Online Identification of Cascading Events in Power Systems with Renewable Generation using Machine Learning

PhD Thesis

Georgios Nakas

Institute for Energy and the Environment
Electrical and Electronic Engineering
University of Strathclyde, Glasgow

February 27, 2024

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

This PhD project deals with the Modelling of Cascading Events in Power Systems and their Online Identification with Machine Learning, considering the integration of Renewable Energy Sources. Cascading events involve highly complex dynamic phenomena and in some cases can pose significant challenges to the stability and reliability of power grids, leading even to blackouts. The intermittent nature of renewable generation introduces additional complexities, as the system dynamic behavior following a contingency becomes more unpredictable. Consequently, there is an increasing need for cascading event identification methods that can effectively handle these emerging challenges and ensure secure network operation.

Machine Learning methods can extract complex relationships from power system data, by capturing the underlying dynamics, offering a promising tool for the accurate and timely identification of the online system state. In addition, due to the extensive installation of Phasor Measurement Units in modern power systems, it is possible to acquire measurement data related to electrical system variables in close-to-real time.

The thesis first delves into the understanding of cascading events appearance, as defined by the discrete action of protection devices, using detailed dynamic simulations and considering uncertainties associated with network operating conditions, contingencies and renewable generation. To address the online nature of the problem, supervised machine learning methods that utilize measurement data are developed. Different contemporary machine learning approaches are investigated, to identify the most suitable techniques for the detection of the appearance of cascading events, formulated as a binary classification problem, and the prediction of the reason of the upcoming cascading event, formulated as a multi-class classification problem. Furthermore, this

thesis explores the challenges associated with the application of machine learning models on power system data, such as the online inference time, class imbalance, practical considerations related to measurement data and investigates techniques for model explainability to enhance the trustworthiness of the developed models.

The contributions of this thesis lie in the development of machine learning-based techniques for online identification of cascading events in power systems, enabling more proactive and efficient situational awareness. These insights have the potential to significantly enhance the resilience and stability of power grids, minimizing the risk of large-scale blackouts and improving the overall reliability of the system.

Georgios Nakas is sponsored through Engineering & Physical Sciences Research Council (EPSRC) Research Excellence Award (REA) and is supervised by Dr. Panagiotis Papadopoulos and Professor Graeme Burt.

Contents

List of Figures

List of Tables

List of Tables

Acknowledgements

First and foremost, I would like to express my gratitude and appreciation to my first supervisor Dr. Panagiotis Papadopoulos for his ongoing support and valuable guidance through my Ph.D. His interest and eagerness to help contributed as a motivation to work efficiently and accomplish my Ph.D.

Furthermore I would also like to express my gratitude to all the members from my research team for all the constructive discussions and the collaborative work that we carried out together.

I am also very grateful to Engineering & Physical Sciences Research Council (EPSRC) for granting me a Research Excellence Award (REA) scholarship that covered the funding of my Ph.D.

Last but not least, I would like to express my deepest gratitude and appreciation to my family and friends for all the patience and support through my Ph.D. program.

Acronyms

AI Artificial Intelligence.

AGC Automatic Generation Control.

ANN Artificial Neural Network.

AVR Automatic Voltage Regulator.

CART Classification and Regression Tree.

CNN Convolutional Neural Network.

DSA Dynamic Security Assessment.

DT Decision Tree.

ET Extra Trees.

EV Electric Vehicle.

FRT Fault-Ride Through.

GOV Governor.

GRU Gated Recurrent Unit.

HV High Voltage.

HVDC High Voltage Direct Current.

IEC International Electrotechnical Commission.

IEEE Institute of Electrical and Electronics Engineers.

KNN K-nearest neighbors.

LDA Linear Discriminant Analysis.

LSTM Long Short-term Memory.

LTC Load Tap Changer.

MAE Mean Absolute Error.

ML Machine Learning.

MLP Multilayer Perceptron.

NB Naive Bayes.

NSG Non-synchronous Generator.

OEL Over-excitation Limiter.

OPF Optimal Power Flow.

PMU Phasor Measurement Unit.

PSS Power System Stabilizer.

PV Photovoltaic.

RES Renewable Energy Source.

RF Random Forest.

RMS Root Mean Square.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

SG Synchronous Generator.

SHAP SHapley Additive exPlanations.

SVM Support Vector Machine.

TCN Temporal Convolutional Network.

UFLS Under-frequency Load Shedding.

WF Wind Farm.

WT Wind Turbine.

XAI eXplainable Artificial Intelligence.

XGBoost eXtreme Gradient Boosting.

Chapter 1

Introduction

1.1 Introduction to Research and Motivation

In recent years, the global transition towards a sustainable energy future has witnessed a remarkable surge in the integration of Renewable Energy Sources (RES), as well as other power electronic interfaced devices, in numerous countries, including the UK. This transformative shift towards sustainable energy solutions is driven by a global effort to combat carbon emissions and mitigate the environmental impact of traditional energy generation methods which heavily relied on fossil fuels. As a result, Synchronous Generators (SGs) tend to be displaced by Converter Interfaced Generators (CIGs), which possess different dynamic attributes.

This wider variation in power system operating conditions and different dynamic behaviour can potentially lead to unforeseen dynamic interactions which might cause the appearance of cascading events. Cascading events are low-probability, high-impact events, the propagation of which can lead to load-shedding events, and in the worst case, large-scale blackouts, disrupting the electricity supply and resulting into consequences that can severely impact society [2]. The fast and precise online identification of such events can provide vital information in exposing network vulnerabilities and designing appropriate predictive measures or taking corrective measures with higher security to avoid cascading events from spreading.

The variability and intermittent nature of RES introduce a level of unpredictability

and complexity, leading to significantly different dynamic behaviour of the power system that may vary in a both temporal and spatial manner. RES are usually distributed across different geographical locations, and their power output can exhibit fluctuations over time, leading to different generator dispatch patterns and system topologies. The introduction of new technologies, such as the deployment of energy storage systems, and the connection of Electric Vehicles (EVs) to distribution grids, bring more uncertainty to the demand side as well. Combined with the presence of interconnected networks with High Voltage Direct Current (HVDC) lines and fast evolving market regulations, the number of possible operational scenarios that require stability assessment is constantly increasing.

In addition, some CIGs, such as Photovoltaic (PV) arrays, might not rely on mechanical generators for power conversion, and their response generated by power electronic devices demonstrates very fast and non-linear characteristics. This complex dynamic behaviour appears even in the case of wind generators, where the mechanical generators operate asynchronously with the network. Based on the aforementioned reasons, the displacement of SGs by RES and the consequent reduction to system inertia can cause challenges for maintaining power system stability.

The ability of an electrical power system to remain stable following a contingency is referred to as power system stability [3]. The power system stability is classified into categories, in order to identify the types of instability involved, and design targeted preventive or corrective measures. According to the original power system stability classification [3], the three instability types are the following:

- Rotor angle stability, which refers to the ability of the SGs in a power system to remain in synchronism after being subjected to a disturbance.
- Voltage stability, which refers to the ability of a power system to maintain steady voltages at all network buses after being subjected to a disturbance
- Frequency stability, which refers to the ability of a power system to maintain a steady frequency following a severe disturbance that causes a significant imbalance between generation and demand.

The changes in power system operation that the introduction of CIGs has brought, resulted in a departure from the previously established norms and behavior patterns. The effect has been of such magnitude that it necessitated the re-establishment of power system stability. For this reason, in [1] two additional categories have been introduced to account for the faster dynamic response of RES:

- Converter-driven stability, which refers to the dynamic interactions of the control systems of power electronic-based components.
- Resonance stability, which refers to subsynchronous resonance related to resonance between series compensation and the mechanical oscillations of the generator shaft or to purely electrical resonance.

As any type of instability can occur independently of the other types, it is significant to always consider the overall stability of the power system. The revised power system stability classification is illustrated in Fig. 1.1.

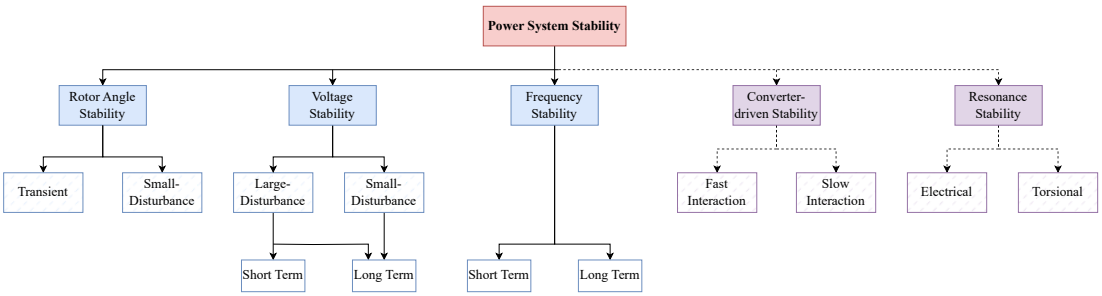

Figure 1.1: Revisited Power System Stability Classification [1].

In practice, the protection devices that are installed in power systems might activate before instability limits are reached, as the thresholds and settings of these devices are set with a margin of safety. The main objective of protection devices is to isolate components that do not operate under safe operating conditions in order to leave the unaffected parts of the system in normal operation. To achieve this, protection devices monitor the electrical parameters of the network and activate protective measures, such as circuit breakers. A component outage can cause further outages, leading to a cascading event sequence. As the examination of previous blackouts in [4] has shown,

in many cases a singe initial fault has caused a rapid succession of cascading events, which ultimately resulted in large blackouts.

The analysis of historical transmission outage data in [5] has shown that the evolution of cascading events propagation in time appears in two phases, a slow and a fast phase, with the fast phase being in the order of minutes and faster by at least one order of magnitude compared to the slow phase. The rapid evolution of cascading events during the fast phase requires targeted and swift control actions, which is not always possible by human operators. It is also worth noting that even though existing regulations mandate the consideration of cascading events, practical frameworks for effectively assessing cascading failures are still lacking in development [4]. This creates the need for more research in preventive and mitigation strategies regarding the appearance and propagation of cascading events.

The extensive installation of Phasor Measurement Units (PMUs) in modern power systems, has made it possible to provide data that hold valuable information about the system dynamics, offering unprecedented visibility into its operational status. PMUs are devices that can be installed at different locations of transmission systems and provide real-time measurements of electrical phasor quantities. The time measurements are synchronized using a common time reference point, usually provided by the Global Positioning System (GPS). These time synchronized measurements are referred to as synchrophasors [6]. The main advantage of PMUs is that they can provide detailed information much faster than SCADA technologies [7], and therefore equip power system utilities with enhanced monitoring and control capabilities. The utilization of real-time measurement data provided by PMUs by data-driven methods has the potential to facilitate the fast prediction of imminent cascading events and enable corrective control actions, all faster than with human system operators.

As for proper system monitoring PMUs are placed across various locations of a power grid, they generate large volumes of data, in the format of multivariate time-series measurements. The high-dimensional data, combined with the inherent complexity and nonlinear temporal characteristics of power systems, create challenges for the processing and extraction of relevant information from such large datasets, which can

be computationally intensive. In addition, PMU data might contain noise and missing values, making the utilization of such data a challenging task.

Artificial Intelligence (AI), and especially Machine Learning (ML), has been gaining significant attention nowadays, as it is incorporated into practical application in various domains, with electrical power systems being no exception. The integration of such techniques in power systems has created new possibilities for intelligent situational awareness and enhanced overall system performance. ML algorithms can extract valuable insights and learn patterns from vast amounts of data, uncovering predictions from complex power system dynamics, which was not possible with previous traditional methods based on statistics and pre-defined rules. Furthermore, when deployed in an online setting these models can provide predictions with a fast inference time, enabling power system operators to make informed decisions and take control actions with improved security.

