.3. An example of a tagged data output sequence from DyPSyFI.



Instance ID	100	9	..	m	..	1	0	Sag Type
x	0.92	0.93	..	0.80	..	0.80	0.85	1

Figure 7.4. An example of a data instance.

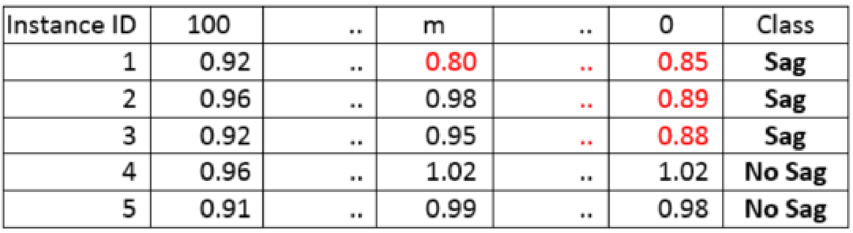
the length L are extracted from what has been left from the original data stream and each subset is placed in the matrix as one data instances of the class "no sag".

A simplified explanation of the idea behind the described procedure is that we have conditioned the data in such a way that each instance represents a stream of data that may lead to a sag (instances classified as the "sag" class) or may not lead to a sag (instances classified as the "no sag" class). The constructed data matrix is used for prediction model training.

We observe that none of the simulated sags have lasted more than 30 cycles. As for improving availability we aim at using OLTCs that may be activated with a short activation delay in a range of 10ms (less than one cycle), it is reasonable to have relatively short lead time that is in the range of one second (50 cycles). Following the results of predicting failures in computer systems [67], we may expect that longer lead time will decrease prediction quality. As for that, we set L to 100 cycles. This is sufficient to perform proactive actions and also gives a possibility to evaluate how increasing lead time may affect the quality of prediction. Moreover, setting L to 100 is a good compromise between the length of one data instance and the total number of instances in the specific case. Setting L to a higher value would decrease the total number of instances to the level that would have not been sufficient enough for training the model.

It is important to observe that in this interpretation lead time is defined with respect to the sag classification time that occurs at the end of the sag. Namely, lead time means how much in advance a class of the sag (that depends on its duration) may be predicted. This is because of the structure of the data and varying duration of sags. For this reason, only the end of the sag could be fixed in time. However, as sags are lasting never more than 30 cycles (in fact 99% last less than 25 cycles), this also means that a start of the sag is predicted with sufficient lead time. For example, when sag class (end) is predicted with lead time of 50 cycles, having in mind that it could not last more than 30 cycles, means that start of the sag is predicted with, at least, 20 cycles lead.

We inject faults on one bus at time and perform the data conditioning procedure. The procedure is performed on the data stream for every feature, detecting sags on all the buses. For each combination of a feature and a bus where the sag is detected, we generate a matrix that we use for training the models. For example, for the case when faults are injected on Bus 2, the matrix has 1400 instances. An extract of the table generated from measurements of the voltage on Bus 14 when sag detection is performed on Bus 6 is presented in Figure 7.5.



Instance ID	100	..	m	..	0	Class
1	0.92	..	0.80	..	0.85	**Sag**
2	0.96	..	0.98	..	0.89	**Sag**
3	0.92	..	0.95	..	0.88	**Sag**
4	0.96	..	1.02	..	1.02	**No Sag**
5	0.91	..	0.99	..	0.98	**No Sag**

Figure 7.5. An example of a part of a data matrix.

To get good classification results, it is preferred that the classes are balanced. This means that the number of instances of one class is similar to the number of instances of the other class. A number of instances of the two classes for different buses when a fault is injected on Bus 2 are given in Table 7.5. In this case, Bus 6 has a good balance between the "sag" and "no sag" classes (687 vs. 713). For this reason, in the rest of the chapter, we focus on the results from predicting sags on Bus 6 when faults are injected on Bus 2. For the comparison, we also provide results of different states of the methodology for the sag-predictor design on other buses as well.

Table 7.5. Summary of the number of "sag" and "no sag" instances per bus.

Bus ID	# "no sag" instances	# "sag" instances
1	205	1195
2	201	1199
3	98	1302
4	237	1163
5	316	1084
6	713	687
7	1103	297
8	1200	200
9	1098	302
10	965	435
11	934	466
12	1200	185
13	1300	100
14	570	830

For describing a predictor, besides parameters that were introduced in Subsection 4.1.1, namely precision, recall, and lead time, we introduce two additional parameters relevant for online prediction with time series. These are sampling period and prediction window. Sampling period represents the time between two consecutive measurements. Prediction window is time frame for which the data are considered when making a prediction. For clarification, in Figure 7.6, the case when lead time is 30 cycles and prediction window size is 50 cycles is depicted. The lightening symbol indicates time when a sag is detected (and classified), whereas the bell symbol is used to mark time of the sag prediction. For the convenience, we also indicate the time for which data are are considered when creating one data instance with length of 100 cycles.

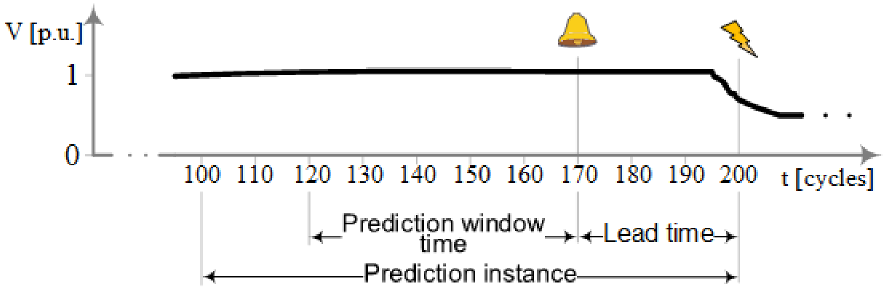


Figure 7.6. An example of prediction window and lead time.

To evaluate how lead time, prediction widow size and the sampling period may affect the quality of prediction, we generate additional data sets. From each of 56 data matrices for individual features, we derive additional ones by having ten different values for the lead time in the range from 0 to 90 cycles and 5 different sizes for the prediction window for each lead time value. The numbers of different values for lead time and prediction window size are selected as a trade-off between having sufficient number of points to depict the effect of the parameter change on prediction quality on one side and limiting the number of simulations and the total computation time on the other. For each combination of the lead time and the prediction window size, we change the sampling period in range from 20 ms to 1280 ms (in steps of (2^i*20) ms for i = 0, 1,..,6). This corresponds for a range of measuring frequency from 50 Hz down to about 0.8 Hz. In total, we generate more than 15000 data matrices for each combination of a fault-injection and fault-prediction bus.

7.4 Prediction Quality Evaluation Method

We use the term prediction quality (prediction performance) in a sense of quantitative measure that assesses how well a model generalizes on an independent data set. Typically, the entire data set is split into two sets, one for model generation (training) and the other for the model evaluation (testing). One way of splitting the data is to use two thirds of the data set for training and one third for evaluation. However, more accurate model verification may be obtained with k-fold cross validation. In this approach, the entire data set is split into k subsets. In each run, k-1 subsets are put together and are used for training, whereas the remaining set is used for evaluation. The average of the performance evaluation results in all the runs are then calculated and the average value is reported as the final result. Observe, that in each run, the remaining k^{th} set that is used for the evaluation, has not been used for the model training in that run and, for the model trained in the run represents a fully new data set. In practice, when the number of data instances in sufficiently high (typically more than 1000), 10-fold set validation is the most frequently used. More details may be found in [165].

For quality evaluation, at this stage, for practical reasons we use F-measure, precision and recall. Namely, we use algorithms developed as a part of the Weka tool that incorporate optimization with respect to F-measure, precision and recall. The F-measure will be used only for feature selection and preliminary algorithm evaluation whereas the final selection of the precision-recall pair will be performed with A-measure.

7.5 Preliminary Feature Selection

As features are collected via simulation, they are limited to phase angle, active and reactive power, frequency, voltage and current on each bus. These features are also most widely used to describe power systems. More importantly, it is most likely that the same set of features is stored in data logs that may be used to train predictors. Nevertheless, it is important to point out that other features may also be collected in a real system and they might be more indicative for prediction of sags or other disturbances. These include the rate of change of frequency as well as raw waveforms sampled with a PMU. Moreover, additional features may be derived as, for example, difference between two consecutive raw waveform samples or peaks of voltage samples.

As frequency is regulated at the transmission level and does not depend on events in the distribution grid, it has been excluded from the list of features. Current values were also not considered, as current may be derived from voltage and power. Preliminary feature selection is further performed for each of the selected algorithms separately. For this purpose, we took five most commonly known and widely used machine-learning classification algorithms.

A brief description of the selected algorithms is given in Table 7.6. Time complexity for training the algorithms using Big O notation is included in the same table (m stands for the number of instances and n for the number of features). Versions of the algorithms that are implemented in Weka version 3.8 have been considered when estimating complexity. More details on these and other algorithms may be found in [154].

The use of filtering methods for the preliminary selection was not practical as features are placed in separate tables, and finding correlation between individual features would require more effort (and no better results) than using a wrapper method for each algorithm. Also, it is important to observe that, from the algorithm's point of view, there are more than 56 features (14 buses with 4 features per bus). In this view, a feature is a combination of a system parameter (voltage, phase angle, etc.) and a time before the event.

Performance of each algorithm is evaluated with F-measure using ten-fold cross validation on every combination of lead time, prediction window size and sampling period for each of 56 features. Then, an average F-measure for every combination is set as a feature rank.

As an example, feature ranks when logistic regression algorithm is used for predicting sags on Bus 6 for the case when a fault occurs on Bus 2, are given in Table 7.7. It should be noted that a feature is uniquely identified with a bus number (1 to 14) and a feature type (V, P, S, T).

Table 7.6. Brief description of used machine-learning algorithms.

Name	**Description**	**Training-time Complexity**
Naïve Bayes	Uses Bayes' theorem to calculate a probability that an instance belongs to a class, assuming that features are fully independent. A threshold (typically 0.5) is applied to make the classification decision.	O(m) [166]
Logistic Regression	Creates a function where weights are associated with features. The function is used to calculate a probability that an instance belongs to a class. A threshold (typically 0.5) is applied to make the classification decision.	O(mn^2) [167]
SVM	(Support Vector Machine) Creates an optimal hyperplane in the feature space to split between two classes. See [168] for more details.	O(m^3) [169])
IBk (k-NN)	Uses k nearest neighbors in the feature space to classify a new instance.	O(mn) [170]
J48 (C4.5)	Creates a decision tree using information entropy. See [171] for more details.	O(mn^2) [172]

Following the ranking procedure, ten most indicative features for every algorithm are identified. In Tables 7.8a and 7.8b we present results for the case of sag detection on Bus 6 when faults are injected on all other buses. The case of injecting a fault and predicting a sag on the same bus has not been considered as, in this case, a fault has an immediate affect and no prediction is possible. Features are represented by a feature type symbol as in Table 7.7 (V - Voltage, P - Active power, S - Reactive power, and T - Phase angle) followed by the bus number. They are ranked with integers from one to ten, with one being the highest rank.

It may be observed that features that are closer to the fault-injection bus have higher rank. In fact, following the results presented in the two tables, we may derive more specific rules:

Table 7.7. Feature ranking with F-measure for Logistic regression for the case of sag prediction on Bus 6 when faults are injected on Bus 2.

Bus	Voltage (V)	Active power (P)	Reactive power (S)	Phase angle (T)
1	0.636	0.819	0.736	0.808
2	0.663	0.702	0.826	0.824
3	0.675	0.787	0.690	0.808
4	0.683	0.799	0.800	0.784
5	0.700	0.803	0.803	0.798
6	0.724	0.725	0.757	0.800
7	0.702	0.518	0.517	0.791
8	0.656	0.626	0.730	0.790
9	0.715	0.806	0.806	0.793
10	0.727	0.806	0.806	0.795
11	0.738	0.807	0.807	0.798
12	0.741	0.806	0.806	0.800
13	0.753	0.806	0.806	0.801
14	0.740	0.805	0.805	0.797

- Phase angle, active and reactive power values on the bus where the fault is injected are typically among four most indicative features;

- Phase angles on buses with large loads that are connected or close to the fault-injection bus, in most of the cases, among the five most indicative features;

- Active powers and phase angles of the buses with renewable generators (windmills in particular) are among the most indicative features if the bus is electrically close to the one where a fault has been injected;

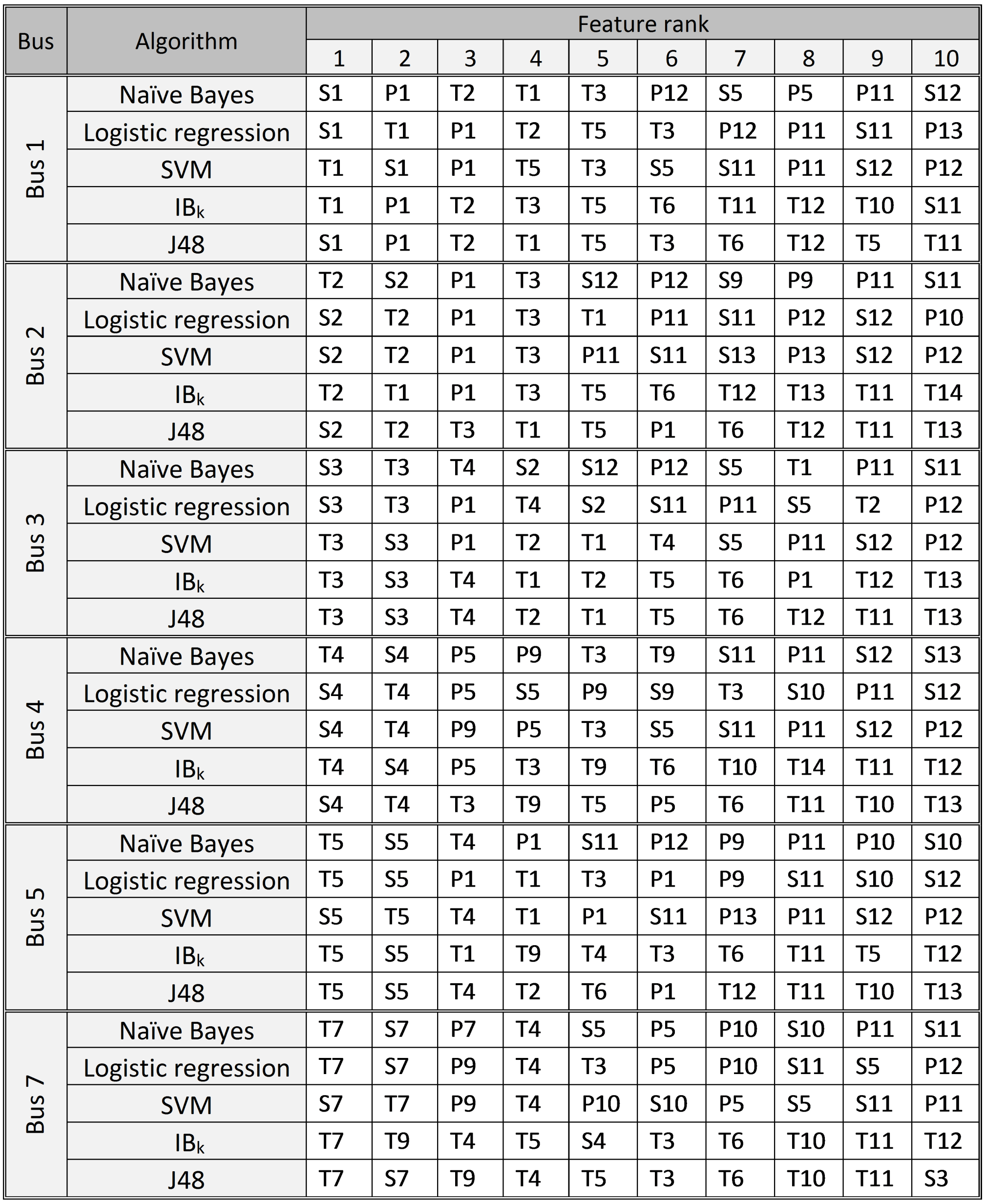
- Active and reactive power values, as well as phase angles of buses that are electrically close to the bus where a sag is being predicted are among ten most indicative features.