For the aforementioned reasons, this thesis focuses on the following main points:

- Better understanding of the appearance of cascading events in modern power systems with RES penetration and on the effective prediction of these events based on ML techniques and measurement data. More specifically, this PhD project aims to investigate the causes, the involved mechanisms, and the characteristics of cascading events, in an effort to identify key factors that contribute to their occurrence as well as potential network vulnerabilities. This deeper understanding can facilitate the development of preventive measures that can be set in place and prevent the occurrence of cascading events.

- The development of effective strategies to predict the appearance of cascading events. To this end, contemporary ML algorithms are implemented in order to effectively predict the cascading events that follow the first external contingency during online application, by utilizing measurement data which can be acquired by PMUs. The need for interpretability in black-box models applications is also taken into consideration. A large number of detailed dynamic RMS simulations, including the action of protection devices, and considering a wide range of sce-

narios with different levels of renewable energy penetration and system loading, are conducted to evaluate the performance and effectiveness of the proposed techniques.

- The methods developed in this thesis can also be utilized for planning studies. Instead of running long time domain simulations, the simulations can be performed for a short time window and then the ML model can be used to predict the outcome, reducing significantly the computational time needed.

- Ultimately, this thesis aims to provide a set of methods for the mitigation of risks associated with cascading failures, that can contribute to the reliability and secure operation of modern power systems, while meeting the increasing demands of RES penetration.

1.2 Research Methodology

The research work presented in this thesis has been carried out in certain stages through the course of this PhD program. The initial step was to perform a review of the existing methods that investigate the modelling of cascading events in power systems and identify potential contributions. This literature review has revealed that most of the existing methods have focused on the modelling of cascading events using static methods [8], [9]. However, recent studies have shown that dynamic simulations can provide more details about the evolution of cascading events and the mechanisms involved, at the expense of increased computational effort [10], [11]. Another important observation has been that some of these methods do not consider the penetration of RES [12], which is a critical factor that has to be accounted when investigating the secure operation of modern power systems.

Furthermore, through a review of the existing methods for the online identification of the dynamic system state following a contingency, it has been identified that most of these methods focus on the online identification of either transient, voltage or frequency instability events [13], [14], and not on the prediction of cascading events as defined by the action of protection devices. As revealed from this review, ML-based model

applications for online power system security using measurement data have shown very promising results [15], [16].

Based on the previous identified research gaps, first studies focused on the modelling of cascading events in a relatively small, but detailed, transmission network. To capture the appearance of cascading events, the test system has been augmented with protection devices and non-synchronous generators that represent the RES penetration. Dynamic RMS simulations have been performed to capture in detail the system dynamic behaviour following the initial contingency. Each cascading event is characterized by the reason of the event, the component that trips and the time of the event, giving the sequence in which the cascading events appear.

The next step was to move to a larger test network, in order to be able to identify more complicated cascading event patterns, and to identify the impact that power system control mechanisms have on the appearance of cascading events. For this reason, a larger test model was augmented with the action of protection devices, RES units and the two control mechanisms examined, Automatic Generation Control (AGC) and Load Tap Changers (LTCs). In a similar manner, dynamic RMS simulations have been performed and a thorough analysis of the results has taken place.

Taking into account the insights from the previous steps, supervised ML-based frameworks have been proposed for the prediction of cascading events before they appear. The available datasets as generated from the previous steps have been preprocessed to represent typical PMU measurements. It should be noted that only simulated data have been used. The time-domain simulated measurements have been interpolated using a typical PMU sampling rate. Experiments using data with added noise and missing features, which can be challenges associated with PMU data, have been conducted in order to highlight the suitability of the proposed framework for real-life applications. Techniques to address the highly imbalanced datasets, the large size of data and to reduce the inference time have been employed. Furthermore, explainability methods that can provide information about which power system features are important for the ML models prediction are applied. The proposed frameworks have been evaluated using commonly applied metrics for each task (e.g. regression,

binary classification, multi-class classification) as identified from relevant literature, paying also particular attention to the power systems application context (e.g. how wind generation affects the model performance).

1.3 Thesis Contributions

The research work presented in this thesis focuses on the modelling of cascading events using dynamic simulations, and on the online identification of cascading events using measurement data. The key contributions of this thesis are outlined below:

- The development of a modelling approach and analysis for the characterization of cascading events in power systems with RES and protection devices. A large number of simulations is performed on a detailed dynamic system and the cascading event sequences that appear are identified and characterized by the time, the components, and the reason of trippings involved. The effect that Automatic Generation Control (AGC), a frequency related mechanism, and Load Tap Changers (LTC), a voltage related mechanism, have on the appearance of cascading events is also identified.

- A methodology for interpretable predictions of the appearance of cascading events using ML algorithms by utilizing the initial operating condition values during steady-state grid operation. An explainable AI technique, SHapley Additive exPlanations (SHAP), is utilized to provide further insights about the decision-making process of the ML models. This aims to enhance the trustworthiness of these models and their widespread use in system monitoring and control operation.

- A framework for the online identification of the appearance of cascading event utilizing time-series measurements that are available from PMUs. Deep Learning models that can handle sequential data and have a fast inference time (in the range of 42ms) are implemented in order to predict if a cascading event will appear or not, while considering practical aspects related to PMUs, such as network delays,

noisy signals and missing features.

- A method for the online identification of the reason of the cascading event that follows in a sequence, utilizing moving time-windows. In this case, the specific reason for the next cascading event in a sequence is predicted, which can provide more information to system operators. Techniques to address the highly imbalanced dataset, reduce the inference time and the method application on a different system topology are showcased.

1.4 Thesis Layout

An overview of the chapters contained in this thesis is presented below:

Chapter 2 presents a detailed literature review of research works focused on the modelling of cascading events and on the existing methods for the online identification of security assessment in power systems. The aim of this investigation is to find out the state-of-the-art of cascading events modelling using dynamics, and to identify what methods have been so far developed to predict the power system online state following a disturbance. Furthermore, potential research gaps and opportunities for research contributions in the context of cascading events are identified. The findings of this analysis set the ground for the technical work presented in the following chapters.

Chapter 3 presents a framework for the modelling and characterization of cascading events in power systems with renewable generation. The cascading event patterns that appear are identified, and each cascading event is characterized by the power system component that trips, the time of the event and the reason for tripping. The proposed method is initially applied on a modified version of the Anderson-Fouad 9 bus model, augmented with renewable generation and protection devices, the action of which can capture tripping events related to transient, voltage and frequency stability. The sequence of cascading events is captured through RMS simulations, considering a wide range of initial operating conditions. The cascading event patterns that appear are thoroughly analyzed, based on how many times they appear, the components that are involved and the type of instability that is involved. Additionally, it is investigated

how these patterns change according to the amount of wind penetration and system loading. Next, the impact of Load Tap Changers, a voltage related control mechanism, and the impact of Automatic Generation Control, a frequency related mechanism, on the appearance of cascading events is investigated. To achieve this, three scenarios with and without considering the action of these mechanisms are defined and implemented. The impact of the two control mechanisms is showcased on a larger network model, a modified version of the IEEE-39 bus network that includes RES and protection devices. The number and reason of cascading events, the average load loss and the time between consecutive events are compared to quantify the impact of these mechanisms. Furthermore, an analysis is conducted on the most common cascading event patterns and how these differentiate across the defined scenarios.

Chapter 4 investigates the use of various ML models to predict: i) the probability of cascading events by utilizing initial operating conditions, ii) the appearance or not of cascading events as a binary classification by including also information about the initial fault location. Furthermore, an Explainable AI (XAI) technique is utilized to explain the predictions of the ML model with the best performance. A dataset from the simulations carried out on the modified version of the IEEE-39 bus test system in Chapter 3 is used, considering only the values of the initial conditions during steady-state, before the application of the initiating three-phase fault. Without considering the initial fault location, regression ML models are trained and evaluated on predicting the probability of cascading events. Next, the initial fault location is added to the input features as a discrete event, and the problem is formulated as binary classification, predicting if cascading events will appear in this case. A permutation feature importance analysis and the SHAP XAI technique are performed on the best performing model in order to gain more insights about its decision making process.

Chapter 5 introduces the use of time-series measurement data and ML models with recurrent architecture to predict the appearance of cascading events in close-to-real time. Particular focus has been given on the impact that the time window length has on the prediction model performance. The evaluation metrics as calculated on the test dataset are presented and discussed, investigating additionally how these change for

individual operating conditions, in this case system loading and wind penetration. A permutation feature importance analysis is also carried out in this chapter, to identify which PMU time-series measurements have the biggest impact on the model predictions. Taking into consideration practical applications, the model performance is evaluated with limited availability of measurement data and added noise.

Chapter 6 provides a framework for the online identification of the reason of cascading event sequences in power systems, by utilizing deep learning models. The methodology presented in this chapter utilizes measurement data, as in Chapter 5, but the problem in this case is formulated as a multi-class classification problem, with the aim of predicting specifically the reason of the cascading events, and not only if a cascading event will appear or not. What is more, the formulation considered in this chapter utilizes moving time windows, to predict the reason of the next cascading event in a sequence, as they appear in an online setting. The proposed methodology is applied on the modified version of IEEE-39 bus, with the addition of distance protection relays. In this study case, an efficient sampling technique is utilized to define the initial operating conditions, in order to efficiently capture the input variability. The models evaluation metrics are presented and analyzed, as well as the model performance per cascading event sequence and on a different system topology.

Finally, Chapter 7 presents an overview of the research work that has been carried out as part of this thesis and summarizes the key findings and contributions. Furthermore, potential areas of research for future work are highlighted.

1.5 Publications

The following publications have resulted from the research work that is included in this thesis:

Journal Papers - Leading Author:

- G. A. Nakas, A. Dirik, P. N. Papadopoulos, A. R. R. Matavalam, O. Paul and D. Tzelepis, "Online Identification of Cascading Events in Power Systems With Renewable Generation Using Measurement Data and Machine Learning," in IEEE

Access, vol. 11, pp. 72343-72356, 2023, doi: 10.1109/ACCESS.2023.3294472. [17]

Conference Papers - Leading Author:

- G. A. Nakas and P. N. Papadopoulos, "Investigation of the Impact of Load Tap Changers and Automatic Generation Control on Cascading Events," 2021 IEEE Madrid PowerTech, 2021, pp. 1-6, doi: 10.1109/PowerTech46648.2021.9494775. [18]

- G. A. Nakas and P. N. Papadopoulos, "Investigation of Cascading Events in Power Systems with Renewable Generation," 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2020, pp. 211-216, doi: 10.1109/ISGT-Europe47291.2020.9248818. [19]

Journal Papers - Co-author:

- A. S. C. Leavy, G. A. Nakas and P. N. Papadopoulos, "A Method for Variance-Based Sensitivity Analysis of Cascading Failures," in IEEE Transactions on Power Delivery, 2022, doi: 10.1109/TPWRD.2022.3199150. [20]

Journal Papers - Leading Author that have been submitted and are currently under review:

- G. A. Nakas and P. N. Papadopoulos, "Online Identification of Cascading Event sequences in Power Systems using Deep Learning," submitted to IEEE Systems.

Chapter 2

Literature Review

In this Chapter, a literature review is presented, outlining the main points from representative studies that focus on two main sections: First literature related to current methods of modelling cascading events in power systems is reviewed, followed by a review of the main state-of-the-art methods used for the online identification of the power system state following a contingency.

2.1 Modelling Cascading Events in Power Systems

In [4], the IEEE Task Force on Understanding, Prediction, Mitigation, and Restoration of Cascading Failures analytically describes the state-of-the-art methodologies for cascading event analysis, as well as the main challenges involved. It is highlighted that cascading events can be impacted to a great extent by the system state, and the action of the many power system mechanisms that can cause subsequent cascades. Furthermore, recent changes in power systems operation, such as the uncertainty due to wind generation and rapid evolving electricity market policies, can contribute to higher risk of cascading events, creating the need for understanding better the appearance of cascades. It is also noted that based on historical data, a lot of large blackouts comprised of cascading event sequences that originated from a single initial contingency. Based on the results of this thorough review, the authors conclude that effective tools for the assessment and mitigation of large cascading sequences have not yet been well

developed.

The IEEE working group on cascading failures has presented in [2] the main approaches for the validation of cascading failure tools, as well as directions for future work. It is emphasized that the analysis of cascading failures is a very challenging task, because of the many complicated involved mechanisms and interactions during diverse time scales. For this reason, it is of significant importance to include the accurate modelling of such mechanisms in the test models used for cascading failure analysis. More specifically, the modelling of protection devices, the accurate definition of the initial network conditions and the SGs dispatch as defined by the generator cost, while respecting power flow constraints, are key aspects that need to be considered. Furthermore, through cascading event studies, statistics and metrics regarding the patterns of cascades that appear need to be provided. It is concluded that further research is needed for the definition of the aspects and the level of detail that need to be modelled in a test system for cascading event analysis.

In [8] a method for the identification of sequence of cascades that lead to catastrophic failures is presented. For each contingency a severity index is calculated, based on voltage instability, active and reactive power margins, and frequency deviation. The severity indices are created and evaluated using a load flow model that takes into account frequency and voltage dependencies. Next, pattern recognition and fuzzy estimation are utilized to identify the most probable failure sequences for various operating conditions. The proposed method is showcased on the IEEE 39-bus system and on the 390-bus Northern Region State Electricity Board (NREB) system, and the results present an analysis of the identified collapse sequences. Some of the limitations of this method is that transient phenomena are not considered, and that the method can not identify the events with low risk, therefore not all the possible cascading event sequences are taken into account.

Another approach using static simulations is described in [21]. A security-constrained optimal power flow (SCOPF) problem is solved by balancing the pre-fault security cost with the expected corrective actions cost and the cost of loss of energy due to the possible outages. The proposed framework assesses both generation and transmission

N-k contingencies and is applied on a 2-bus system. The results show that common mode outage significantly impact the contingency probabilities, affecting subsequently the security optimal level.

The framework introduced in [9] proposes a stochastic "Random Chemistry" algorithm for the identification of sets of multiple contingencies that result in system failure. The proposed stochastic algorithm initially considers a large set of contingencies that cause cascading failures. Next, subsets of the original set are randomly picked up and simulated until a smaller subset that causes cascading failures is found. This subset then takes the place of the original set and the same procedure is repeated until the minimum number of contingencies that causes cascading failures is found. This framework is showcased on the Polish transmission system with 2383 buses using DC power-flows. According to the results, random search requires at least 85 times more simulations compared to the random chemistry approach for the identification of cascades. In this method, it is assumed that the multiple contingencies appear simultaneously, while in reality they can occur in various timings. Furthermore, the utilized power system simulator can capture only a subset of the mechanisms involved in cascading failures.

A different approach using the same simulator and case data as in [9] is presented in [22], in which the results from many cascading simulations are used to create an influence graph that provides information on how the cascades evolve in a particular system. The cascading failures propagate locally on the generated influence graph, which is different compared to the original network structure. Results show that the cascading events do follow patterns, the identification of which is useful in reducing the risk of large cascading blackouts. Furthermore, with the proposed method critical components, the tripping of which can lead to blackouts, can be identified. Since this is a statistical method, several assumptions have been made, such as the assumption that multiple cascades propagate independently during a generation and assumptions related to the model parameters and data size. The authors conclude that future work is needed for the evaluation of these assumptions.

The method presented in [23] is based again on the generation of an influence graph, combined with Markov chain, to provide information about the mitigation of

cascades. In this case, the method is illustrated on historical line outage data. The results showcase that the proposed method can accurately calculate the probability for cascading sequences of different sizes and identify critical lines.

In [24], the statistics of cascades are estimated using branching processes. The proposed methodology is evaluated on the IEEE-118 bus and NPGC 500 bus systems, using DC and AC load flow OPA simulations. The simulated cascading outage data are used to estimate the parameters of the branching process models, which can then be used to estimate the distribution of lines tripped and the load shed. This way, the propagation size can be estimated by performing fewer simulations.

The previous methods that focus on the modelling of cascading events are showcased using static simulations, which can capture only a set of the cascading event mechanisms. The study in [25] proposes a model that combines DC power flows with dynamic elements, allowing for more accurate representation of under-frequency load shedding. While this method can increase the modelling detail, it still does not capture phenomena like voltage collapse.

Dynamic simulations, although more computationally intensive, can provide more details about the appearance of cascading events, compared to static methods. A comparison between static and time domain simulations running on the same test system and applying the same contingencies is presented in [10]. The results reveal a good consistency in the early steps of the simulation, however the last evolutions are not accurately caught by the static model because of the occurrence of fast instability events.

A similar behaviour is observed in [11], where the authors present the COSMIC dynamic model and compare it to a quasi-static model. The results showcase that both models behave similarly during the first events of a cascading sequence, however the later stages of the sequence are only captured by the dynamic model. In [26] a static simulator is compared to a novel dynamic one, based on the simulations on the IEEE-39 bus, a-200 bus and a 2000-bus network model. Static simulations were found to underestimate the power loss, especially on larger models. In addition, the more complex transient behaviours observed in larger networks could not be captured by the static

model. The authors further expand on their work in [27], and introduce a mitigation measure for cascading failures by identifying critical lines through a sensitivity index and limiting the power flow on these critical lines.

A search algorithm for identifying plausible harmful $N-k$ contingencies is presented in [12]. Starting from an initially defined list that contains contingency sequences, new contingencies that may lead to cascades are identified, based on whether a contingency sequence leads to cases of instability or new system steady-state. Dynamic simulations are performed on the 32-bus IEEE Nordic Test System, with the addition of protection devices, considering up to three contingencies. Examples of the identified cascading event sequences that cause instability or non-viable system conditions, after applying initial contingencies, are analytically described. The proposed method however considers only conventional generation, not taking into account challenges associated with RES penetration.

The importance of transient dynamical behaviour on the appearance of cascading events is highlighted in [28], using the swing equation to represent system dynamics and eventually investigating line failures and identifying the critical lines in the network.

The method in [29] introduces a risk-based security assessment to investigate the reliability of power systems with high penetration of RES, focusing on transient instability events. The proposed method is applied on a synthetic test system with type-4 wind turbines that represent the network RES generation and protection devices. The results show that this methodology can identify critical operating conditions, information which can be subsequently used to re-define the standards for transient stability.

The overview of power system dynamics and protection devices presented in [30] identifies that further research is needed in the area of modelling using dynamic models to capture the fast cascading sequences, with adequate representation of protection devices, which drive the cascading events appearance. Furthermore, the inclusion in the simulations of RES models with tripping mechanisms, that are more sensitive to instabilities, is a key factor.

2.2 Online Identification of Instability Events in Power Systems.

In modern power systems, the uncertainty that comes with the integration of renewable generation makes it not feasible to address the problem of online dynamic security in real-time by a brute-force approach, as it is not possible to perform timely dynamic simulations for every possible pre-fault network operating condition, involving different loading and wind penetration states. For this reason, intelligent approaches which have the ability to predict uncertainty with the use of real-time measurement data are being investigated to ensure the secure operation of modern power systems with increasing renewable generation.

The computational platform developed within the iTesla Project for improving the network operational security under load and renewable generation related uncertainty is described in [13]. This platform, by incorporating ML techniques, can be used for online static and dynamic security assessment and can also provide possible control actions in order to avoid instability cases with the use of optimization tools. The application of the proposed framework is showcased on a version of the French HV network, focusing on the overload security problem. The results highlight the critical significance of representing forecast uncertainty, as seemingly stable forecast network conditions can actually lead to unstable cases that must be proactively addressed. The offline phase of this platform is presented in [31], where a large amount of dynamic simulations, for various operating conditions and contingencies, is performed and decision trees are used to extract security rules.

In [32] the authors present an online transient stability assessment method for prediction of cases with instability. The proposed method is based on the generation of bitmaps consisting of trajectories acquired from PMUs of generators, which are used to train and evaluate a convolutional neural network (CNN). The model can predict stable, aperiodic unstable or oscillatory unstable cases. The proposed method is showcased on the New England 39-bus model and on the Western Electricity Coordinating Council (WECC) 179-bus system, performing with 97.7% and 98.1% accuracy respec-

tively, outperforming other ML classifiers, such as support vector machine, decision tree and ensemble methods. Moreover, experiments with noisy PMU signals show small reduction in model performance.

A similar approach using a CNN model is presented in [33], this time applied on small-signal stability. The state of the power system is transformed into images, by mapping the active power, reactive power and voltage (PQV), to RGB channels. The dataset consists of simulations performed on the NESTA 162-bus system, split into 70%-10%-20% for training, validation and test sets respectively. According to the results, the model performs with an accuracy higher than 98%, and with an inference time over 255 times faster than traditional small-signal stability analysis.

A probabilistic framework for online identification of power system dynamic behaviour, considering the impact of RES penetration is presented in [14]. The proposed methodology, based on decision trees and hierarchical clustering, identifies the unstable generator groups and the order in which these groups become unstable. Dynamic simulations on the 68-bus IEEE model with the addition of renewable generation are performed for the method application. The accuracy of the decision trees across all the considered test cases is around 99%, however it can drop at 62% when they are tested on a different system topology than the one they were trained with. Overall, it is concluded that the penetration of RES can significantly affect the network dynamic behaviour.

A data driven framework to address the problem of transient stability assessment is presented in [34]. The original power system data are transformed into a low-dimensional representation state using a deep belief network. The deep belief network is fine-tuned by considering the power system topology and the expected classification accuracy index during the learning process. Finally, the low-dimensional representation space is input to a linear model for the classification of unstable and stable cases. The method application is showcased on a real 1300-bus system in central China. Results show that the proposed method outperforms a SVM model.

In [15] an online identification method for transient stability assessment using Long Short-Term Memory (LSTM) network is proposed. The main difference from the pre-

vious methods is that it allows the model to learn from temporal data dependencies, therefore capturing more accurately the dynamic evolution of the system. Three different study cases are considered, on a 10-machine, on a 162-bus and on a 145-bus network model. The LSTM-based model achieves an almost perfect accuracy of at least 99.98%, with an average response time of 1-2 cycles. Also, in this paper a sensitivity study of the PMU measurements is carried out, using a sequential feature selection algorithm to find the minimum number and the best locations of PMUs in the test system model.

A LSTM-based model with added spatial attention is also utilized in [16], for short-term voltage stability assessment. The model learns from both temporal and spatial information, without the need of prior knowledge about the specific network topology. The proposed framework is evaluated on the Nordic test system and on a 2000-bus network, performing with an accuracy of 98.20% and 98.64% respectively. The model is also evaluated using noisy and missing PMU signals, which are shown to affect only slightly the model performance.