In addition, we may observe that for $\mathrm{IB_k}$ and J48 algorithms the most indicative features mostly include phase angles.

Also, it is worth pointing out that, when the objective is to create a general sag predictor on a selected bus that does not take into account position of the fault, then, active, reactive power values, and phase angles of the buses that are electrically close to the bus where a sag is being predicted should be used.

However, the prediction performance in this case will not be the maximum one as these features typically have rank above six.

Table 7.8a. The most indicative features for different machine-learning algorithms when sags are predicted on Bus 6 for fault injected on Buses 1 to 7.



Bus	Algorithm	Feature rank									
		1	2	3	4	5	6	7	8	9	10
Bus 1	Naïve Bayes	S1	P1	T2	T1	T3	P12	S5	P5	P11	S12
	Logistic regression	S1	T1	P1	T2	T5	T3	P12	P11	S11	P13
	SVM	T1	S1	P1	T5	T3	S5	S11	P11	S12	P12
	IB$_k$	T1	P1	T2	T3	T5	T6	T11	T12	T10	S11
	J48	S1	P1	T2	T1	T5	T3	T6	T12	T5	T11
Bus 2	Naïve Bayes	T2	S2	P1	T3	S12	P12	S9	P9	P11	S11
	Logistic regression	S2	T2	P1	T3	T1	P11	S11	P12	S12	P10
	SVM	S2	T2	P1	T3	P11	S11	S13	P13	S12	P12
	IB$_k$	T2	T1	P1	T3	T5	T6	T12	T13	T11	T14
	J48	S2	T2	T3	T1	T5	P1	T6	T12	T11	T13
Bus 3	Naïve Bayes	S3	T3	T4	S2	S12	P12	S5	T1	P11	S11
	Logistic regression	S3	T3	P1	T4	S2	S11	P11	S5	T2	P12
	SVM	T3	S3	P1	T2	T1	T4	S5	P11	S12	P12
	IB$_k$	T3	S3	T4	T1	T2	T5	T6	P1	T12	T13
	J48	T3	S3	T4	T2	T1	T5	T6	T12	T11	T13
Bus 4	Naïve Bayes	T4	S4	P5	P9	T3	T9	S11	P11	S12	S13
	Logistic regression	S4	T4	P5	S5	P9	S9	T3	S10	P11	S12
	SVM	S4	T4	P9	P5	T3	S5	S11	P11	S12	P12
	IB$_k$	T4	S4	P5	T3	T9	T6	T10	T14	T11	T12
	J48	S4	T4	T3	T9	T5	P5	T6	T11	T10	T13
Bus 5	Naïve Bayes	T5	S5	T4	P1	S11	P12	P9	P11	P10	S10
	Logistic regression	T5	S5	P1	T1	T3	P1	P9	S11	S10	S12
	SVM	S5	T5	T4	T1	P1	S11	P13	P11	S12	P12
	IB$_k$	T5	S5	T1	T9	T4	T3	T6	T11	T5	T12
	J48	T5	S5	T4	T2	T6	P1	T12	T11	T10	T13
Bus 7	Naïve Bayes	T7	S7	P7	T4	S5	P5	P10	S10	P11	S11
	Logistic regression	T7	S7	P9	T4	T3	P5	P10	S11	S5	P12
	SVM	S7	T7	P9	T4	P10	S10	P5	S5	S11	P11
	IB$_k$	T7	T9	T4	T5	S4	T3	T6	T10	T11	T12
	J48	T7	S7	T9	T4	T5	T3	T6	T10	T11	S3

Table 7.8b. The most indicative features for different machine-learning algorithms when sags are predicted on Bus 6 for fault injected on Buses 8 to 14.

Bus	Algorithm	Feature rank									
		1	2	3	4	5	6	7	8	9	10
Bus 8	Naïve Bayes	T7	S7	S8	T4	T9	P9	S9	P10	S10	S5
	Logistic regression	T7	T4	S8	T3	T1	P9	T9	P10	P14	P11
	SVM	T7	S8	T3	P8	P9	T9	P10	P11	S13	P13
	IB$_k$	T7	T9	T3	T4	T5	T6	T10	T12	T11	T13
	J48	P8	T7	T4	T9	T3	T5	T6	T10	T11	T13
Bus 9	Naïve Bayes	P9	T9	S9	P7	T3	P10	P11	S12	P12	S14
	Logistic regression	S9	T9	P9	T3	P7	T4	P10	P11	S11	P12
	SVM	T9	S9	P9	T4	T3	S14	P11	P10	S10	S11
	IB$_k$	T9	P9	T7	T6	T4	T3	T10	T14	T11	T12
	J48	T9	P9	T7	T4	T5	T6	T10	T14	T3	T11
Bus 10	Naïve Bayes	T10	S10	P10	P9	S11	P11	S5	P12	P13	P14
	Logistic regression	T10	S10	P9	P7	P11	P12	P5	S11	S10	P13
	SVM	P10	T10	P9	S10	P11	P12	S13	P5	P14	P12
	IB$_k$	T10	T9	T5	T13	T11	T14	T12	S10	T7	T1
	J48	T10	P10	T9	P7	T14	T11	T12	T13	T7	T5
Bus 11	Naïve Bayes	T11	S11	P10	S10	S12	P13	S10	P12	P5	S5
	Logistic regression	T11	S11	P10	T13	P12	S5	P5	P9	S10	P13
	SVM	S11	T11	P10	T13	S12	S10	P9	P12	S5	P5
	IB$_k$	T11	T10	T9	T6	T5	T12	P6	T14	T13	T1
	J48	T11	S11	T9	T10	T12	T13	T6	T5	T1	T13
Bus 12	Naïve Bayes	T12	S12	P11	P10	S11	P10	S13	P5	T1	P2
	Logistic regression	S12	T12	P10	T11	T6	P5	P9	S10	S11	P13
	SVM	S12	T12	P11	S11	S10	P10	P13	P5	T1	P2
	IB$_k$	T12	T11	T10	T6	T5	T2	T1	T14	T13	T9
	J48	T12	P12	T10	T6	T12	T13	T9	T5	T1	T13
Bus 13	Naïve Bayes	T13	S13	P12	P11	P5	P10	P13	P5	T1	P2
	Logistic regression	S13	T12	T13	P11	S11	P5	P2	P1	P9	P10
	SVM	T13	S13	P12	S12	P11	S5	P5	P2	T1	P10
	IB$_k$	T13	T12	T11	T6	T10	T5	T1	T9	T2	T14
	J48	T13	P12	P13	T11	T6	T5	T2	T3	T1	T10
Bus 14	Naïve Bayes	T14	P14	P9	T9	P5	P10	P11	S5	P12	P13
	Logistic regression	S14	P14	T14	P9	P11	P5	P12	S11	S12	P1
	SVM	P14	T14	S14	P9	P11	S11	P12	S12	P13	P5
	IB$_k$	T14	T9	T12	T6	P11	T5	T2	T1	T13	T7
	J48	T14	P9	P13	T11	T6	T5	T1	T2	T10	T12

7.6 Prediction Algorithm and Feature Set Selection

To identify the most appropriate algorithm as well as the number of the most indicative features whose combination gives the best prediction performance in a combination with the algorithm, we evaluate performance of each of the five algorithms from Table 7.6 for combinations of the most indicative features identified in Tables 7.8a and 7.8b. We focus on the case when faults are injected on Bus 2 and sags predicted on Bus 6 as, according to the values of F-measures for individual highly-ranked features (presented in Table 7.7), prediction performance with individual features is the highest for this case (when compared to other combinations of fault-injection and sag-prediction buses). Thus, we may also expect the best performance with combinations of the most indicative features that will serve well for demonstrating to what extent availability may be improved with proactive approach. However, it is also important to point out that prediction performance with the most highly-ranked individual features for other combinations of fault-injection and sag-prediction buses is, in the worst case, lower than for the selected fault-injection and sag-prediction bus combination, at the order of 10^{-2}. Therefore, we may expect that combinations of the most indicative features for different fault-injection and sag-prediction buses may under-perform, with respect to the selected case, in the same order of magnitude.

For each algorithm the features are combined by starting with the single most indicative feature and adding, one by one, the next one from the list. Hence, a combination of n features is, in fact, a combination of n most indicative ones. This is also a standard practice in feature ranking and the most common heuristic [151]. Prediction performance is evaluated with average F-measure for all variations of lead time, prediction window and sampling period. Results are presented in Figure 7.7.

Intuitively, we may expect that performance increases with higher number of features. However, in most of the cases, significant improvement may be observed only while the number of features is low. Specifically, in the case of IBk algorithm (that also shows the best overall performance), prediction performance increases until the number of features reaches four. After this number of features, the performance starts to decrease. The decrease may be contributed to a combination of feature correlation and overfitting. Overfitting (also known as high variance) occurs when a model includes too many features so that it almost perfectly describes training data set and fails to generalize [173]. In other words, instead of capturing the main phenomena that drives behavior of the system, the model also captures noise. This is, to some extent, not possible to avoid but too much noise may overshadow the main behavior of the system. Also, with more

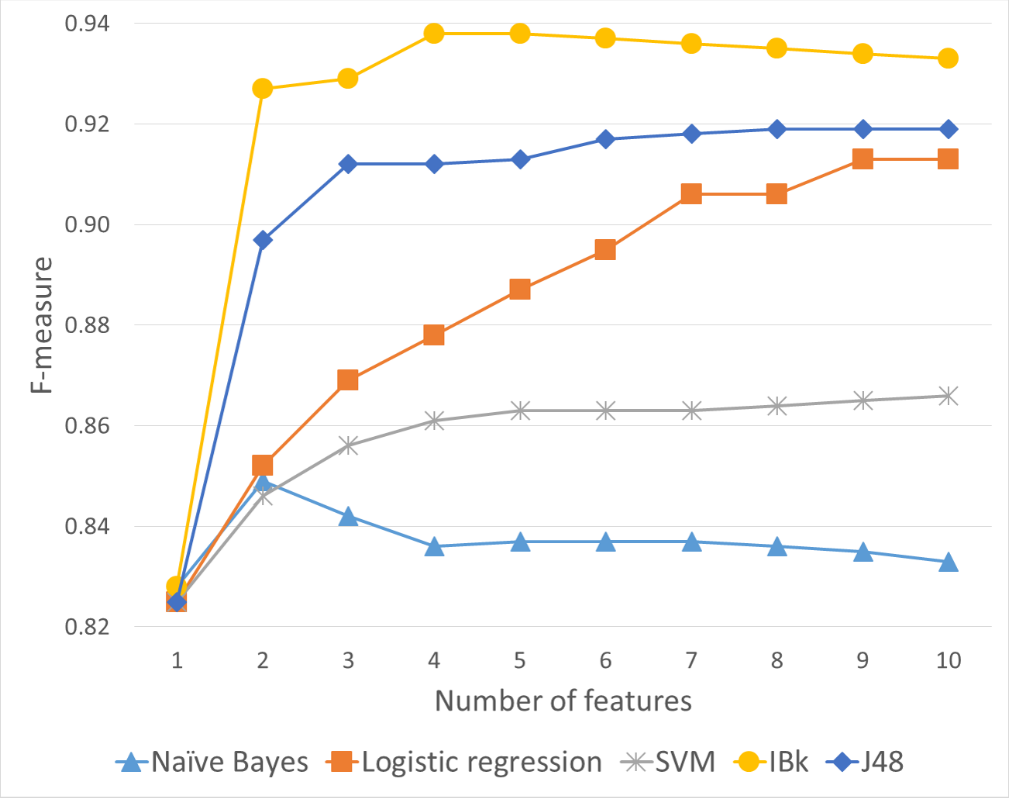


Figure 7.7. Evaluation of performance of different algorithms for combinations of the most indicative features.

features, there is a higher chance of their correlation. With high correlation, too much importance may be given to a certain property of the system. In an extreme case, when two features are fully correlated, adding them both to the model would be the same as adding one feature twice and thus misleading the model. For example, power is a product of voltage and current. If two out of these three features are already included in the model, adding the third one may only decrease the model's performance. The same goes to active power, reactive power and the phase angle on the same bus as these features are also correlated.