An approach to online static security assessment using neural networks is presented in [35]. In this study, two neural network models have been used to assess the system security for variable operating conditions, including different loading states and $N-1$ line contingencies. The dataset consists of load-flow results performed on the IEEE 30-bus test system. The proposed neural networks, trained with these input data, predict two performance indices related to active power and voltage.

A hybrid system composed of three components for transient stability prediction is presented in [36]. A pre-processor splits the synchronous machines into groups, with each group containing two generators. Each of these groups gets assigned to one neural network, which is responsible for extracting the mapping function of each group. These responses are combined by an interpreter which determines the status of the system in a binary fashion, that is transient instability or stability. The proposed method is applied on a 4-bus system and on the 39-bus New England system. The authors conclude that this hybrid approach decreases the probable errors of the neural networks, resulting in higher accuracy of the final response, and additionally it reduces the computational burden.

In [37] a method for detecting early stages of instabilities is proposed. Online measurements are used to define the parameters of the dynamic equivalent model, which includes an aggregated generator and controllers of each area. The dynamic states of the equivalent generators are used to create a real-time transient stability index, as a measure of the distance to instability. The results show good performance of the proposed index, however it is tested only on very specific scenarios.

A different approach to transient stability assessment is presented in [38]. Instead of utilizing ML techniques, this method approaches non-linear dynamics by an analytical method, using energy functions, a specific form of Lyapunov functions. Unlike the previous methods using energy functions, in this approach an algorithm is introduced to choose the best suited function to specific contingency cases. The method is showcased on a 2-bus, a 9-bus and a 39-bus system. A significant limitation of the proposed framework is that it can not provide assessment when the system reaches a transiently unstable state following a disturbance.

Another analytical method for transient stability is explored in [39]. In this study the authors address the problem of transient stability under stochastic continuous disturbances, which are brought to modern power systems due to sources of uncertain nature, such as converter connected generators and electric vehicles. A probability measure of instability is introduced using stochastic averaging, the performance of which is superior compared to the Monte Carlo simulations, the common approach to this problem. An assumption during the evaluation process of this method is that all SGs are subjected to stochastic continuous disturbances with the same magnitude.

An early data-driven methodology for the prediction of blackouts in presented in [40]. This study introduces a temporal induction algorithm, which is based on decision trees and comprises of two steps, the definition of the tree structure, followed by the pruning of the trees, in order to reduce complexity and avoid over-fitting. The proposed method is evaluated only on the prediction of voltage collapse phenomena, however the authors emphasize on the flexibility of ML methods.

To address the issue of the interpretability of ML black-box models, Explainable AI techniques have been applied on the prediction of instabilities. The use of XAI to

identify risks related to frequency stability is presented in [41]. A gradient boosting model is used to predict the frequency status and the SHAP XAI technique is applied to explain the ML model predictions. The proposed framework is demonstrated on three European networks. According to the results, the use of XAI together with domain specific knowledge can reveal key features acting as frequency stability indicators. Another explainable method that uses SHAP is presented in [42], this time applied on small-signal stability, considering also the penetration of RES. A random forest regression model is implemented to predict the critical eigenvalues, and the SHAP method is used to identify how the input features, that represent topological metrics, affect the model prediction. The study concludes that the proposed XAI framework can provide insights about the prediction of small-signal stability by considering the network topology.

An explainable method for stability assessment is introduced in [43]. In this method, a neural network is trained to predict the voltage stability margin and the predictions are explained using Shapley values. The proposed method is demonstrated on the IEEE-39 bus network, without considering the presentation of RES. The interpretable method presented is [44] is demonstrated on the same test system, and is focused on transient stability. In this case, a deep belief network is utilized, the predictions of which are explained using a local linearization method. A tree-regularization method is investigated in [45], for the interpretability of a deep neural network that predicts transient stability. An explainable method for transient stability that considers RES penetration is introduced in [46], and utilizes a bayesian neural network and Gradient SHAP.

An intepretable method for the estimation of the Critical Clearing Time is presented in [47]. Multiple decision tree regressor models are trained using power system variables at specific network locations, capturing critical locational, physical and economic aspects. The permutation feature importance technique is then used to identify the most important variables for stability margin changes. Based on this information, targeted interventions at specific locations can be performed. More detailed insights about the same problem (Critical Clearing Time estimation), are provided in [48]. In

this paper, the authors utilize neural network models and apply the SHAP method to provide both local and global explanations.

2.3 Chapter Conclusions

As described above, various methods have been proposed so far for the investigation of cascading events appearance in power systems. A significant percentage of them focuses on static simulations, which can capture limited cascading failure mechanisms, such as line over-loading and long-term voltage instability. The effect of renewable penetration on cascading events using dynamic models has not yet been thoroughly studied, as integrating RES units into dynamic models adds significant complexity to the modelling process, which needs to accurately capture the fast dynamics and interactions between converter interfaced and conventional power system components. Dynamic simulations, although more computationally challenging than static, can describe in more detail the dynamic behaviour of the system after an initial contingency, capturing faster dynamic phenomena related to transient, frequency and voltage instability. In addition, in order to capture these cascading events and the effect of renewable penetration on system stability, it is necessary to develop dynamic models that include detailed implementations of renewable generation, protection devices and system controls to represent as realistically as possible the behaviour of modern power systems. As the uncertainty that comes with renewable penetration affects the operational states and system topology in various ways, multiple operating conditions and possible contingencies have to be considered in order to investigate different cascading event sequences that can possibly appear and compromise network security. A set of carefully selected papers was chosen to be reviewed in this Section, as each referenced work represents a strategic contribution to the methodologies and findings on the modelling of cascading events. This analysis significantly enriches the understanding of this complex phenomenon, while also revealing the contribution of this Thesis on the modelling of cascading events using dynamic models and RES penetration.

So far, methods trying to assess the online security of power systems have been focusing on identifying only early stages of transient, frequency or voltage instability.

ML methods have been widely used for addressing this problem, as the offline data provided by static or dynamic simulations can be used to train such algorithms and eventually predict the behaviour of the system in close-to-real time. As documented by the methods evaluation, ML algorithms can effectively handle the complexity and capture the non-linearity inherent in power systems, allowing for accurate and fast predictions. Compared to analytical methods, data-driven methods can capture detail to a greater extent, as analytical methods often require predefined assumptions. However, the scalability and adaptability, when faced with unseen system conditions, of ML methods needs to be further investigated. The reviewed studies focus on the prediction of early cases of instability, not accounting for the action of protection devices that can cause sequences of cascading events. Even more, methods that aim to predict the reason of the upcoming cascading events have not yet been investigated. This project aims to utilize ML algorithms and measurement data, in order to predict not only cases of instability, but the appearance and the reason of cascading events, as defined by the discrete action of protection devices.

As it can be concluded from the previous review, the modelling, and subsequent prediction, of cascading events in power systems with renewable generation is a very challenging task, because of the complex interactions between the involved mechanisms and the uncertainty in the dynamic behaviour of the system following an initial fault. Recent rapid advancements in power systems operation necessitate the development of accurate and robust methodological frameworks for the prevention of blackouts, which has not yet been achieved. The above mentioned points and identified gaps drive the main motivation for the technical research work presented in the following Chapters of this thesis.

Chapter 3

Modelling and Investigation of Cascading Events in Power Systems with Renewable Generation

3.1 Introduction

3.1.1 Motivation

As highlighted by the previous Chapters, better understanding the nature of cascading events and the dynamics involved is of significant importance in securing against and preventing blackouts. So far, in power systems dominated by synchronous generation, there have been several studies that examine cascading events. However, the effect that RES penetration has on the appearance of cascading events has not been adequately addressed. As the identification and investigation of cascading failures is a very complicated problem because of the large number of possible contingencies and their combinations, it is important to develop accurate methods and model realistic test cases, that include data about generator cost, protection systems and power system loading, amongst others, as more research towards this direction is needed [2], [4].

Identifying and investigating cascading failures is a very challenging task, since dynamics at various timescales as well as discrete actions of protection devices need to be properly represented. In addition, a large number of possible initial operating conditions (including different system loading and generation output as dictated by economic dispatch) and contingencies need to be accounted for [2], [4].

Based on the analysis in 2, the appearance of cascading events in dynamic models with RES penetration has not been thoroughly investigated, and even more how voltage or frequency related mechanisms affect these events. In [49] the importance of a model implementing the action of Load Tap Changers (LTCs) and Over-excitation Limiters (OELs) that can capture longer voltage-related instability phenomena is discussed. The LTCs change the ratio of the transformer, in order to keep the distribution voltage within limits. In some cases that will increase the power losses and require more reactive power injection. If the generators reactive power output capabilities reach the limits of the OELs, that will create reactive power imbalance in the system leading to voltage instability.

In order to realistically capture phenomena of frequency instability, it is significant to implement the main system components that take part in the power system frequency control. A sudden change in active power balance may lead to frequency deviation from the nominal value. Primary frequency control depends mainly on the droops of the synchronous machine governor, stabilizing the frequency of the system at a value which may differ from the nominal one. In order to restore the nominal frequency, secondary frequency control is used. In some systems this might require a manual intervention by the system operator or this task is performed by Automatic Generation Control (AGC). Thus, the simulation of the AGC operation [50] that considers the different dynamic characteristics of the synchronous generators is important in order to represent secondary frequency control mechanisms.

3.1.2 Contributions

This Chapter introduces a framework for the characterization and investigation of cascading events in power systems using time domain dynamic simulations. The proposed

framework aims at identifying and analysing the sequence in which these cascading events occur, the power system components involved and the reason for failure (e.g. voltage/frequency) while considering a wide range of possible operating conditions defined by economic dispatch. Changes in observed cascading failure patterns for different operating conditions are identified and investigated, taking also into consideration the impact of renewable generation. Furthermore, an investigation of the impact that the action of LTCs, a voltage related mechanism, and AGC, a frequency related mechanism, have on the way that cascading events propagate in order to highlight the importance of different modelling aspects on representing cascading failures is carried out. The framework is demonstrated on a modified version of the Anderson-Fouad 9 bus model and on a modified version of the IEEE-39 bus model, incorporating renewable generation and protection devices, capturing phenomena related to voltage, frequency and transient instability. This information could be vital in exposing network vulnerabilities and designing preventive as well as corrective measures to avoid cascading events from spreading. In more detail, the contributions of this Chapter include:

- A detailed analysis of the cascading event patterns that appear, based on the frequency of appearance, the power system components involved, and the reasons for tripping. It is also investigated how the appearance of cascading events changes for different wind penetration and system loading conditions.

- Identifying the components involved in the cascading events and their reasons for tripping from a locational aspect perspective, aiming to point out the most vulverable area of the system.

- Defining three scenarios with and without the action of LTCs and AGC and comparing the number and reason of cascading events, the average load loss and the time between consecutive events in order to identify the impact of these mechanisms. This information could be vital in identifying the importance of the inclusion of such mechanisms in dynamic modelling of cascading failures.

- Investigating how the number of cascading events and the average load loss are impacted by the system loading for the defined scenarios. In addition, the most

common cascading event patterns that appear and how these change across each scenario are identified.

3.2 Modelling Approach

3.2.1 Procedure Overview

For the initial study conducted in this chapter, a brute force approach is followed to simulate a wide range of possible operating conditions. This is performed by discretising multiple variables such as system loading and wind penetration, as described later, that also include stressed network situations. After considering the initial operating conditions for load and wind generation, an Optimal Power Flow (OPF) problem is solved in order to determine the dispatch of conventional generators and consequently the amount of disconnection of conventional generation.

Dynamic RMS simulations are performed to capture the system response to an applied contingency. In RMS models the network components are modelled in detail using differential algebraic equations considering the fundamental frequency values. The differential equations are solved iteratively in time using a pre-defined time step and known initial parameter values.