It is interesting to observe how performance when Naïve Bayes algorithm is used, starts to decrease already after the second feature has been added. This is because the algorithm assumes that features are not correlated and thus it is very sensitive to correlation.

With the exception of the case with Logistic regression, when performance reaches a steady level after the ninth feature has been added, and the case with Naïve Bayes that has already been discussed, we may observe that the performance with other three algorithms reaches a steady level with four up to six features. Obtaining good results with only a few features is also in line with [67] where the best performance of predicting failures in computer systems is obtained when only two features are used. The observation has to be analyzed in the light of the conclusions on the most indicative features listed at the end of the previous subsection. We may conclude that the best prediction performance may be obtained when combining features that are phase angles, active and reactive powers on the bus where the fault occurs and the buses that feed large loads or to which renewable generators are connected (and that are also electrically close to the fault-occurrence bus).

The case when IBk (k-nearest neighbors) algorithm is used, is obviously the most interesting one as it outperforms all the other cases. The best performance (F-measure of 0.938) is obtained when only four most indicative features are used. These are phase angle and active power on the fault-injection bus, and phase angles on two buses connected to it that have a large load and a renewable generator.

Clearly, we cannot derive a general conclusion based on a case of a relatively small grid but, following our results, it would make sense to, for a case of another grid, first look at results with IBk with phase angles on the fault-injection and neighboring buses.

7.7 Analyzing the Effect of Prediction Parameters on Its Quality

For the case of IBk algorithm with four features, we first analyze how the size of a prediction window affects prediction performance. For each lead time, we quantify the performance for five different prediction window sizes. The values are set in a range from zero (excluded) to the maximum prediction window size (included) so that the difference between two consecutive values is constant. As each data instance has a "length" of 100 cycles there is no fixed value for the maximum prediction window size but it varies with the lead time so that the sum of lead time and prediction window size is 100 cycles. For example, when lead time is 50 cycles, the maximum prediction window size is also 50 cycles. In this case the prediction window size takes values of 10, 20, 30, 40, and 50 cycles.

In Figures 7.8a, 7.8b and 7.8c we depict how, for lead time of 10, 30 and 90 cycles, the size of the prediction window affects prediction performance.

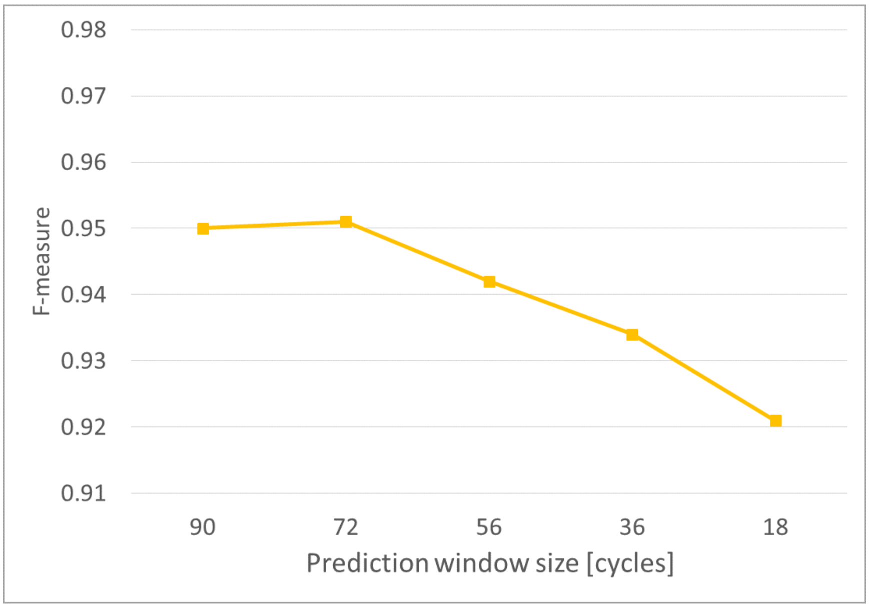


Figure 7.8a. The effect of prediction window size on prediction performance when lead time is 10 cycles.

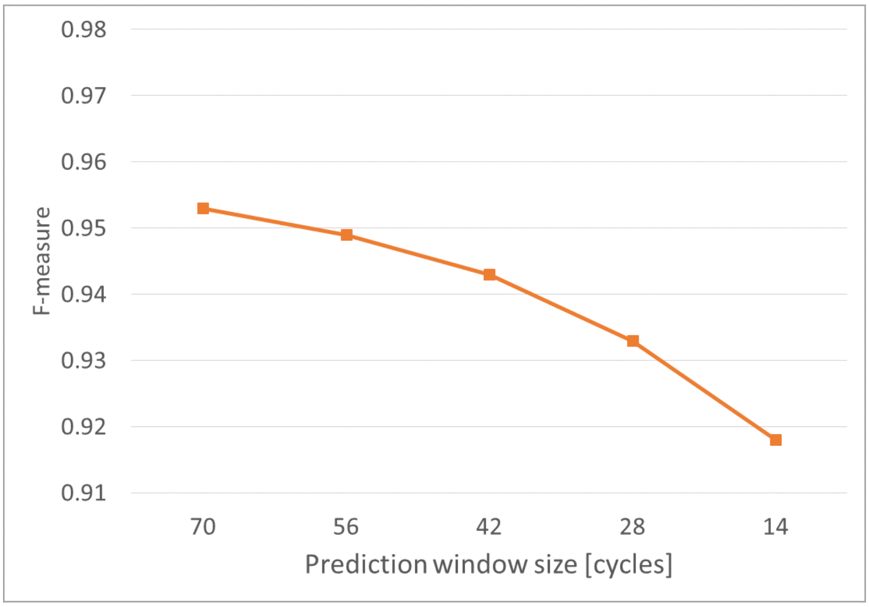


Figure 7.8b. The effect of prediction window size on prediction performance when lead time is 30 cycles.

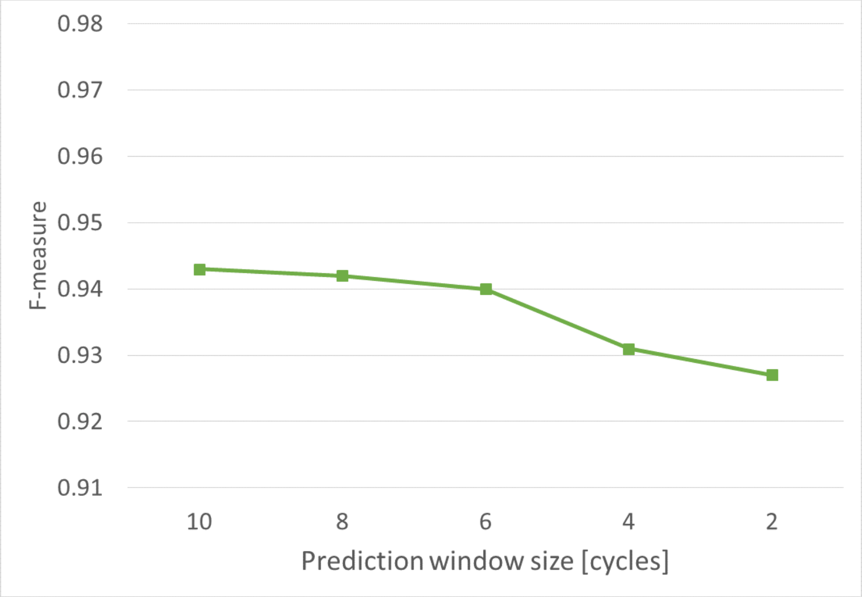


Figure 7.8c. The effect of prediction window size on prediction performance when lead time is 90 cycles.

Also, we have analyzed other cases for lead time and observed similar trends as in the ones presented in the figures. We repeat here that due to the data structure, lead time is defined with respect to sag classification (the end of the sag). Having in mind that sags last up to 30 cycles, prediction with lead time 50 cycles means that prediction is at least with 20 cycles lead time with respect to the sag start.

It may be observed that a performance of the predictor decreases with a decrease of size of the prediction window as less data are provided to the predictor. However, this may not always be the case as, especially for some algorithms, more data may cause overfitting and larger prediction window may decrease algorithm's performance. Following the analysis, when optimizing prediction performance, one should first consider the maximum prediction window but also evaluate performance as prediction window decreases in the range of about 20%.

With this respect, it is also interesting to observe how performance of a sag detector (that may be considered as a predictor with lead time zero) is affected by the prediction window size. We depict this case in Figure 7.8d.

Unlike in the previous cases, the performance decreases with increasing window size. For efficient detection, only data at the time of sag are sufficient. Adding any data before the sag introduces noise and thus decreases performance.

We further evaluate how lead time affects prediction performance. For all the

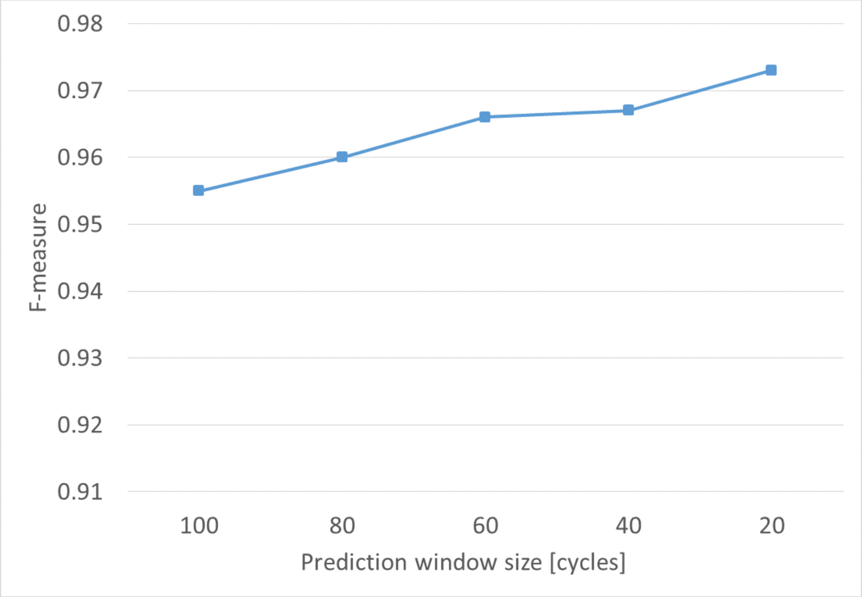


Figure 7.8d. The effect of prediction window size on sag detection performance.

lead time values, we set the prediction window size to 30 cycles and the sampling period to the maximum (20 ms). Having a fixed value for the prediction window size is important for the comparison as, having in mind the generated data set, the maximum prediction window size value would mean that different ones are used for different lead times. Specifically, as the size of an instance is 100 cycles, this means that the maximum size for prediction window when lead time is 10 cycles is 90 cycles, when lead time is 80 cycles maximum prediction window is 20 cycles and so on. To have a valid comparison, we fix the prediction window size rather than using the maximum value (that, in general may give the best performance for individual values of lead times). As indicated in Figure 7.9, performance drops as lead time increases.

A lesson that may be learned from this result is that increasing lead time comes with a relatively significant performance decrease. In particular, the difference between the highest and the lowest F-measure obtained is 0.03 (0.955 for lead time of 10 cycles and 0.922 for lead time of 70 cycles). For these two F-measure values and the specific case, precision and recall are 0.931 and 0.981 when F-measure is 0.955, and 0.895 and 0.951 when F-measure is 0.922. In practice this means that with lead time of 10 cycles (0.2 s) out of 1000 sags 981 will be predicted, whereas only 19 will be missed. The total number of alarms in this case equals 1054. This is a quotient of true-positive alarms (981) and precision (0.931). Thus, the predictor will raise additional 73 false alarms in this

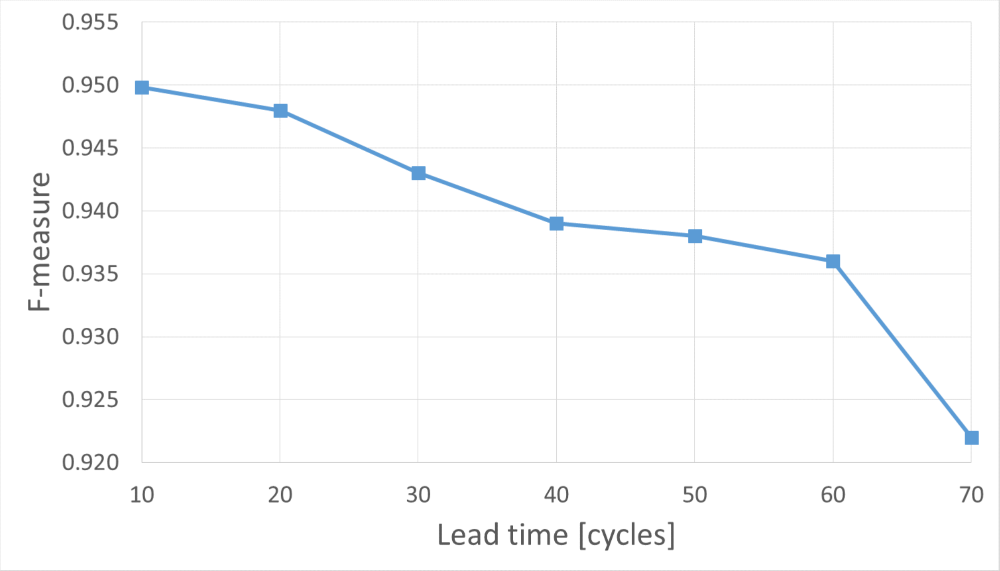


Figure 7.9. The effect of lead time on prediction performance.

case. When lead time is 70 cycles (1.4 s) out of 1000 sags, 951 will be predicted and 49 will be missed. The total number of alarms in this case is 1063 meaning that even 112 false alarms will be raised. This means 30 fewer predicted sags and 39 more false predictions. Depending on the application of the predictor and the selected countermeasures, this may have higher or lower impact on system properties such as availability or the total cost of ownership. The effect on availability will be addressed later in this chapter.

Lastly, we analyze the effect of the sampling period on prediction quality. The evaluation is performed for different lead times (presented with different lines) and, for each lead time the maximum size of the prediction window is taken. Results are presented in Figure 7.10.