Three phase faults on lines are considered as initiating events in this study. The faults get cleared by the protection devices included in the model, and in some cases lead to a cascading event sequence involving possible failures. The cascading events are caused by tripping of components, due to intentional interventions of the protection devices after the relevant limits are violated (e.g. under-/over- voltage or frequency). The cascading events are then characterised by the component that is disconnected (capturing locational aspects), the time of disconnection (capturing the sequence of events) and the reason for disconnection (capturing the potential instability mechanism).

For the possible discretized operating conditions and contingencies that are considered, the patterns in which the cascading events occur are identified. These patterns are characterised by metrics relative to frequency of appearance, whether they lead to a viable or non-viable case, the reason of the first cascading event, the specific compo-

nents that trip along with the reason for tripping, and how these metrics relate to the amount of wind penetration and system loading. An overview of the methodology is presented in Fig. 3.1.

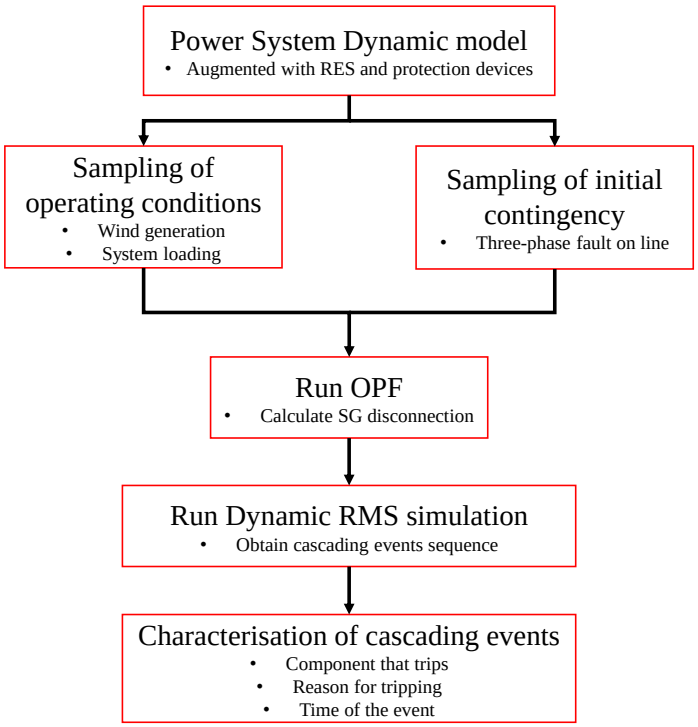

Figure 3.1: Flowchart illustrating cascading events identification procedure.

The aim of this characterization and introduction of above mentioned metrics is to identify vulnerabilities of the network, related to instability mechanisms and areas or specific components that might lead to onset of cascades for the network under study. In addition, describing a potential sequence of cascades in detail can offer insights into strengthening the network to prevent cascades from occurring or designing special protection schemes to stop them at their onset.

3.2.2 AC OPF and Conventional SG Disconnection

An AC OPF [51] is solved to determine the dispatch of the SGs using the inbuilt OPF solver function in DIgSILENT PowerFactory. The objective of the OPF problem is the minimisation of the total synchronous generation cost, while respecting constraints set

by the active and reactive power limits of the generators, the maximum loading of the lines and the bus voltage limits. The OPF problem objective is described by:

$$f = min \sum_{g \in \mathcal{G}} C_g(P_g) \tag{3.1}$$

where $C_g(P_g)$ is the individual cost function of generator $g \in G$, where G is the set of generators in the network. The network constraints are given by:

$$P_{min,g} \leq P_g \leq P_{max,g} \ , \ g \ \in \ G \tag{3.2}$$

$$Q_{min,g} \leq Q_g \leq Q_{max,g} \ , \ g \ \in \ G \tag{3.3}$$

$$p_l^2 + q_l^2 \leq S_{max,l}^2 \ , \ l \ \epsilon \ L \tag{3.4}$$

$$V_{min,j} \leq V_{g,i} \leq V_{min,i} \ , \ j \ \epsilon \ B \tag{3.5}$$

where Equation 3.2 describes the constraints set by the active power limits of the generators, Equation 3.3 describes the constraints set by the reactive power limits of the generators, Equation 3.4 describes the constraints set by the maximum loading of the lines, where L is the set of Lines in the system, and Equation 3.5 describes the constraints set by bus voltage limits, where B is the set of Buses in the system.

Each SG in the system is allocated either a high, medium, or low cost establishing a merit order among the SGs. SG1 is allocated a medium, SG2 a low and SG3 a high incremental cost. According to the study in [52], this cost allocation for this system leads to a lower Critical Clearing Time, meaning that the network is more stressed in this configuration. The wind farms operate at 100% of their rated power, which is not constant but is defined according to the wind generation output in each test case, as described in Section 3.2.4. The cost function of each SG is described by:

$$Cg(\$/hr) = c_0 + c_1 P_g + c_2 P_g^2 \tag{3.6}$$

where $c_0(\$/hr)$, $c_1(\$/MWh)$, $c_2(MW^2h)$ are the cost coefficients.

The amount of conventional SG disconnection, and consequent inertia variation in the network, as the wind and load varies, is also considered in a simple manner after the OPF solution. Each generator is assumed to have 15% additional spare capacity (headroom) on top of the operating point taken from the OPF solution, assuming this does not violate its initial nominal capacity [53]. If the resulting nominal capacity is larger than the initial nominal power of the generator (as described in [54]), then it is set to the initial nominal value. In this case, there is no room for further disconnection of conventional generation.

3.2.3 Modified version of the Anderson-Fouad 9 bus model

In order to perform a comprehensive cascading events study, there is need for detailed representation of related fast and slow dynamics, protection device operation as well as pre-fault operating conditions affected by cost of generation (in operation driven by economic dispatch), variation of system load and intermittent renewable generation [2]. For this purpose, a dynamic RMS model of a test network with high penetration of RES, including protection devices was used to identify and characterise possible cascading events. The original Anderson-Fouad 9 bus model has been modified to represent dynamic phenomena related to voltage, frequency and transient stability by implementing the actions of protection devices related to over-/under- voltage and frequency, pole slip protection and distance protection. In addition, the model has been modified with the addition of automatic voltage regulators, power system stabilizers, over-excitation limiters, wind generator controllers, as well as tap changer actions and governors to capture slower voltage related phenomena as well as primary frequency response actions.

Dynamic RMS simulations are performed in order to model all the system dynamic components in detail and accurately capture the system response to a contingency. The power system differential-algebraic equations with known initial values are solved iteratively in time, using the adaptive time step function of DIgSILENT PowerFactory, with a maximum step size of $10^{-3}s$. The power flow equations are given by:

$$P_i(t) = \sum_{n=1}^{N} |y_{i,n}|\,|V_n(t)||V_i(t)|\cos(\delta_n(t) - \delta_i(t) + \angle y_{i,n}) \tag{3.7}$$

$$Q_i(t) = -\sum_{n=1}^{N} |y_{i,n}|\,|V_n(t)||V_i(t)|\sin(\delta_n(t) - \delta_i(t) + \angle y_{i,n}) \tag{3.8}$$

where $y_{i,n}$ is minus the admittance connecting buses i and n, and $\delta_i(t)$ is the voltage phase.

Electromagnetic Transients (EMT) simulations offer more details than RMS, however both the computational effort and the modelling complexity are significantly increased. For this reason, dynamic RMS simulations are chosen for this study, as a balance between detailed modelling and computational effort. In addition, by using RMS it is possible to capture dynamic phenomena evolving over longer periods of time (up to 180s in this study).

The modified version of the Anderson-Fouad 9 bus model (Fig. 3.2) [52] developed in DIgSILENT Powerfactory [55] is used for the initial modelling considerations and results. The network nominal voltage is 230 kV, and the nominal frequency is 60Hz. The network total active power demand is 315MW, and the loads are modelled as balanced three-phase constant impedances in the RMS simulation.

The three synchronous generators (SGs) in the network are represented by full detail four winding models (6th-order). The 6-th order model is described by the following equations [56]:

The electrical model of d axis is given by:

$$\begin{cases} T'_{d0} \cdot \frac{dE'_q}{dt} = E_f - E'_q - \frac{x_d - x'_d}{x'_d - x''_d}\left(E_{q'} - E''_q\right) \\ T''_{d0} \cdot \frac{dE''_q}{dt} = E'_q - E''_q - (x'_d - x''_d)i_d + T''_{d0} \cdot \frac{dE'_q}{dt} \\ u_q = E''_q - x''_d \cdot i_d \end{cases} \tag{3.9}$$

The electrical model of q axis is given by:

$$\begin{cases} T'_{q0} \cdot \frac{dE'_d}{dt} = -E'_d - \frac{x_q - x'_q}{x'_q - x''_q}(E_{d'} - E''_d) \\ T''_{q0} \cdot \frac{dE''_d}{dt} = E'_d - E''_d + (x'_q - x''_q) i_q + T''_{q0} \cdot \frac{dE'_d}{dt} \\ u_d = E''_d - x''_q \cdot i_q \end{cases} \tag{3.10}$$

The rotor motion is given by:

$$\begin{cases} \frac{d\delta}{dt} = \omega - 1 \\ mM \cdot \frac{d\omega}{dt} = T_m - T_e - D(\omega - 1) \end{cases} \tag{3.11}$$

where E''_d,E''_q,E'_d,E'_q are the voltages behind x''_d, x''_q, x'_d, x'_q, which are the substransient and transient reactances of d/q axis, T''_{d0}, T''_{q0}, T'_{d0}, T'_{q0} are the subtransient and transient open circuit time constants of d/q axis, E_f is the field voltage, δ is the generator power angle, M is the inertia constant, D is the damping coefficient, and T_m, T_e are the mechanical and electrical torques respectively.

SG1, the reference machine of the system, is a hydro type machine equipped with a governor (GOV) rated at 247.5MVA. SG2, which is rated at 192 MVA, and SG3, rated at 128MVA, are coal type generators in voltage control mode (PV buses) equipped with IEEE DC1C Automatic Voltage Regulator (AVR), Power System Stabilizer (PSS), Governor (GOV), and Over-excitation Limiter (OEL). All generators have an operating region from 0.30 to 1p.u. active power loading. More details about the system parameters can be found in [54].

The wind generators in this study are modeled using International Electrotechnical Commission (IEC) type 4A wind turbines [57]. The installed capacity of wind generation is considered to be equal to 20% of the installed conventional generation of each area, which is 247.5MVA, 192.0MVA and 128.0MVA for area 1, 2 and 3 respectively. A windfarm is treated as an aggregate of individual 1MW operating in PQ control mode, where the total output of the windfarm is the summation of each individual turbine's active power output.

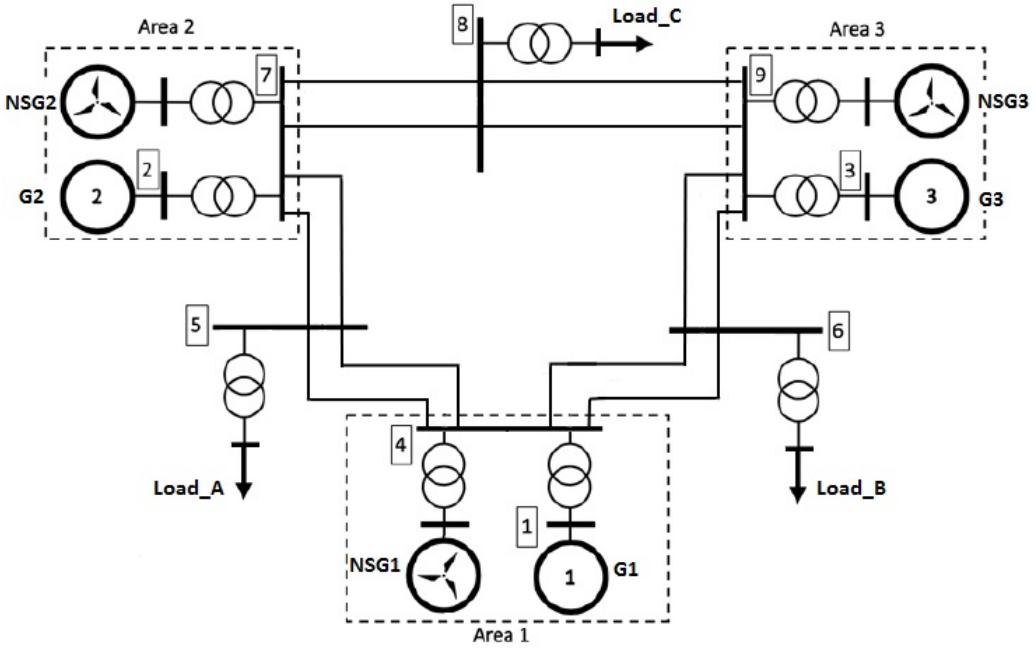

Figure 3.2: Modified version of the Anderson-Fouad 9 bus model.

Load Tap Changer

The loads are connected to distribution rating voltage buses and are modelled as balanced three-phase constant impedance loads. The P and Q values of the loads are given by:

$$P = P_0 \left(\frac{V}{V_0} \right)^2 \tag{3.12}$$

$$Q = Q_0 \left(\frac{V}{V_0} \right)^2 \tag{3.13}$$

where the subscript 0 indicates the initial operating condition.

The loads are connected to the network via step-down transformers, equipped with Load Tap Changers (LTCs). The LTCs adjust the transformer ratios keeping the distribution voltage within the deadband [0.99-1.01] p.u. When the distribution voltage leaves this deadband, the LTC acts after an intentional delay of 10s. The LTCs adjust the transformer ratios in the range [-6.5% to +6.5%] over 8 positions. For each step the ratio varies by 1.63%. The RMS model to represent the LTCs is developed in DIgSILENT PowerFactory using the inbuilt DSL language.

Protection Devices

The following protection devices have been modelled in the network:

- SGs protection: The generators are equipped with an under-/over-speed protection relay [-6%/+4%], an under-voltage protection relay with Fault-Ride Through (FRT) capabilities (ranging from 0 to 0.9 p.u. including multiple steps with different delays), and pole-slip protection.

- RES protection: The non-synchronous generators (NSGs) in the system are protected with an under-/over-voltage protection relay with FRT (ranging from 0.15 p.u. to 1.2 p.u. including various steps with different delays) and an under-/over-frequency protection relay [-6%/+4%].

- Demand protection: An Under-Frequency Loading Scheme (UFLS) with four stages was implemented for the disconnection of a percentage of demand at low frequency. (from -3% to -4% of the nominal frequency, with each stage disconnecting 10% of demand)

- Transmission line protection: Each line is protected by two distance protection relays positioned at the line ends. The relays have two zones of protection: the first one is set at 80% of the line's reach and acts instantaneously and the second one is set at 120% of the line's reach with a 400ms delay. An inter-tripping scheme between the relays has also been modelled.

All the protection devices have been implemented using standard models found in the DIgSILENT PowerFactory library. The relays settings have been adopted from the UK grid code to comply with the settings for protection devices connected in the transmission system as referred in [58].

3.2.4 Test Cases

As the modern power systems operation is dependent to a large number of uncertain parameters, it is important to take into consideration a large number of possible cases to

investigate the system response to a contingency. In this study, a brute force approach is followed, by discretizing pre-fault system operating conditions within a certain step, including stressed network conditions and applying faults to different locations of all lines in the system as initiating events. The effect of the fault location, the system loading and the amount of wind penetration is investigated. The maximum wind penetration amount per case is defined by the system loading, the operating range of the synchronous generators and the nominal capacity of the installed wind generation. The minimum active power dispatch of synchronous generators and the system loading set the limit for the maximum possible wind generation as the wind and synchronous generation output must equal the system loading and the network power losses. The SG and RES network parameters are shown in Table 3.1.

Table 3.1: SG and RES Network Parameters.

SG Machine Rating		
SG1 MVA	SG2 MVA	SG3 MVA
247.5	192.0	128.0
Minimum Active Power Dispatch (30% of Machine Rating)		
SG1 MW	SG2 MW	SG3 MW
74.25	57.6	38.4
Wind Generation Installed Capacity		
NSG1 MW	NSG2 MW	NSG3 MW
49.5	38.4	25.6

The system loading is assumed to range from 60% to 130% of the total network demand (315MW, 115MVar) as described in [54], in 10% steps. The output of each of the three wind generators in the network ranges from 0 to the maximum allowable in 10% steps. It should be noted that the wind generation output % value in all results refers to the assumed wind installed capacity as described in Section 3.2.3 e.g. 100% maximum total wind generation output means 100% out of the installed wind capacity, that is there is still synchronous generation in the network. As shown in Table 3.2, in this study 154026 cases in total were run. Three phase faults in three different locations (10%, 50%, 90%) on each Line (1-6) are considered as initiating events. That gives 18

Table 3.2: Number of Cascading Events and Patterns.

System Loading (%)	Maximum Total Wind Generation Output (%)	Number of Cases
60	17	1350
70	44	10944
80	72	21942
90	100	23958
100	100	23958
110	100	23958
120	100	23958
130	100	23958
		Total Cases
		154026

different cases for each given network operating condition.

The duration of the RMS simulations has been set to 180 s. The simulations have been performed on a standard desktop computer (CPU Intel i7-8700 3.20 GHz, 16 GB RAM) using the interface between Python and DIgSILENT PowerFactory [59]. More specifically, for each running instance, a script in Python is developed that samples the initial operating conditions, and then calls the DIgSILENT PowerFactory in engine mode to perform the OPF. Following that, the result values of the OPF are transferred to Python, where the SG disconnection is calculated. After sampling the initial contingency, through a function call the DIgSILENT PowerFactory performs the dynamic RMS simulation, and the results are extracted in .csv format. The statistical analysis of the results is then carried out using Python. The described procedure is illustrated in Fig. 3.3.

The approximate time that a single case simulation run takes is 6s. A parallel processing approach has been implemented to speed up the process, by running multiple (up to four in this study) simulated cases in parallel. It should be noted that due to the large computational effort this brute force approach might be challenging for real scale large networks. While this approach refers to planning phase where more time is available for studies, still an importance or efficient sampling technique might need to be adopted [60], [61], [62] or a screening technique as in [12]. An efficient sampling technique is employed for the definition of the initial conditions in Chapter 6.

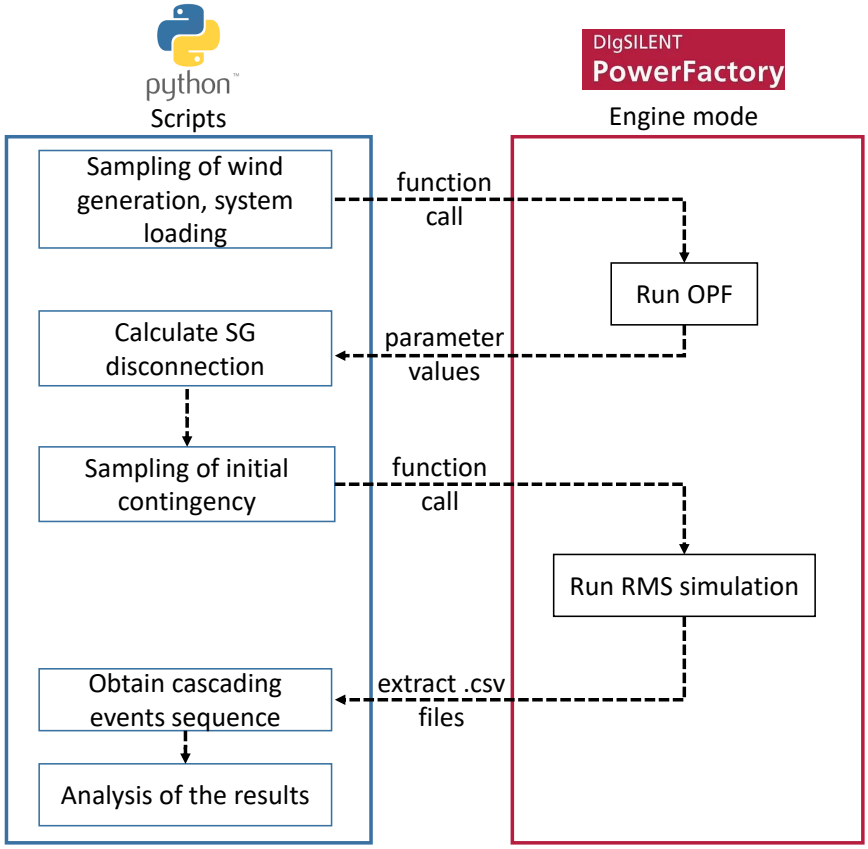

Figure 3.3: Flowchart illustrating the interface between Python and DIgSILENT PowerFactory utilized for the identification of cascading events.

3.3 Initial results on the modified version of the Anderson-Fouad 9 bus model

3.3.1 Cascading Events Characterisation

In this study, cascading events have appeared in 31250 cases, out of the 154026 that are simulated (20.3% of the simulated cases), 15031 (9.8%) of which have led to system collapse. In this study, system collapse is defined as a full power system blackout. These cascading events have appeared in 161 different patterns. Each pattern represents a unique cascading evet sequence. The cascading events are characterized by the component that trips and the reason for tripping. Also, the time of each event gives the sequence of cascading events. As shown in Table 3.3, all of the cases with cascading

events have occurred for increased system loading, at 110, 120 and 130%. When the system loading increases from 110 to 120%, 67 out of the 84 patterns that appear are new, and when the loading increases from 120 to 130%, 69 out of the 143 patterns are new. Higher system loading leads to more frequent appearance of cascading events and an increase in the number of different patterns, as the network operates closer to its limits.

Table 3.3: Number of Cases per parameter.

System Loading (%)	Number of Cases with Cascading Events	Number of Cases leading to System Collapse	Number of Cascading Event Patterns
Up to 100	0	0	0
110	1313	479	20
120	12066	4909	84
130	17871	9643	143
	Total Number of Cases with Cascading Events	Total Number of Cases leading to System Collapse	Total Number of Cascading Event Patterns
	31250	15031	161

3.3.2 Most common cascading events patterns

In Table 3.4 the most common cascading events patterns that have appeared are presented. The name of each cascading event is defined by the name of the component that trips, followed by the reason of disconnection, where "UV", "OV" correspond to under-voltage and over-voltage, "US", "OS" correspond to under-speed and over-speed respectively, and "UF" corresponds to "under-frequency" (e.g."G2-UV" means that the synchronous machine "G2" is disconnected by the under-voltage protection relay). The three most common patterns involve the disconnection of RES units only due to under-voltage and do not cause any further trips of other components. The rest of the patterns, involve the disconnection of synchronous generation due to under-voltage, that leads to a drop in the frequency of the system, which causes the disconnection of loads due to under-frequency and eventual disconnection of all the synchronous machines. Out of the 161 total patterns, 156 of them result in non-viable cases. It should

be noted that in these 156 patterns G2 is the first SG that gets disconnected. This is the machine with the lowest cost, and as a result from the solution of OPF, has the highest loading.