A general trend is that performance decreases with increasing sampling period. However, a few exceptions from this trend occur. On average, the best performance is obtained not when the sampling frequency is maximum (period of 20ms) but for the next sampling period value (40 ms). As in the previous cases this may be explained with overfitting. Namely, higher sampling frequency means that more data are taken into consideration and these data are also correlated in time. With decreasing sampling frequency (increasing the period) this correlation in time is decreased as some data are not taken into consideration.

Similarly, performance improvement as lead time increases to 30 cycles may be explained with overfitting. Namely, we have to keep in mind that, in this case, prediction window also changes with lead time so that it equals 90 when lead

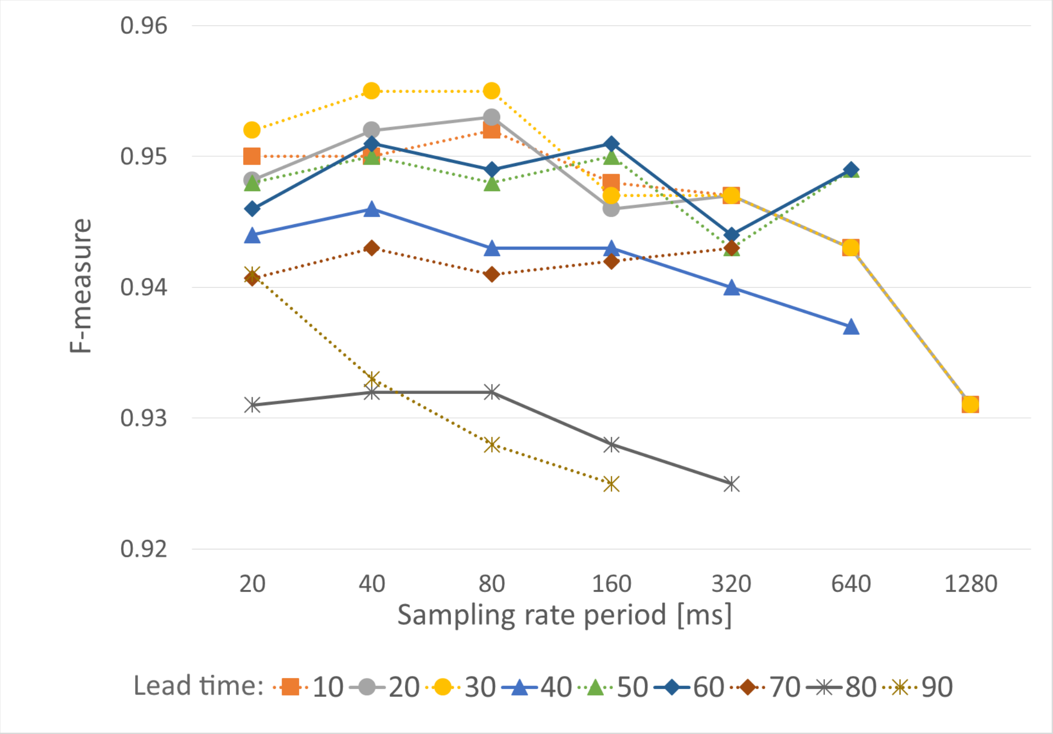


Figure 7.10. The effect of sampling period on performance of prediction with different lead times.

time is 10, 80 when lead time is 20, and 70 when lead time is 30 cycles. Again, more data may introduce more correlation and noise that decrease performances.

In general, having in mind that machine learning algorithms are based on statistics, we may also not expect fully deterministic results. Thus, our conclusions may be considered as generic only to a limited extent and variations and exceptions may occur. Hence, just as in the case of the sampling period, when the goal is to maximize performance, one should first evaluate it with the minimum sampling period (maximum frequency) but also consider the cases when period is multiplied by two and four.

Finally, at the end of this section, we have to repeat that the impact of prediction, sensitivity on prediction performance and the importance of performance measures such as precision and recall (that reflect the number of true-positive, false-positive and false-negative alarms) depend on the application.

7.8 Analyzing the Effect on Availability

We analyze to what extent, with respect to a purely reactive approach to voltage sag mitigation, availability may be enhanced if sags are predicted and preventively mitigated. In both cases we use an Online Tap Changer (OLTC) for voltage control as, besides shunt capacitors, they are the most common means used in distribution systems for this purpose [14, 58]. In addition, OLTCs are good for our case study as it is relatively easy to understand how penalty and reward, that are introduced in Chapter 4, Subsection 4.1.1, may be calculated to estimate downtime decrease and availability enhancement with a proactive approach.

An OLTC is a mechanism that allows to change a transformer's ratio, while the transformer is carrying a load (while it conducts current) without a need to interrupt electric power delivery. This is used for online voltage control by reacting on detected or anticipated voltage changes (sags or swells). Depending on the direction, changing the tap increases or decreases voltage on the secondary side of the transformer.

The basic principle of an OLTC operation is presented in Figure 7.11 that has been adapted from [58]. Annotations 'P' and 'S' are used to indicate primary and secondary sides of the transformer. A voltage level on the secondary side (U_S) is being constantly monitored and the rms value is compared to the nominal voltage level (U_{SET}). If the difference is higher than the maximum tolerable one (U_{DB}), a tap changer is triggered. The tap change direction depends on if the U_S is higher or lower than the U_{SET}. As OLTCs are still expensive devices, with a limited number of tap changes during lifetime, it is not economically justifiable to react on every voltage sag or swell. As for that, the tap is changed only when a sag lasts sufficiently long as, typically, longer sags are more severe. For that reason a time delay (t_{delay}) is introduced to postpone the tap change with respect to the voltage deviation detection. The exact value of the delay depends on the network structure, distribution of voltage deviations and availability of other voltage control means (e.g. shunt capacitors). For large grids it may be up to a few seconds or even minutes. More details on how to set the time delay may be found in [174].

For the 14-bus network, for which we have developed a predictor, we caused only momentary voltage sags that last up to 30 cycles (0.6s). In Figure 7.12 we show a distribution of the number of sags with respect to their duration.

These sags may be mitigated with an OLTC. We set the OLTC delay to 10 cycles, so that only sags that last more than 10 cycles are mitigated. To simplify, we assume that one tap change is sufficient for voltage control and that, after the initial tap change when the voltage drops below 90% of the nominal value,

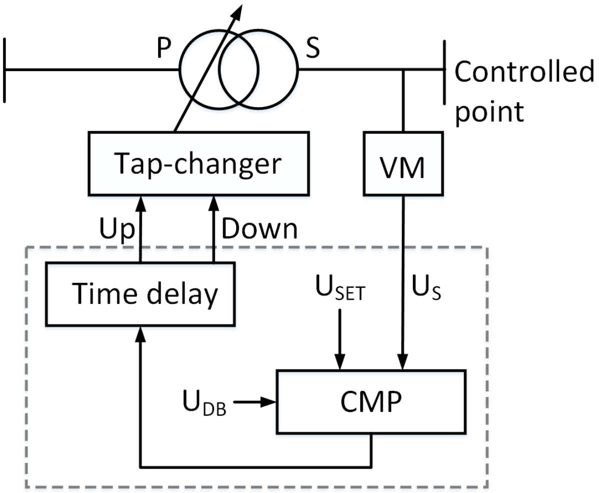


Figure 7.11. Basic principle of an OLTC operation.

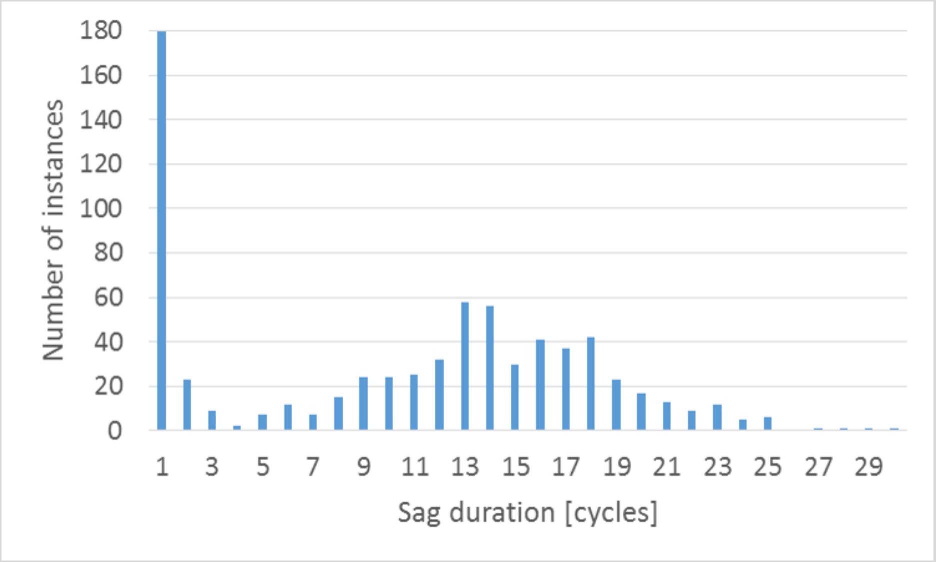


Figure 7.12. Distribution of the number of sags with respect to their duration.

no changes are needed if the voltage drops further. Following the introduced nomenclature, the OLTC is set as follows: $U_{SET} = 1$ p.u., $U_{DB} = 0.1$ p.u., and $t_{delay} = 10$ cycles.

With the selected setting and a purely reactive approach, a transformer's tap is changed if a voltage drop below 0.9 p.u. is detected and if it lasts for at least 10 cycles. We assume that sags are detected with no delay. This is a realistic assumption as with advanced monitoring equipment, such as a PMU, voltage fluctuations may be detected almost instantaneously (sampling period of a PMU is typically 20ms). Also, we may assume that a reaction of an OLTC is instantaneous as well. These assumptions may make the estimation of downtime and availability with individual strategies somewhat less accurate but the comparison between the approaches should not be affected significantly. For the particular case, provided the distribution from Figure 7.12, a delay time of 10 cycles means that 410 out of 713 sags are mitigated, whereas 303 are ignored. In total, the unmitigated sags cause a downtime of 993 cycles (19.83 s). As in a reactive approach the tap is changed with a delay of 10 cycles, users still experience a downtime during the first 10 cycles of a sag even for those sags that are mitigated. This brings additional 4100 cycles (82 s) of downtime. Thus, the total downtime with a reactive approach equals 5093 cycles or 101.86 s.

In a proactive approach, we first anticipate, before the start of a sag, if the sag will last for more than 10 cycles or not. If the prediction is that the sag's duration is more than 10 cycles, an OLTC is immediately triggered upon the sag's detection to avoid additional downtime that is introduced with a delayed tap change. The proactive approach that we propose relies on both, sag prediction and sag detection, and as such may be also considered as a proactive-reactive approach.

For this purpose we first design a predictor that predicts only sags that last more than 10 cycles. Following the previous analysis, we use IBk algorithm and set the lead time to 40 cycles to have sufficient time for proactive reaction. The prediction window size is set to 60 cycles and sampling period is 40 ms. A precision-recall curve of the developed algorithm is presented in Figure 7.13.

A selection of the optimal point on the curve is performed using the A-measure (Equation 4.8 from Chapter 4, Subsection 4.1.1) and following the procedure from 4.2.2. For the convenience, we repeat the equation here as 7.1 . We also remind the reader that a corresponds to the failure rate increase factor due to prediction, whereas c is a proactive action success probability. Unlike in computer systems where a prediction may interfere with the execution of the main process and increase the failure rate, prediction itself cannot affect the voltage level and the number of voltage sags (at least not in the analyzed case). Thus, a equals zero. Also, as a proactive action (tap change) is triggered with the sag detection, just like in the reactive case, when comparing reactive and proactive approach we may assume that the success probability equals one as the same action is used in both approaches.

$$A_{measure} = R * \left(reward * c - \left((1-c) + \frac{1-P}{P}\right) * penalty\right) \tag{7.1}$$

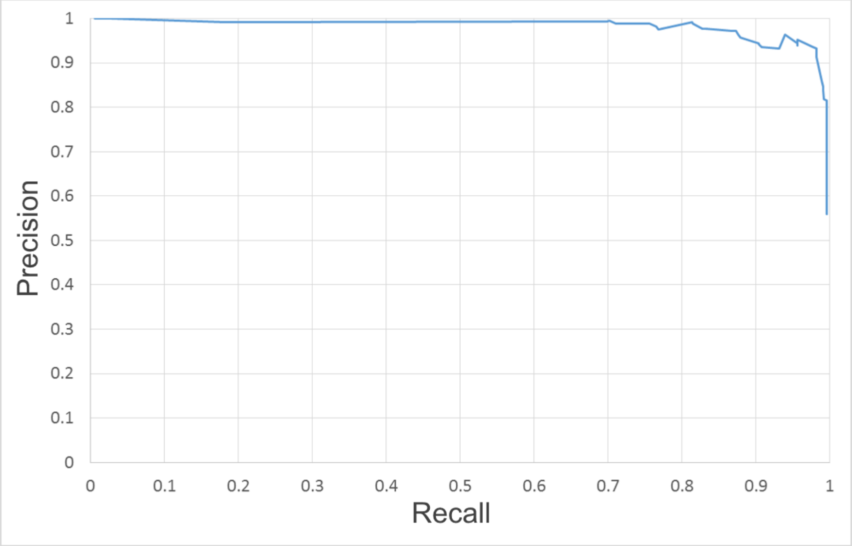


Figure 7.13. A precision-recall curve of the predictor of sags that last more than 10 cycles.

Reward is a downtime decrease in the case of a correct prediction and a successful proactive action. This corresponds to the OLTC time delay that, in our case, is 10 cycles. Finally, penalty corresponds to introduced downtime when the action was needless or unsuccessful. Again, as actions are triggered only upon a detection, there cannot be a case where a tap is changed only because of a sag prediction and where a sag has not been detected, and thus the penalty equals zero. It is important to point out that it may still happen that a sag that lasts less than the OLTC delay time is predicted as a longer sag and that the tap is, unnecessary, changed when such a sag is detected. This case will not introduce additional downtime (in fact it will even contribute to the total downtime decrease) but it is not favorable from the economical point of view as the OLTC should be triggered only for longer sags. However, the effect of prediction on the operational cost is considered to be out of the scope of this work.

As penalty is zero, the procedure for finding an optimal PR point is simplified as only recall has to be maximized. The maximum recall value for the designed predictor equals 0.996, whereas precision for this point equals 0.673. Under the assumptions stated before for the case of the reactive approach, the total downtime with the selected point equals 994.64 cycles (19.89 s). Sags that last more than 10 cycles, and that are mitigated, contribute with only 16.4 cycles of downtime (0.33 s).

Estimating the effect on availability first requires to estimate MTTF and MTTR of voltage sags. According to industrial reports, such as [175] and statistical analysis as [176], a typical number of voltage sags in distribution networks is from about 20 to 50 per year. However, provided the challenges that come with renewable generation and load increase, we may only expect this number to increase in ADNs. As for that, we evaluate availability for the range of 20 to 500 sags per year. This corresponds to a range of MTTF from 18.25 days to 17.5 hours.

Following the results of our simulation, and assumed OLTC delay of 10 cycles, MTTR for the reactive case equals 7.14 cycles (0.143 s), whereas MTTR for the proactive case is 1.39 cycles (0.028 s). Availability with reactive and with proactive policy for different number of sags per year and different OLTC delay time is presented in Figure 7.14.

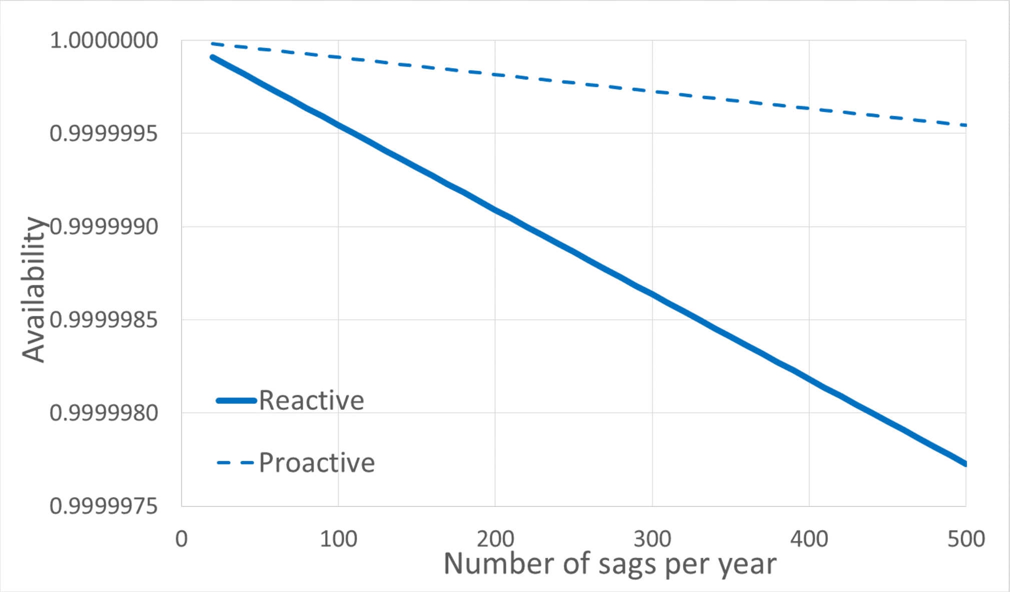


Figure 7.14. Availability for OLTC delay of 10 cycles and different number of sags per year.

The extent to what availability is enhanced strongly depends on the number of sags per year. When the number of sags per year is 50, availability with both approaches is very high. With the reactive approach it is 0.99999977 and with the proactive one it is 0.99999995. This corresponds to annual downtime of only about 300 ms in the reactive and 60ms in the proactive case. However, as the number of sags increases, their affect as well as the difference between the two strategies, become more evident. When this number reaches 500, availability improvement is almost by the order of magnitude, namely from about five and a half nines in the reactive, to six and a half nines in the proactive case. This corresponds to downtime of 3 s in the case of the reactive and 0.6 s in the case of the proactive approach.

However, we must keep in mind that the analyzed network is relatively small and that the sags that we were able to simulate in such network are relatively short. In a larger network where faults are more frequent and may affect larger number of customers the benefit of a proactive approach will be even higher.

We also analyze how availability changes in a larger network where the OLTC delay time is also longer and goes up to 100000 cycles (3.33 min). This corresponds to cases of also longer momentary and temporary sags. We assume that the distribution is similar to the one from Figure 7.12 in a sense of the number of sags that last less than the delay time and those that last more is similar to the case that we have simulated. The results are presented in Figure 7.15. Numbers in brackets in the figure legend indicate the size of the delay time in cycles of 20 ms (10 cycles is 0.2 s and 10.000 cycles is 200 s or about 3.3 min).

When the OLTC delay is set to 1000 cycles (20 s) and the number of sags per year is 50 (that is already a realistic scenario in today's distribution grids that do not incorporate renewable generation), availability with the reactive approach is 0.99997727, whereas availability with the proactive one is 0.99999544. Again, the improvement is almost by the order of magnitude (from four and a half to five and a half nines). This corresponds to a downtime per year of almost 30 s with the reactive and almost 6 s with the proactive approach.

This difference is even higher when the number of sags per year and OLTC delay increase. In Table 7.9 we compare availability and downtime per year for the selected points from Figure 7.15.

It may be observed that, regardless of the number of sags and the OLTC delay, proactive approach always improves availability by, approximately, an order of magnitude when compared to the reactive one.

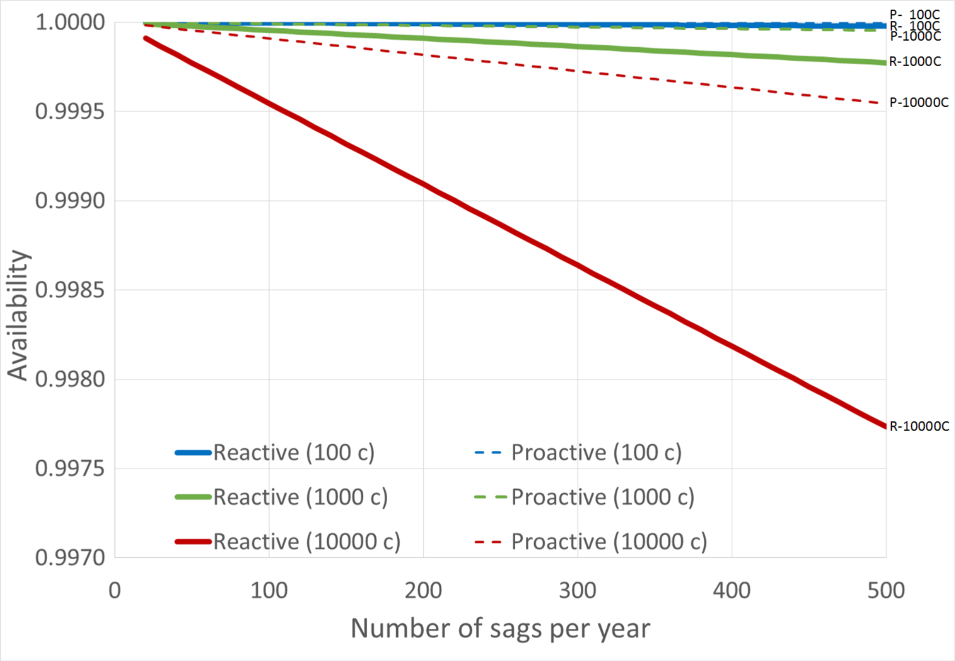


Figure 7.15. Availability for a range of values for the OLTC delay and a different number of sags per year.

Table 7.9. Availability and downtime per year for the selected values of the number of sags per year and OLTC delays.

Sags per year	OLTC delay [s]	Availability (reactive)	Availability (proactive)	Downtime per year (reactive) [s]	Downtime per year (proactive) [s]
50	2	0.9999977	0.9999995	2.99	0.60
50	20	0.9999773	0.9999954	29.86	6.00
50	200	0.9997728	0.9999544	298.55	59.98
500	2	0.9999773	0.9999954	29.86	6.00
500	20	0.9997728	0.9999544	298.55	59.98
500	200	0.9977325	0.9995437	2979.44	599.54

Chapter 8

Concluding Remarks, Main Contributions and Future Work

With the evolution towards Smart Grids, the power system and distribution grids in particular, are facing numerous challenges. These include increasing power demand, rising penetration of intermittent renewable energy resources, structural changes and growing complexity of the grid, new types and higher frequency of faults and disturbances, as well as a demand for more flexible and even more efficient grid management. In the light of these challenges, fostering dependability of the electric power delivery service has become ever-more difficult. This requires novel control and management approaches to attain the current level and to further enhance dependability of the power system as a critical infrastructure. In this dissertation, a proactive control approach, based on short-term prediction of disturbances and their mitigation, for improved availability of Smart Grids has been proposed. The main focus of the work is on prediction of disturbances in Active Distribution Networks.

8.1 Summary and Conclusions

Smart Grid dependability has been comprehensively analyzed by taking an integrated approach to address dependability of a Smart Grid as a whole, considering its cyber and physical infrastructures as well as their interdependencies. Generally, different approaches to dependability in electric power and computer systems communities pose a danger of compromising dependability of Smart Grids. For that reason, dependability definitions, as used in the two communities, have been reviewed and compared. As a result, a unique set of Smart Grid dependability attributes have been identified and their definitions and figures of merit have been proposed. A large set of major blackouts has been analyzed to identify the most common root causes of power system disturbances and to conduct their classification in a form of a developed taxonomy of Smart Grid faults. The proposed set of definitions and the taxonomy contribute to better communication between the two communities and give a unique view on the Smart Grid dependability. In that way, they also help in identifying future research directions in dependability and security of Smart Grids. Moreover, the taxonomy helps to better understand the overall dependability of Smart Grids as it identifies major threats that may compromise it. This is particularly important in the light of ongoing grid's digitalization as some of the threats that we have identified and classified, or their combinations, may become more frequent in the near future as the grid evolves. Being aware of these threats is a first step towards developing methods, such as proactive ones, for their prevention and mitigation before they compromise grid's dependability.

To better understand how proactive management affects availability and what the required criteria for the quality of prediction are, a model of proactive (predictive) disturbance management has been developed. The model is generic and it may be applied in different fields and for modeling the effect of the proactive approach on different system properties but availability remains our main focus. Using the model, availability of a system with a proactive approach may be quantified and then compared to the one with a reactive approach to answer the fundamental question whether to manage a system reactively or proactively. To accomplish this we have extended the availability equation with parameters that characterize reactive and proactive management approaches. A proactive approach is modeled with precision and recall, penalty and reward, proactive action success probability and potential failure rate increase due to the prediction load. The model and the equation are intuitive, and easy to understand and apply. From the equation we have derived A-measure that may be used to find the optimal trade-off between the precision and recall. As data on disturbances and availability in power systems are still limited, the model has been evaluated on a case from computer systems, namely a virtualized server system. We conclude that, for realistic system parameters, availability may be improved when failure prediction quality is sufficiently high. The break-even point, when availability of the system is the same as with or without predictions, depends on precision only but the total improvement of availability is strongly affected by recall. In fact, we conclude that the sensitivity of a system's availability with respect to recall is comparable to the one with respect to mean-time-to-failure. Therefore, it may be more cost-effective to invest in high-quality prediction algorithm than in acquiring costly components with lower failure rates to increase availability. We also provide guidelines for applying the method to other types of systems. Using our approach developers and system designers may evaluate the benefits of using a proactive management before investing in its implementation. We focus on availability but the approach may also be extended to evaluate the affect of prediction on other system properties including the total cost of ownership.

A methodology and methods for proactive control in Smart Grid, in a sense of proactive management of disturbances for improved availability is then proposed. They are based on data analytics and inspired by well-established solutions employed in computer engineering. The concept relies on statistical analysis of historic pre-event data coupled with online monitoring of the most indicative features for prediction of near-future disturbances and their mitigation. The approach may also be applied to address different types of disturbances or undesirable changes, or to enhance other dependability properties apart from availability. The methodology and methods for the design of a disturbance predictor are described in detail. Moreover, the proposed methodology is fully modular and the implementation of the modules is not limited to the proposed methods. When evaluating how a specific combination of prediction quality and mitigation method affects availability, the previously described availability equation and the A-measure may be used.

The main obstacle to the design of a disturbance predictor and application of a proactive control approach is a lack of relevant disturbance-related data. To overcome this obstacle, a framework for synthetizing power systems disturbance-related data based on simulation and fault injection has been developed. The framework performs simulations using PSAT power systems simulator. We have developed a mechanism and provided an interface to inject and to clear faults at predefined times. To reflect dynamic aspects of a system, we have also implemented a mechanism to vary generation and load at runtime. Finally, we have implemented a mechanism for detection of voltage sags. Being an extension of an existing tool, the framework supports a wide set of model formats. Most importantly, it allows to record a set of selected system features with a predefined sampling rate. Thus, it may be used to simulate behavior of a system in presence of faults that may cause disturbance and to log the data that also include information on observed disturbances. This capability clearly contributes to Smart Grid dependability enhancement by providing an instrument to facilitate investigation of disturbances. We use the framework to synthesize disturbance-related data for design and evaluation of a sags predictor but it may also be employed when designing a new generation of disturbance detectors and evaluating sensitivity on generation and load variations as well as monitoring sampling rate. It may be further enhanced to support the analysis of grid's robustness to variations in (distributed) generation and its capability to accept additional renewable generators. Finally, it may be used to evaluate how effective different protection mechanisms and management methods are with respect to different types of faults that we have classified in the scope of our taxonomy of Smart Grid faults.