To highlight how the characterisation of cascading events in the suggested way can be helpful, a suggestion for this specific system is to improve voltage support close to wind farms located mainly in area 2. This can be done either in planning timescale by potentially adding devices (e.g. FACTS) or in operational timescales by ensuring reactive support from nearby generators is available. Consequently, this could help stop more serious cascading events from evolving that can lead to disconnection of loads.

Table 3.4: Most Common Cascading Events Patterns.

Times that this pattern has appeared	Cascading Events patterns
9744	NSG2-UV
6046	NSG2-UV, NSG3-UV
399	NSG3-UV
371	NSG2-UV, NSG3-UV, NSG1-UV, G2-UV, Load_C-UF, Load_B-UF, Load_A-UF, G3-US, G1-US
349	NSG2-UV, NSG3-UV, G2-UV, Load_C-UF, Load_B-UF, Load_A-UF, G1-US, NSG1-UV, G3-UV
335	NSG2-UV, NSG3-UV, NSG1-UV, G2-UV, Load_C-UF, Load_B-UF, Load_A-UF, G1-US, G3-UV

3.3.3 Wind penetration impact

The number of protection devices that trip and the reason for tripping as function of the total wind generation output for 130% system loading are presented in Fig. 3.4. As it is shown, the number of tripping protection devices due to under-frequency is higher than due to under-voltage regardless of the wind penetration. However, when the wind penetration exceeds a threshold, it seems that the number of protection relays that trip gets higher. In this case for 130% system loading, when the wind generation output increases over 90%, it has been noticed that more protection devices trip mainly due to under-frequency, and in a more frequent manner. This can be explained by the disconnection of synchronous generation and the resulting lower inertia that the

introduction of RES causes.

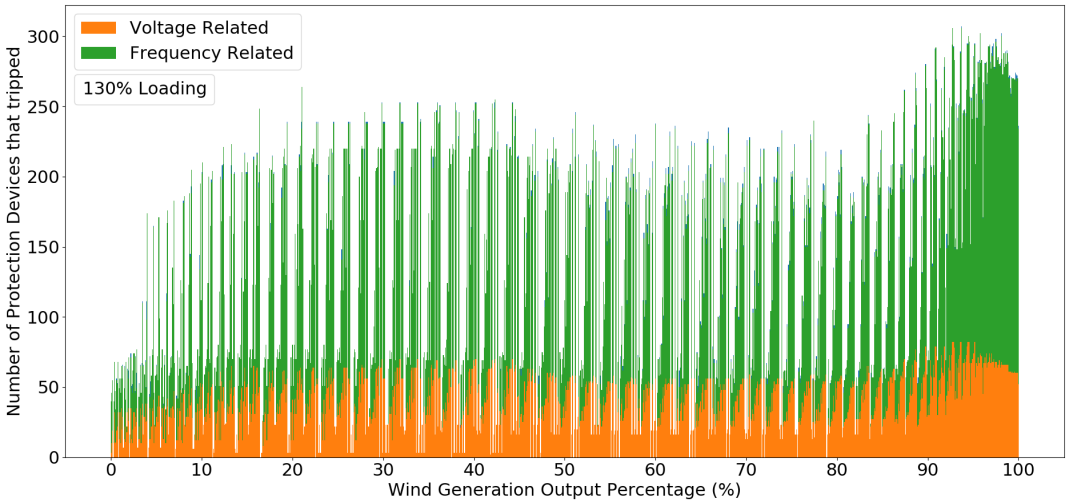

Figure 3.4: Number of protection devices that tripped and reason for tripping as function of wind penetration.

In an attempt to understand better the effect that wind penetration has on the appearance of cascading events, an investigation on how many devices trip when there is RES penetration at only one area at a time has been performed. When there is wind generation only at Area 1, no protection devices trip. In Fig. 3.5 it can be noticed that as the wind penetration at Area 2 and the system loading increase, the number of protection devices that trip gets higher. More specifically, there seems to be a threshold that changes with system loading, which when exceeded causes a significant rise in protection device trippings. For example, for this network and for area 2 this threshold is around 50% for 130% loading, 70% for 120% loading and almost 100% for 110% loading.

On the other hand, in Fig. 3.6 a very different behaviour is seen, highlighting the complexity of the system dynamic behaviour and the need for systematic characterisation of possible cascading events. When there is wind penetration only at Area 3 no protection devices trip when the system loading is 110%. When the system loading is 120% there is a trend for protection device trips to reduce as penetration increases, especially above 60%. As SG3 is being displaced by the renewable generation in Area

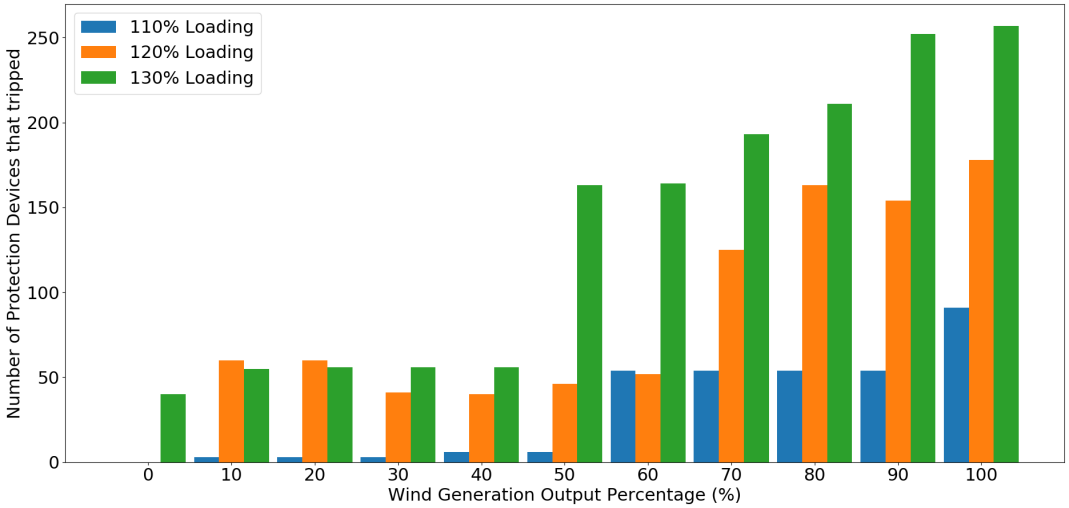

Figure 3.5: Number of protection devices that tripped with wind penetration only at Area 2.

3, the overall stability of the power system increases. It should also be noted that SG3 is the machine with the smallest nominal rating, meaning that it can reach its maximum nominal range for high system loading scenarios. For system loading 130% protection device trips are generally high and are not affected by wind penetration in area 3. Therefore the effect of wind penetration on the appearance of cascading events is difficult to be predicted, it is dependent on the specific network topology and no 'worst-case' scenario can be safely assumed.

The results (for this specific system) suggest that a clear penetration threshold where the possibility of cascading events increases cannot be defined in a straightforward manner, as the dynamic behaviour of the system regarding such events is very complex. Therefore, the resulting rules from the suggested framework should be defined in more detail (taking into consideration locational aspects as well as the specific reasons for tripping of devices) rather than as simple thresholds.

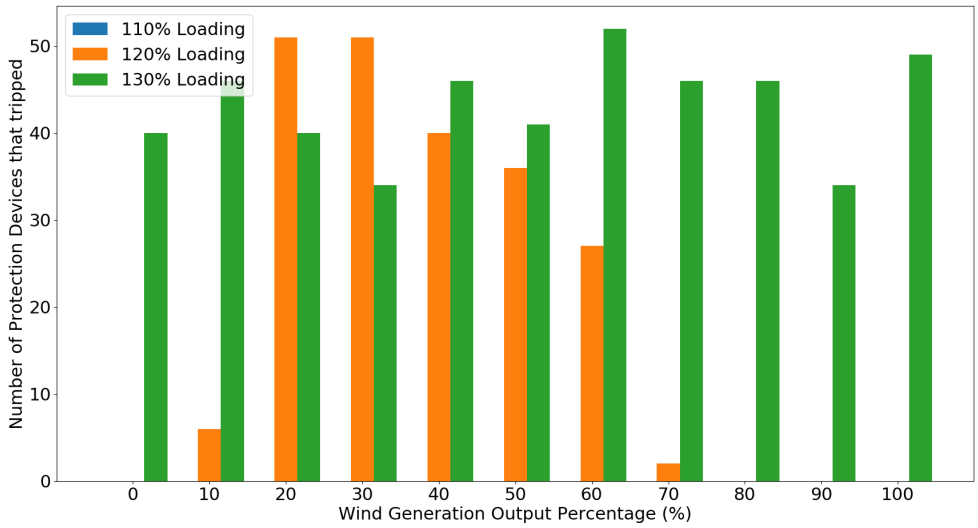

Figure 3.6: Number of protection devices that tripped with wind penetration only at Area 3.

3.3.4 Locational Aspects, components involved and reasons for tripping

The initiating fault location impact on cascading events is presented in Fig. 3.7. More frequent appearance of cascading events occurs when the fault happens on the lines that are closer to Area 2. In Fig. 3.8 the number of patterns that each component has appeared in along with the reason for tripping are presented. The synchronous machine G2 gets disconnected in all the cases due to under-voltage and appears in most patterns (156 out of 161), whereas G1 and G3 get disconnected most of the times due to under-speed (145 and 120 out of 161 respectively) and in less patterns due to under-voltage (10 and 24 out of 161). The RES units NSG1, NSG2 and NSG3 get disconnected in all the patterns that they appear in by the UV protection relays, with the NSG2 appearing in higher number of patterns (149 out of 161). It can be concluded that when the initiating fault occurs close to Area 2 it has a high impact and that components in that area appear in a high number of cascading events patterns due to disconnection by under-voltage relays.

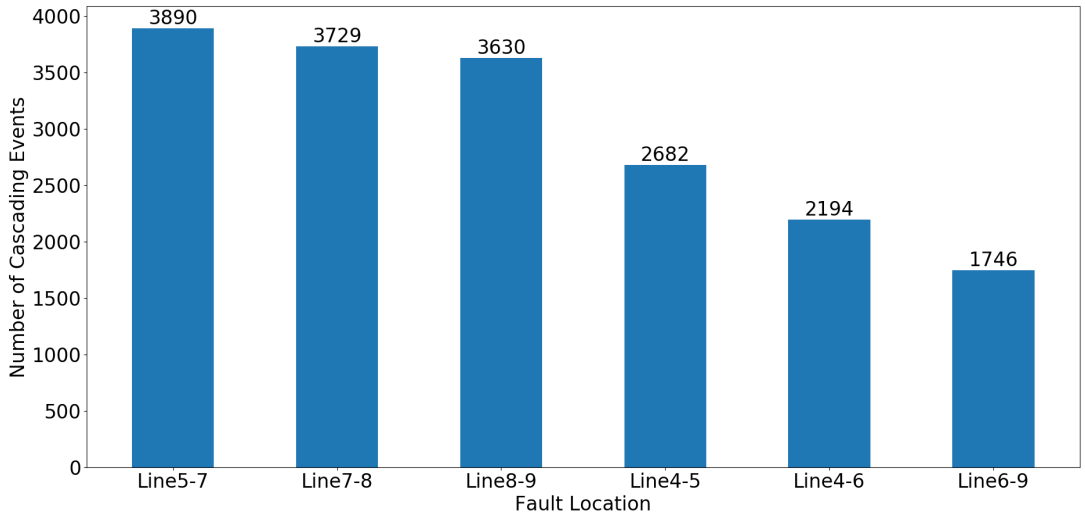

Figure 3.7: Number of cascading events according to fault location.