As a part of the case study, following the proposed methodology, we have implemented a predictor for voltage sags in an Active Distribution Network. To illustrate our methodology, voltage sags have been selected as they are marked as one of the most frequent and severe disturbances in distribution networks. We have simulated a behavior of a 14-bus ADN in the presence of balanced short-circuit faults and varying load and generation that cause voltage sags. Machine learning classification algorithms are applied when designing a predictor. The quality of prediction is evaluated with F-measure as well as with precision and recall using ten-fold cross validation. We have also evaluated how lead time, sampling period and prediction window size affect prediction quality. We conclude that, in the simulated grid, momentary voltage sags that last up to 0.6 s may be predicted up to 1.8 s in advance with a very good quality of prediction (precision of 0.909, and recall of 0.968). As a part of predictor design, prediction accuracy with individual features and their combinations have been evaluated. We observe that the best prediction quality may be obtained when combining four most indicative features, and that phase angle and active power values on the bus where the fault is injected are the most indicative for sag prediction. Even though this may have been expected by the domain experts, it is an important conclusion as it helps to decrease the amount of data that need to be monitored, collected and propagated in realtime. Moreover, we conclude that the best results are obtained with the k-nearest neighbors machine-learning algorithm with a combination of phase angles on the fault-injection (fault-occurrence) bus and the buses that are connected to it and that host large loads of renewable generation. We also observe that longer lead time and shorter prediction window typically decrease predictor's performance and that sampling rate has a strong impact on the quality of prediction, especially when the lead time is long. Even though it is difficult to provide general conclusions based on a relatively small grid that has been modeled and that it is unrealistic to expect deterministic results when statistical methods, such as machine learning, are used we may still derive general guidelines. First, a disturbance predictor design depends on the fault location and it may not be practical to invest into a design of a generic predictor that does not take into account a type and a location of a fault that causes the disturbance. Also, such a predictor would not be very useful as it may also require to perform fault localization before its mitigation. Second, prediction performance may be affected by ovefitting and feature correlation. For that reason it is not always the case that shorter lead time, large prediction window and higher sampling rate will improve prediction performance as this also increases the amount of data. Thus, designers should also evaluate prediction performance when these parameters vary, in the limited scope, from their values that would be expected to maximize performance (e.g. consider decreasing sampling rate by 20% from the maximum). Third, in practical applications it would make sense to develop a predictor considering parts of the grid that have had the most frequent failures in the past. Finally, using the developed model of proactive control and A-measure, we have optimized prediction and evaluated how it affects availability. For this purpose we have modeled sag mitigation with an OLTC. In our conclusion, with prediction quality as the one that we were able to obtain and with proactive mitigation of voltage sags, availability may be improved approximately by an order of magnitude when compared to reactive approach. Obviously, the effect of proactive approach is more evident in cases when sags are more frequent.

The proposed proactive approach that includes the availability equation, the fault-injection and simulation framework the methodology and the disturbance predictor may be applied to other types of disturbances except for voltage sags.

8.2 Overview of Main Contributions

The main contributions of the work presented in this manuscript may be summarized as follows:

- Smart Grid dependability attributes have been proposed and a taxonomy of Smart Grid faults has been developed.
- A model of proactive disturbance management has been developed, supported by a methodology and a metric for optimizing failure prediction for improving availability. Guidelines for optimizing availability with the proposed model and the metrics have been defined.
- A methodology for proactive management of disturbances in Smart Grid has been developed and appropriate methods have been implemented.
- Simulation and fault-injection framework for analyzing behavior of power grids in the presence of faults has been designed and implemented. The framework focuses on Active Distribution Networks and is mainly intended for generation of disturbance-related data for the design of predictors but may also be used for other types of disturbance analysis.

- To illustrate the methodology, a predictor for voltage sags caused by balanced short-circuit faults and fluctuating load and distributed generation has been developed as the case study. The effect of sag prediction and mitigation on availability has been analyzed. Also, we have investigated how monitoring sampling rate, prediction window size and lead time affect the quality of prediction.

8.3 Future Work

The presented work gives solid foundations to further develop and implement proactive control approaches in Smart Grids. However, improvements and additional verification may be needed before putting these solutions into practice for a selected grid. This should also consider characteristics of the grid and constraints most critical points and types of disturbances, and available mitigation means. Moreover, and especially having in mind the novelty of the proposed concept and the ongoing grid modernization, the work may be further extended in different research directions as it has raised additional questions in the sphere of Smart Grid dependability and the use of proactive methods for its management. The continuation of this work may include but is not limited to the following major topics that are foreseen by the author:

- The developed model of proactive control may be enhanced to evaluate the effect of prediction and proactive mitigation on other system properties such as security, maintainability, operational cost and life-cycle management. In the scope of such extension, mitigation means and metrics for prediction optimization with respect to the properties should be identified or developed. These metrics may be developed in a similar manner as those that we have proposed for optimizing availability. In the same spirit, the proposed proactive control methodology may be adapted for the minimization of the operational cost. Such a work could become a part of a wider research on minimizing the total cost of ownership of ADNs and Smart Grid in general. Moreover, a similar approach may be applied for addressing dependability of other types of systems. In fact, in [134] we have already proposed how it may be adopted for improving availability of UPS'es.

- With its digitalization, the grid is more exposed to cyber-attacks and security of the Smart Grid, as a critical infrastructure, is becoming one of the major concerns. In fact, cyber-attacks have already been identified as causes of a number of blackouts and we may only expect that they become more frequent. Early-warning systems, that are essentially based on a similar approach as the one for prediction of disturbances, may be developed. Moreover, as cyber-attacks in power grids are not yet that frequent, there may be a need to develop a similar framework to simulate security aspects of Smart Grids. Finally, it would be worth combining the two approaches and evaluating to what extent cyber-attacks may affect dependability, and availability in particular, and how early-detection methods may help to decrease these effects.

- The developed simulation and fault-injection framework has a potential for extension in numerous directions, especially that the fault-injection concept was not commonly used in power systems simulation until recently. Moreover, the framework was built as user-friendly in a sense of a simplified environment suitable for engineers with limited background in power engineering. This is an important property as it helps computer scientists to get more involved in Smart Grid development. The current implementation includes limited choice for fault-injection that was sufficient for the case-study data generation. A possibility to inject other types of faults could also be included. In fact, expanding the simulator and adding a possibility to inject cyber-attacks, as proposed in [177] for computer networks, would be very beneficial for evaluating security and dependability of Smart Grids in an integrated fashion. It is also worth considering to use the tool for evaluating robustness of the grid against specific types of faults that may not be present at the moment but that are expected in the future as well as the increased penetration of renewable resources and higher fluctuations of consumption. In this way, the framework would become a comprehensive environment for analyzing Smart Grid faults and disturbances. Moreover, it could be coupled with the predictor so that predictions are used at simulation runtime to trigger proactive actions. Thereby, the effect of the proactive control on availability could be analyzed through simulations instead of using the model.

- Finally, once real-life data become available, the approach could be evaluated with that data. Moreover, the implementation in a real grid should consider other aspects such as time needed to transfer the data from different parts of the grid as well as time needed for online processing. Both, communication and processing must be very efficient processes considering that disturbances may, according to our analysis, be predicted only with a very short lead-time. Different architectures for implementation should also be considered as, for example, centralized and decentralized.

In conclusion, electric power grid is a critical infrastructure whose dependability should never be compromised. With its evolution, novel methods are needed to ensure and to further endure its dependability. Even though the proposed proactive management approach gives promising results for improving availability, it must be incorporated into a wider strategy of Smart Grid management that combines different approaches to ensure grid's dependability and security in the light of an ever-increasing number and types of threats. In fact, as Smart Grid is truly a system-of-systems, it is necessary to involve experts and stakeholders from various domains including power and computer engineers as well as experts in law, business and marketing, and to combine different methods in Smart Grid development as only with a unified and an interdisciplinary approach we can make sure to keep it trustworthy, secure and efficient.

Bibliography

[1] D. Von Dollen, "Report to NIST on the smart grid interoperability standards roadmap," Electric Power Research Institute (EPRI) and National Institute of Standards and Technology, Tech. Rep., May 2009. [Online] Available: https://www.smartgrid.gov/document/report_nist_smart_grid_interoperability_standards_roadmap

[2] "SmartGrids: Strategic deployment document for Europe's electricity networks of the future," European Technology Platform, Tech. Rep., Apr. 2010. [Online] Available: http://www.smartgrids.eu/documents/SmartGrids_SDD_FINAL_APRIL2010.pdf

[3] S. Rohjans, M. Uslar, R. Bleiker, J. Gonzalez, M. Specht, T. Suding, and T. Weidelt, "Survey of smart grid standardization studies and recommendations," in *1st IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, USA, Oct. 2010

[4] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - the new and improved power grid: A survey," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 944–980, 4th quarter 2012

[5] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 57–64, Jun. 2010

[6] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, Jan. 2010

[7] E. Santacana, B. Husain, F. Pinnekamp, P. Halvarsson, G. Rackliffe, L. Tang, and X. Feng, "The next level of evolution," in *ABB Review: Smart Grids*, P. Terwiesch, Ed., pp. 10 – 15. ABB Corporation, 2010

[8] J. Giri, "Proactive management of the future grid," *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 43–52, Jun. 2015

[9] E. Marris, "Energy: Upgrading the grid," *Nature News*, 2008

[10] A. von Meier, *Electric Power Systems: A Conceptual Introduction*. Wiley-IEEE Press, 2006

[11] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011

[12] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug. 2011

[13] G. Brauner, W. D'Haeseleer, W. Gehrer, W. Glaunsinger, T. Krause, H. Kaul, M. Kleimaier, W. Kling, H. Prasser, I. Pyc, W. Schröppel, and W. Skomudek, "Electrical power vision 2040 for Europe," Eurelectric, Tech. Rep., 2013. [Online] Available: http://www.eurel.org/home/TaskForces/Documents/EUREL-PV2040-Full_Version_Web.pdf

[14] F. Viawan, "Voltage control and voltage stability of power distribution systems in the presence of distributed generation," Ph.D. Thesis, Division of Electric Power Engineering, Department of Energy and Environment, Chalmers University of Technology, Göteborg, SE, 2008

[15] G. Strbac and N. Hatziargyriou, "The need for a fundamental review of electricity networks reliability standards," European Technology Platform on Smart Grids: Smartgrids security and resilience task force, Tech. Rep., 2016. [Online] Available: http://www.smartgrids.eu/documents/ETP_SG_Future_Network_Reliability_Standards_2016.pdf

[16] D. Watts, "Security and vulnerability in electric power systems," in *35th North American Power Symposium (NAPS)*, Denver, CO, USA, Sep. 2003

[17] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, "Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence," *ACM Communications*, vol. 55, no. 4, pp. 86–97, Apr. 2012

[18] E. Sortomme, M. M. Hindi, S. D. J. MacPherson, and S. S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 198–205, Mar. 2011

[19] W. Y. Chiu, H. Sun, and H. V. Poor, “Energy imbalance management using a robust pricing scheme,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 896–904, Jun. 2013

[20] D. Pudjianto, C. Ramsay, and G. Strbac, “Virtual power plant and system integration of distributed energy resources,” *IET Renewable Power Generation*, vol. 1, no. 1, pp. 10–16, Mar. 2007

[21] S. Lukovic, I. Kaitovic, M. Mura, and U. Bondi, “Virtual power plant as a bridge between distributed energy resources and smart grid,” in *43rd IEEE Hawaii International Conference on System Sciences (HICSS)*, Honolulu, HI, USA, Jan. 2010

[22] I. Kaitovic and S. Lukovic, “Adoption of model-driven methodology to aggregations design in smart grid,” in *9th IEEE International Conference on Industrial Informatics (INDIN)*, Lisbon, PT, Jul. 2011

[23] G. Joos, B. T. Ooi, D. McGillis, F. D. Galiana, and R. Marceau, “The potential of distributed generation to provide ancillary services,” *IEEE Power Engineering Society Summer Meeting*, vol. 3, pp. 1762–1767 vol. 3, Jun. 2000

[24] W. Kempton and J. Tomic, “Vehicle-to-grid power fundamentals: Calculating capacity and net revenue,” *Journal of Power Sources*, vol. 144, no. 1, pp. 268 – 279, Jun. 2005

[25] R. A. Len, V. Vittal, and G. Manimaran, “Application of sensor network for secure electric energy infrastructure,” *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1021–1028, Apr. 2007

[26] “IEEE standard for synchrophasor measurements for power systems,” *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, Dec. 2011

[27] M. Pignati, M. Popovic, S. Barreto Andrade, R. Cherkaoui, D. Flores, J.-Y. Le Boudec, M. M. Maaz, M. Paolone, P. Romano, S. Sarri, T. T. Tesfay, D.-C. Tomozei, and L. Zanni, “Real-Time State Estimation of the EPFL-Campus Medium-Voltage Grid by Using PMUs,” in *6th Conference on Innovative Smart Grid Technologies (ISGT)*, Washington, DC, USA, Feb. 2015

[28] “IEEE guide for synchronization, calibration, testing, and installation of phasor measurement units (PMUs) for power system protection and control,” *IEEE Std C37.242-2013*, pp. 1–107, March 2013

[29] S. Sarri, M. Paolone, R. Cherkaoui, A. Borghetti, F. Napolitano, and C. A. Nucci, "State estimation of active distribution networks: Comparison between WLS and iterated Kalman-filter algorithm integrating PMUs," in *3rd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, Copenhagen, DK, Oct. 2012

[30] P. Romano, "DFT-based synchrophasor estimation algorithms and their integration in advanced phasor measurement units for the real-time monitoring of active distribution networks," Ph.D. Thesis, La Faculté des Sciences et Techniaues de l'ingénieur, École Polytechniaue Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2016

[31] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 5–20, Jan. 2013

[32] Accenture Analytics, "Achieving high performance in smart grid data management: Making sense of the data deluge," Dec. 2010. [Online] Available: https://www.smartgrid.gov/files/Achieving_High_Performance_in_Smart_Grid_Data_Management_201012.pdf

[33] C. Ivanov, T. Saxton, J. Waight, M. Monti, and G. Robinson, "Prescription for interoperability: Power system challenges and requirements for interoperable solutions," *IEEE Power and Energy Magazine*, vol. 14, no. 1, pp. 30–39, Jan. 2016

[34] Z. Zhong, C. Xu, B. J. Billian, L. Zhang, S. J. S. Tsai, R. W. Conners, V. A. Centeno, A. G. Phadke, and Y. Liu, "Power system frequency monitoring network (FNET) implementation," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1914–1921, Nov. 2005

[35] "Managing big data for Smart Grids and Smart Meters," Software Group, IBM Corporation, Tech. Rep., May 2012

[36] P. Romano, M. Pignati, and M. Paolone, "Integration of an IEEE Std. c37.118 compliant PMU into a real-time simulator," in *IEEE PowerTech*, Eindhoven, NL, Jun. 2015

[37] S. Cole and R. Belmans, "MatDyn, a new Matlab based toolbox for power system dynamic simulation," in *IEEE Power and Energy Society General Meeting*, Detroit, MI, USA, Jul. 2011

[38] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011

[39] F. Milano, “An open source power system analysis toolbox,” *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1199–1206, Aug. 2005

[40] A. Z. Faza, S. Sedigh, and B. M. McMillin, “Reliability analysis for the advanced electric power grid: From cyber control and communication to physical manifestations of failure,” in *28th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Hamburg, DE, Sep. 2009

[41] X. She, A. Q. Huang, and R. Burgos, “Review of solid-state transformer technologies and their application in power distribution systems,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 1, no. 3, pp. 186–198, Sep. 2013

[42] N. Hatziargyriou, *Operation of Multi-Microgrids*. Wiley-IEEE Press, 2014

[43] D. Trebolle, P. Hallberg, G. Lorenz, P. Mandatova, and J. T. Guijarro, “Active distribution system management,” in *22nd International Conference and Exhibition on Electricity Distribution (CIRED)*, Stockholm, SE, Jun. 2013

[44] P. Hallberg, et. al, “Active distribution system management: A key tool for the smooth integration of distributed generation,” Eurelectric, Tech. Rep., 2013. [Online] Available: http://www.eurelectric.org/media/74356/asm_full_report_discussion_paper_final-2013-030-0117-01-e.pdf

[45] C. D’Adamo, S. Jupe, and C. Abbey, “Global survey on planning and operation of active distribution networks - Update of CIGRE C6.11 working group activities,” in *20th International Conference and Exhibition on Electricity Distribution (CIRED)*, Prague, CZ, Jul. 2009

[46] M. Paolone, A. Borghetti, and C. A. Nucci, “A synchrophasor estimation algorithm for the monitoring of Active Distribution Networks in steady state and transient conditions,” in *17th Power Systems Computation Conference (PSCC)*, Stockholm, SE, Aug. 2011

[47] T. Sansawatt, J. O’Donnell, L. F. Ochoa, and G. P. Harrison, “Decentralised voltage control for active distribution networks,” in *44th International Universities Power Engineering Conference (UPEC)*, Glasgow, UK, Sep. 2009

[48] M. Bahramipanah, “Advanced Control of Active Distribution Networks Integrating Dispersed Energy Storage Systems,” Ph.D. Thesis, Faculté Sciences et Technique de l’Ingénieur, École Polytechnique Fédérale de Lausanne, Lausanne, Jun. 2016

[49] “The Smart Grid: An Introduction,” Litos Strategic Communication for U.S. Department of Energy, Tech. Rep., 2009. [Online] Available: https://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages%281%29.pdf

[50] R. Arghandeh, A. von Meier, L. Mehrmanesh, and L. Mili, “On the definition of cyber-physical resilience in power systems,” *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 1060–1069, May 2016

[51] J.-C. Laprie, K. Kanoun, and M. Kaâniche, “Modelling interdependencies between the electricity and information infrastructures,” *Computer Safety, Reliability, and Security*, pp. 54–67, Sep. 2007

[52] S. Blumsack and A. Fernandez, “Ready or not, here comes the smart grid!” *Energy*, vol. 37, no. 1, pp. 61 – 68, Jan. 2012

[53] Z. Dong, P. Zhang *et al.*, *Emerging techniques in power system analysis*. Springer, 2010

[54] P. Esteves-Verissimo, M. Völp, J. Decouchant, V. Rahli, and F. Rocha, “Meeting the challenges of critical and extreme dependability and security,” in *IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*, Christchurch, NZ, Jan. 2017

[55] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on the smart grid,” *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011

[56] T. Escritt, “Reuters news: Power returns to Amsterdam after outage hits a million homes,” Mar. 2016. [Online] Available: http://www.reuters.com/article/us-dutch-power-outages-idUSKBN0MN0UJ20150327

[57] I. Kaitovic, S. Lukovic, and M. Malek, “Unifying dependability of critical infrastructures: Electric power system and ICT: Concepts, figures of merit and taxonomy,” in *21st IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, Zhangjiajie, PRC, Nov. 2015

[58] F. A. Viawan and D. Karlsson, “Voltage and reactive power control in systems with synchronous machine-based distributed generation,” *IEEE Transactions on Power Delivery*, vol. 23, no. 2, pp. 1079–1087, Apr. 2008

[59] *IEEE1159-1995 - IEEE Recommended Practice for Monitoring Electric Power Quality*, IEEE Std., 2011

[60] S. Koch, F. S. Barcenas, and G. Andersson, “Using controllable thermal household appliances for wind forecast error reduction,” *IFAC Proceedings Volumes*, vol. 43, no. 1, pp. 261–266, Mar. 2010

[61] K. Christakou, D. C. Tomozei, M. Bahramipanah, J. Y. L. Boudec, and M. Paolone, “Primary voltage control in active distribution networks via broadcast signals: The case of distributed storage,” *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2314–2325, Sep. 2014

[62] A. Marinakis, C. Franke, and M. Larsson, “Evolution strategies-tuned support vector machine-based classification of inter-area oscillations,” in *18th IEEE Power Systems Computation Conference (PSCC)*, Wroclaw, PL, Aug. 2014

[63] T. Weckesser, H. Jóhannsson, and T. V. Cutsem, “Early prediction of transient voltage sags caused by rotor swings,” in *IEEE PES General Meeting (Conference Exposition)*, Washington, DC, USA, Jul. 2014

[64] F. Salfner and M. Malek, “Proactive fault handling for system availability enhancement,” in *19th IEEE International Parallel and Distributed Processing Symposium*, Denver, CO, USA, Apr. 2005

[65] C. Rudin, D. Waltz, R. Anderson, A. Boulanger, A. Salleb-Aouissi, M. Chow, H. Dutta, P. Gross, B. Huang, and S. Ierome, “Machine learning for the New York city power grid,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 328–345, Feb. 2012

[66] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation,” *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, Feb. 2001

[67] G. Hoffman and M. Malek, “Call availability prediction in a telecommunication system: A data driven empirical approach,” in *25th IEEE Symposium on Reliable Distributed Systems (SRDS)*, Leeds, UK, Oct. 2006

[68] Z. Lan and Y. Li, "Adaptive fault management of parallel applications for high-performance computing," *IEEE Transactions on Computers*, vol. 57, no. 12, pp. 1647–1660, Dec. 2008

[69] IBM XIV storage system. [Online] Available: www.ibm.com/systems/storage/disk/xiv/

[70] IBM predictive maintenance. [Online] Available: http://www-03.ibm.com/software/products/en/predictive-maintenance

[71] HP Backup Navigator. [Online] Available: https://saas.hpe.com/en-us/software/backup-monitoring-reporting-software

[72] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Computing Surveys*, vol. 42, no. 3, pp. 10:1–10:42, Mar. 2010

[73] I. Irrera and M. Vieira, "A practical approach for generating failure data for assessing and comparing failure prediction algorithms," in *20th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, Singapore, SG, Nov. 2014

[74] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, Jan. 2004

[75] I. P. Egwutuoha, S. Chen, D. Levy, B. Selic, and R. Calvo, "A proactive fault tolerance approach to high performance computing (HPC) in the cloud," in *2nd International Conference on Cloud and Green Computing*, Xiangtan, CN, Nov. 2012

[76] W. Zhao and H. Zhang, "Proactive service migration for long-running byzantine fault-tolerant systems," *IET Software*, vol. 3, no. 2, pp. 154–164, Apr. 2009

[77] A. B. Nagarajan, F. Mueller, C. Engelmann, and S. L. Scott, "Proactive fault tolerance for HPC with Xen virtualization," in *21st Annual International Conference on Supercomputing*, New York, NY, USA, Jun. 2007

[78] A. M. Johnson, Jr. and M. Malek, "Survey of software tools for evaluating reliability, availability, and serviceability," *ACM Computing Surveys*, vol. 20, no. 4, pp. 227–269, Dec. 1988

[79] M. Malek, G. A. Hoffmann, N. Milanovic, S. Bruening, R. Meyer, and B. Milic, “Methoden und werkzeuge zur verfügbarkeitsermittlung,” Institut für Informatik, Rechnerorganisation und Kommunikation, Humboldt-Universität zu Berlin, 2007. [Online] Available: http://edoc.hu-berlin.de/series/informatik-berichte/219/PDF/219.pdf

[80] K. S. Trivedi and R. Sahner, “Sharpe at the age of twenty two,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 52–57, Mar. 2009

[81] G. Ciardo, J. Muppala, and K. Trivedi, “SPNP: Stochastic Petri net package,” in *3rd International Workshop on Petri Nets and Performance Models (PNPM)*, Washington, DC, USA, Dec. 1989

[82] S. Chiaradonna, P. Lollini, and F. D. Giandomenico, “On a modeling framework for the analysis of interdependencies in electric power systems,” in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Edinburgh, UK, Jun. 2007

[83] M. N. Albasrawi, N. Jarus, K. A. Joshi, and S. S. Sarvestani, “Analysis of reliability and resilience for smart grids,” in *38th IEEE Annual Computer Software and Applications Conference*, Vasteras, SE, Jul. 2014

[84] I. Dobson and B. A. Carreras, “Risk analysis of critical loading and blackouts with cascading events,” Consortium for Electric Reliability Tech. Solutions (CERTS), Tech. Rep., Jan. 2005. [Online] Available: https://certs.lbl.gov/sites/all/files/risk-analysis-loading-blackouts.pdf

[85] “IEEE guide for electric power distribution reliability indices - redline,” *IEEE Std 1366-2012 (Revision of IEEE Std 1366-2003)*, pp. 1–92, May 2012

[86] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. V. Cutsem, and V. Vittal, “Definition and classification of power system stability,” *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, Aug. 2004

[87] J. Endrenyi, S. Aboresheid, R. N. Allan, G. J. Anders, S. Asgarpoor, R. Billinton, N. Chowdhury, E. N. Dialynas, M. Fipper, R. H. Fletcher, C. Grigg, J. McCalley, S. Meliopoulos, T. C. Mielnik, P. Nitu, N. Rau, N. D. Reppen, L. Salvaderi, A. Schneider, and C. Singh, “The present status of maintenance strategies and the impact of maintenance on reliability,” *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 638–646, Nov. 2001

[88] M. Zima, “Special protection schemes in electric power systems (literature survey),” ETH Zurich, Jun. 2002. [Online] Available: http://e-collection.library.ethz.ch/eserv/eth:25263/eth-25263-01.pdf#search=%22(keywords_en:ELECTRIC%20POWER%20ENGINEERING)%22

[89] A. Chakrabortty and M. Ilic, *Control and Optimization Methods for Electric Smart Grids*. Springer, 2012

[90] D. Bienstock and A. Verma, “The N-k problem in power grids: New models, formulations, and numerical experiments,” *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2352–2380, Jun. 2010

[91] P. Palermo, M. Wilks, and D. K. Brancikorinek, “International review of transmission reliability standards,” Australian Energy Market Commission Reliability Panel, Tech. Rep., May 2008. [Online] Available: http://www.aemc.gov.au/getattachment/b077b15a-ea53-4617-b020-653e7df2ee8d/KEMA-Final-Report.aspx

[92] J.-C. Laprie, “Resilience for the scalability of dependability,” in *4th IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, USA, Jul. 2005

[93] K. S. Trivedi, D. S. Kim, and R. Ghosh, “Resilience in computer systems and networks,” in *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, San Jose, CA, USA, Nov. 2009

[94] A. Faza, S. Sedigh, and B. McMillin, “Integrated cyber-physical fault injection for reliability analysis of the smart grid,” in *29th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Berlin, DE, Sep. 2010

[95] C. W. Taylor, D. C. Erickson, and R. E. Wilson, “Reducing blackout risk by a wide-area control system (WACS): Adding a new layer of defense,” in *15th Power Systems Computation Conference (PSCC)*, Liège, BE, Aug. 2005

[96] A. Atputharajah and T. K. Saha, “Power system blackouts - literature review,” in *International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, LK, Dec. 2009

[97] P. Pourbeik, P. S. Kundur, and C. W. Taylor, “The anatomy of a power grid blackout - root causes and dynamics of recent major blackouts,” *IEEE Power and Energy Magazine*, vol. 4, no. 5, pp. 22–29, Sep. 2006

[98] L. L. Lai, H. T. Zhang, S. Mishra, D. Ramasubramanian, C. S. Lai, and F. Y. Xu, “Lessons learned from July 2012 Indian blackout,” in *9th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM)*, Hong Kong, CN, Nov. 2012

[99] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor, and V. Vittal, “Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance,” *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1922–1928, Nov. 2005

[100] B. Liscouski and W. Elliot, “Final report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations,” US Department of Energy, Tech. Rep. 4, Apr. 2004. [Online] Available: https://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/BlackoutFinal-Web.pdf

[101] G. Maas, M. Bial, and J. Fijalkowski, “System disturbance on 4 november 2006 (final report),” Union for the Coordination of Transmission of Electricity in Europe, Tech. Rep., 2007. [Online] Available: http://ecolo.org/documents/documents_in_english/blackout-nov-06-UCTE-report.pdf

[102] C.-H. Lee and S.-C. Hsieh, “A technical review of the power outage on July 29, 1999 in Taiwan,” in *IEEE Power Engineering Society Winter Meeting*, Columbus, OH, USA, Jan. 2001

[103] “The Con Edison power failure of July 13 and 14, 1977,” Federal Energy Regulatory Commission, US Department of Energy, Tech. Rep., 1978. [Online] Available: http://blackout.gmu.edu/archive/pdf/usdept001_050.pdf

[104] A. Kurita and T. Sakurai, “The power system failure on July 23, 1987 in Tokyo,” in *27th IEEE Conference on Decision and Control*, Cambridge, MA, USA, Dec. 1988

[105] “Arizona-Southern California outages on September 8, 2011: Causes and Recommendations,” Federal Energy Regulatory Comission, US Department of Energy, Tech. Rep., Apr. 2012. [Online] Available: https://www.ferc.gov/legal/staff-reports/04-27-2012-ferc-nerc-report.pdf

[106] “The Southern Great Lakes derecho of 1998,” National Oceanic and Atmospheric Administration (NOAA), Tech. Rep., 1998. [Online] Available: http://www.spc.noaa.gov/misc/AbtDerechos/casepages/may30-311998page.htm

[107] “Report on equipment damage of 275-kV Nos. 1 and 2 Koto Lines,” Tokyo Electric Power Company, Tech. Rep., 2006. [Online] Available: http://www.tepco.co.jp/en/news/topics/060824-e.html

[108] “The Southern Great Lakes derecho of 1991,” National Oceanic and Atmospheric Administration (NOAA), Tech. Rep., 1991. [Online] Available: http://www.spc.noaa.gov/misc/AbtDerechos/casepages/jul7-81991page.htm

[109] “The Boundary Waters - Canadian darecho,” National Oceanic and Atmospheric Administration (NOAA), Tech. Rep., 1999. [Online] Available: http://www.spc.noaa.gov/misc/AbtDerechos/casepages/jul4-51999page.htm

[110] “The events of 16 January 2007: Final Investigation Report,” Australian Energy Regulator, Tech. Rep., Sep. 2007. [Online] Available: https://www.aer.gov.au/system/files/Report20into20the20events20of201620January202007_0.pdf

[111] “Year-in-review: 2013 energy infrastructure events and expansions,” US Department of Energy, Tech. Rep., 2014. [Online] Available: http://energy.gov/sites/prod/files/2014/05/f15/2013-YIR-05092014.pdf

[112] “Extensive power outages 19-6-2007 (official announcement),” Enemalta Power Distribution, Tech. Rep., 2007. [Online] Available: http://www.enemalta.com.mt/newsDetails.aspx?id=15714

[113] C. W. Taylor and D. C. Erickson, “Recording and analyzing the July 2 cascading outage (Western USA power system),” *IEEE Computer Applications in Power*, vol. 10, no. 1, pp. 26–30, Jan. 1997

[114] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, “Identifying, understanding, and analyzing critical infrastructure interdependencies,” *IEEE Control Systems*, vol. 21, no. 6, pp. 11–25, Dec. 2001

[115] N. D. Tleis, *Power Systems Modelling and Fault Analysis: Theory and Practice*, ch. Introduction to power system faults, pp. 1 – 27. Newnes, 2008

[116] S. Gautam and S. M. Brahma, "Detection of high impedance fault in power distribution systems using mathematical morphology," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1226–1234, May 2013

[117] D. O. Koval and H. L. Floyd, "Human element factors affecting reliability and safety," *IEEE Transactions on Industry Applications*, vol. 34, no. 2, pp. 406–414, Mar. 1998

[118] A. Dittrich, I. Kaitovic, C. Murillo, and R. Rezende, "A model for the evaluation of user-perceived service properties," in *IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*, Boston, MA, USA, May 2013

[119] P. Kundur, *Power system stability and control*. McGraw-Hill, 1993

[120] M. D. Ilic, L. Xie, U. A. Khan, and J. M. F. Moura, "Modeling of future cyber-physical energy systems for distributed sensing and control," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 4, pp. 825–838, Jul. 2010

[121] Y. Mo, T. H. J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure," *Proceedings of the IEEE, Special Issue on Cyber-Physical Systems*, vol. 100, no. 1, pp. 195–209, Jan. 2012

[122] C. A. Macana, N. Quijano, and E. Mojica-Nava, "A survey on cyber physical energy systems and their applications on smart grids," in *IEEE PES Conference on Innovative Smart Grid Technologies*, Medellin, CO, Oct. 2011

[123] Z. Vale, H. Morais, P. Faria, H. Khodr, J. Ferreira, and P. Kadar, "Distributed energy resources management with cyber-physical SCADA in the context of future smart grids," in *15th IEEE Mediterranean Electrotechnical Conference (Melecon)*, Valletta, MT, Apr. 2010

[124] A. Z. Faza, S. Sedigh, and B. M. McMillin, "The advanced electric power grid: Complexity reduction techniques for reliability modeling," in *27th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Newcastle, UK, Sep. 2008

[125] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344 – 1371, Apr. 2013

[126] A. Borghetti, M. Bosetti, S. Grillo, S. Massucco, C. A. Nucci, M. Paolone, and F. Silvestro, "Short-term scheduling and control of active distribution systems with high penetration of renewable resources," *IEEE Systems Journal*, vol. 4, no. 3, pp. 313–322, Sep. 2010

[127] G. Valverde and T. V. Cutsem, "Model predictive control of voltages in active distribution networks," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2152–2161, Dec. 2013

[128] M. Chertkov, F. Pan, and M. G. Stepanov, "Predicting failures in power grids: The case of static overloads," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 162–172, Mar. 2011

[129] M. R. Qader, M. H. J. Bollen, and R. N. Allan, "Stochastic prediction of voltage sags in a large transmission system," *IEEE Transactions on Industry Applications*, vol. 35, no. 1, pp. 152–162, Jan. 1999

[130] E. Scolari, D. Torregrossa, J. Y. L. Boudec, and M. Paolone, "Ultra-short-term prediction intervals of photovoltaic AC active power," in *International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Beijing, CN, Oct. 2016

[131] Y. Liu, L. Zhan, Y. Zhang, P. N. Markham, D. Zhou, J. Guo, Y. Lei, G. Kou, W. Yao, J. Chai, and Y. Liu, "Wide-area-measurement system development at the distribution level: An FNET/GridEye example," *IEEE Transactions on Power Delivery*, vol. 31, no. 2, pp. 721–731, Apr. 2016

[132] Alstom Press Centre Home, "Alstom and PG&E to advance synchrophasor grid monitoring into proactive grid stability managemen," Aug. 2014. [Online] Available: http://www.alstom.com/press-centre/2014/8/alstom-and-pge-to-advance-synchrophasor-grid-monitoring-into-proac\tive-grid-stability-management/

[133] "Predictive grid quarterly report: Building a predictive grid for the motor city," Tollgrade Communications, Tech. Rep., July 2015. [Online] Available: http://www.tollgrade.com/smart-grid-resources/predictive-grid-quarterly-reports-clinton-global-initiative/

[134] S. Lukovic, I. Kaitovic, G. Lecuona, and M. Malek, "A methodology for proactive maintenance of uninterruptible power supplies," in *7th Latin-American Symposium on Dependable Computing (LADC) - Workshop on Dependability in Evolving Systems (WDES)*, Cali, CO, Oct. 2016

[135] I. Kaitovic and M. Malek, “Optimizing failure prediction to maximize availability,” in *13th IEEE International Conference on Autonomic Computing (ICAC)*, Würzburg, DE, Jun. 2016

[136] N. Taerat, C. Leangsuksun, C. Chandler, and N. Naksinehaboon, “Proficiency metrics for failure prediction in high performance computing,” in *International Symposium on Parallel and Distributed Processing with Applications*, Taipei, TW, Sep. 2010

[137] K. L. Wagstaff, “Machine learning that matters,” in *International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, UK, Jun. 2012

[138] F. Salfner and M. Malek, “Using hidden semi-markov models for effective online failure prediction,” in *26th IEEE International Symposium on Reliable Distributed Systems (SRDS)*, Beijing, CN, Oct. 2007

[139] A. Polze, P. Troger, and F. Salfner, “Timely virtual machine migration for proactive fault tolerance,” in *14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, Newport Beach, CA, USA, Mar. 2011

[140] D. M. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” Dec. 2007. [Online] Available: https://csem.flinders.edu.au/research/techreps/SIE07001.pdf

[141] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi, “Fluid stochastic Petri nets: Theory, applications, and solution techniques,” *European Journal of Operational Research*, vol. 105, no. 1, pp. 184–201, Feb. 1998

[142] J. Dean, “Design, lessons and advice from building large distributed systems,” in *Keynote Address at the 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware (LADIS)*, Big Sky, MT, USA, Oct. 2009

[143] B. Schroeder and G. Gibson, “A large-scale study of failures in high-performance computing systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 337–350, Oct. 2010

[144] T. Chalermarrewong, T. Achalakul, and S. C. W. See, “The design of a fault management framework for cloud,” in *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Hua Hin, TH, May 2012

[145] R. d. S. Matos, P. R. M. Maciel, F. Machida, D. S. Kim, and K. S. Trivedi, "Sensitivity analysis of server virtualized system availability," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 994–1006, Dec. 2012

[146] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," *ACM Computing Surveys*, vol. 46, no. 3, pp. 30:1–30:33, Jan. 2014

[147] F. Salfner, P. Tröger, and M. Richly, "Dependable estimation of downtime for virtual machine live migration," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 1, Jan. 2012

[148] K.-F. Ssu, B. Yao, and W. K. Fuchs, "An adaptive checkpointing protocol to bound recovery time with message logging," in *18th IEEE Symposium on Reliable Distributed Systems (SRDS)*, Lausanne, CH, Oct. 1999

[149] N. H. Vaidya, "Impact of checkpoint latency on overhead ratio of a checkpointing scheme," *IEEE Transactions on Computers*, vol. 46, no. 8, pp. 942–947, Aug. 1997

[150] I. Kaitovic, S. Lukovic, and M. Malek, "Proactive failure management in Smart Grids for improved resilience: A methodology for failure prediction and mitigation," in *IEEE Globecom SmartGrid Workshop*, San Diego, CA, USA, Dec. 2015

[151] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 1157–1182, Mar. 2003

[152] L. C. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *IEEE International Conference on Data Mining (ICDM)*. Washington, DC, USA: IEEE Computer Society, Dec. 2002

[153] F. Milano, "Documentation for PSAT version 2.1.9," Sep. 2014

[154] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *The WEKA workbench (fourth edition)*. Morgan Kaufman, 2016

[155] C. Covrig, M. Ardelean, J. Vasiljevska, A. Mengolini, G. Fulli, E. Amoiralis, M. Jimenez, and C. Filiou, "Smart grid projects outlook 2014," JRC science and policy reports, Tech. Rep., 2014. [Online] Available: http://ses.jrc.ec.europa.eu/smart-grids-observatory

[156] J. Arlat, Y. Crouzet, and J. C. Laprie, “Fault injection for dependability validation of fault-tolerant computing systems,” in *19th International Symposium on Fault-Tolerant Computing (FTCS)*, Chicago, IL, USA, Jun. 1989

[157] F. Obradovic, “Power systems fault injection for design of online disturbance prediction methods,” Master’s thesis, Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland, Jun. 2016

[158] I. Kaitovic, F. Obradovic, S. Lukovic, and M. Malek, “A framework for disturbance analysis in smart grids by fault injection,” *Springer journal on Computer Science - Research and Development (CSRD)*, vol. 32, no. 1, pp. 93–103, Mar. 2017

[159] Power systems test case archive. University of Washington. [Online] Available: https://www.ee.washington.edu/research/pstca/

[160] Distribution test feeders. IEEE PES Distribution System Analysis Subcommittee’s (Distribution Test Feeder Working Group). [Online] Available: http://ewh.ieee.org/soc/pes/dsacom/testfeeders/

[161] “IEEE recommended practice for monitoring electric power quality,” *IEEE Std 1159-2009 (Revision of IEEE Std 1159-1995)*, pp. c1–81, Jun. 2009

[162] F. A. Viawan and D. Karlsson, “Voltage and reactive power control in closed loop feeders with distributed generation,” in *IEEE Lausanne Power Tech*, Lausanne, CH, Jul. 2007

[163] M. H. J. Bollen, K. Stockman, R. Neumann, G. Ethier, J. R. Gordon, K. van Reussel, S. Z. Djokic, and S. Cundeva, “Voltage dip immunity of equipment and installations - messages to stakeholders,” in *15th IEEE International Conference on Harmonics and Quality of Power (ICHQP)*, Hong Kong, China, Jun. 2012

[164] J. A. Martinez and J. Martin-Arnedo, “Voltage sag stochastic prediction using an electromagnetic transients program,” *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1975–1982, Oct. 2004

[165] J. D. Rodriguez, A. Perez, and J. A. Lozano, “Sensitivity analysis of k-Fold Cross Validation in prediction error estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 569–575, Mar. 2010

[166] R. R. Bouckaert, "Bayesian Network classifiers in Weka for version 3-5-7," University of Waikato, New Zealand, Tech. Rep., May 2008

[167] S. le Cessie and J. van Houwelingen, "Ridge estimators in Logistic Regression," *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992

[168] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. Cambridge, MA, USA: MIT Press, 1999

[169] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core Vector Machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research*, vol. 6, no. 1, pp. 363–392, Apr. 2005

[170] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Springer Journal on Machine Learning*, vol. 6, no. 1, pp. 37–66, Jan 1991

[171] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993

[172] J. Su and H. Zhang, "A fast Decision Tree learning algorithm," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, Jul. 2006. [Online] Available: http://dl.acm.org/citation.cfm?id=1597538.1597619

[173] A. Ng, "Preventing "overfitting" of cross-validation data," in *Fourteenth International Conference International Conference on Machine Learning (ICML)*, Nashville, Tennessee, USA, Jul. 1997

[174] M. Hartung, E. Baerthlein, and A. Panosyan, "Comparative study of tap changer control algorithms for distribution networks with high penetration of renewables," in *CIRED International workshop on Challenges on implementing active distribution system management*, Roma, Italy, Jun. 2014

[175] "Short duration voltage sags can cause disruptions," Pacific Gas and Electric Company, Tech. Rep., Jun. 2012. [Online] Available: https://www.pge.com/includes/docs/pdfs/mybusiness/customerservice/energystatus/powerquality/voltagesags.pdf

[176] G. Olguin, “Voltage dip (sag) estimation in power systems based on stochastic assessment and optimal monitoring,” Ph.D. Thesis, Division of Electric Power Engineering, Department of Energy and Environment, Chalmers University of Technology, Göteborg, SE, 2005

[177] N. Neves, J. Antunes, M. Correia, P. Verissimo, and R. Neves, “Using attack injection to discover new vulnerabilities,” in *International Conference on Dependable Systems and Networks (DSN)*, Philadelphia, PA, USA, Jun. 2006