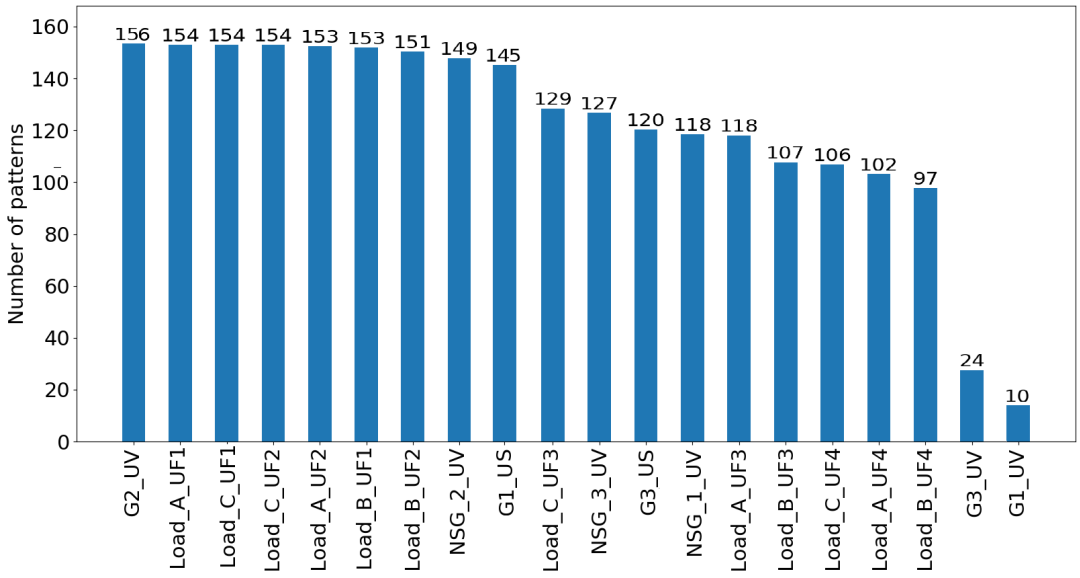

Figure 3.8: Number of patterns that each protection device has appeared in.

3.3.5 Reason for first cascading event

The first cascading event in 143 out of the 161 patterns is the disconnection of NSG2, in 17 the disconnection of NSG3 and in 1 the disconnection of G2, all due to under-voltage. The time of occurrence of the first cascade and how that is affected by system

loading is shown in Fig. 3.9. The first cascading event occurs earlier, as the system loading increases. As it can be concluded, Area 2 is the most vulnerable area of the system with voltage related issues appearing most commonly there. Area 1 appears to be the most stable area as G1 is the machine with the highest inertia constant. In this case, this information could be used in designing preventive measures to enhance voltage stability in Areas 2 and 3.

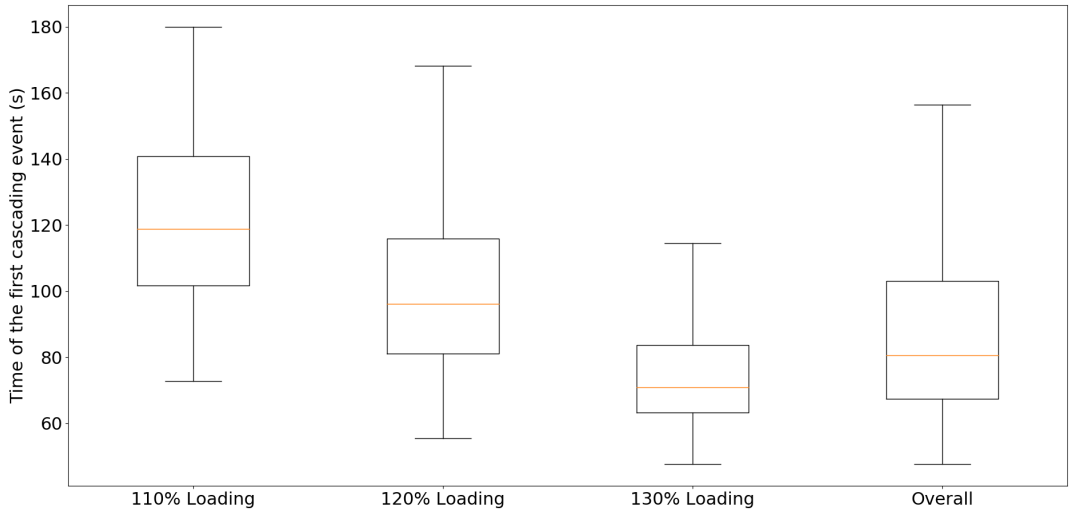

Figure 3.9: Time of the first cascading event.

3.3.6 Example – Case Leading to Cascading Events and System Collapse

An example of a cascading event that leads to system collapse is presented here. This corresponds to the most common pattern resulting in non-viable system conditions. The initial contingency (three-phase fault on 90% of Line 5-7 initiating the trip of the line) causes unacceptably low voltages at a number of buses, causing the disconnection of three NSGs by UV relays (Fig. 3.10). The first synchronous machine that gets disconnected is G2, the machine with the lowest cost curve and the highest loading. As the LTCs attempt to restore the distribution voltage (Fig. 3.11) [49], they force the generators to increase their reactive power injection and as a result their field current. The OELs limit the field current of the G2 and G3 and as G2 is the highest loading

machine with no extra capacity, it loses its capability to control voltage, leading to its terminal voltage drop and disconnection by the UV protection relay (Fig. 3.10). The imbalance between generation and demand caused by the disconnection of G2 (and the 3 NSGs previously), leads to a sudden and large drop in the system frequency (Fig. 3.12). Finally, all four steps of the UFLS get activated, disconnecting 40% of demand at each area, followed by the tripping of G3 and G1 by the under-speed relay, which results in the final system collapse. This pattern of cascading events, including the time of each event, is presented in detail in Table 3.5.

In this case, the system loading has been set at 120% and the output of the RES units NSG1, NSG2, NSG2 set at 80%, 50%, 90% respectively.

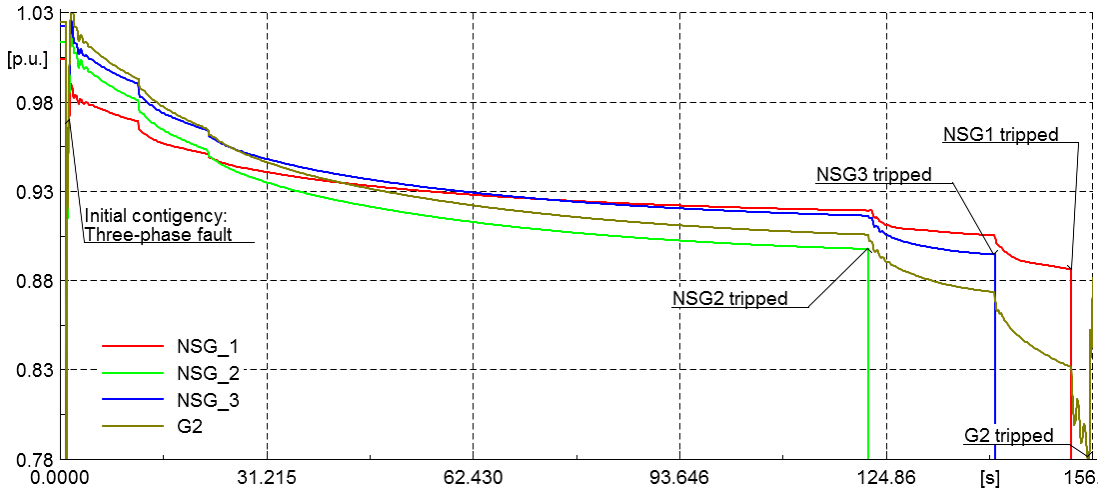

Figure 3.10: Evolution of selected bus voltages.

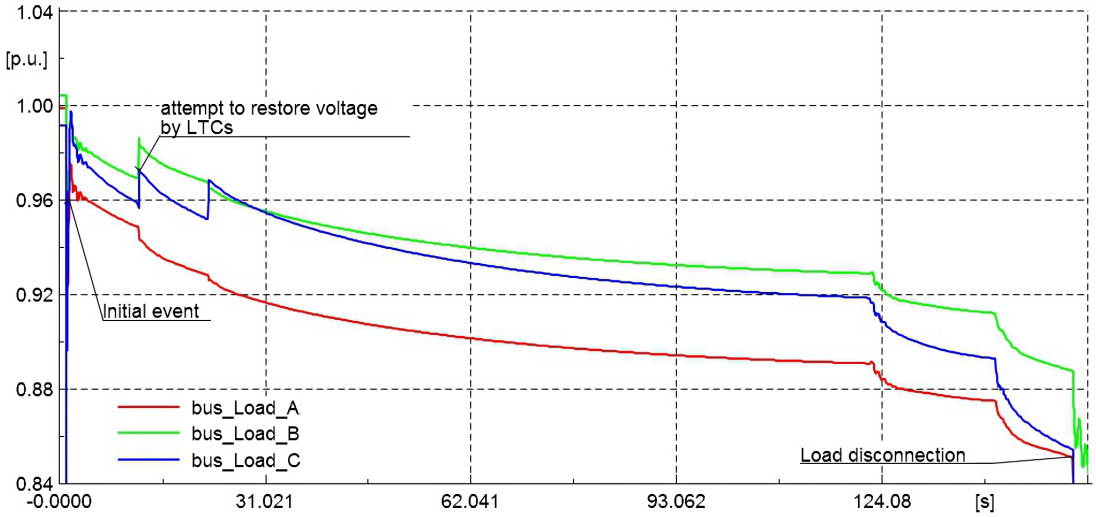

Figure 3.11: Voltage evolution and LTCs action at load buses.

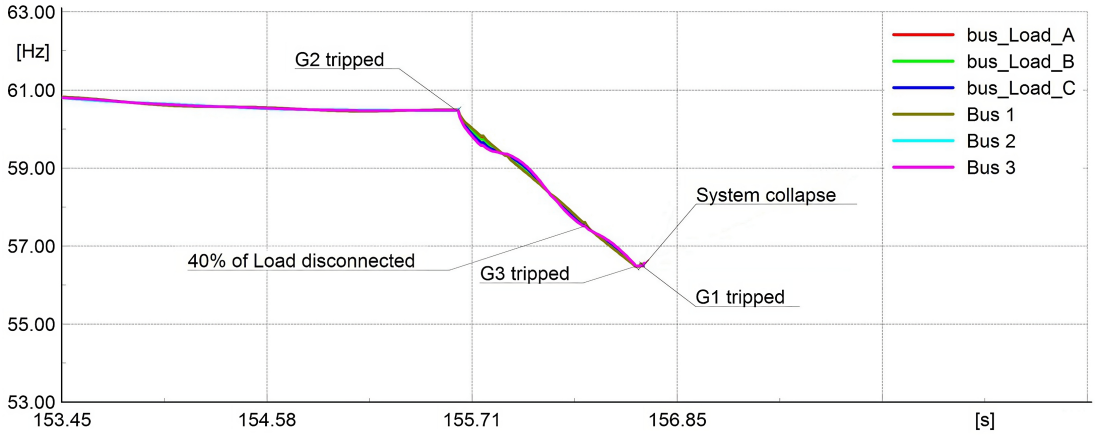

Figure 3.12: Frequency evolution of selected buses at the time of disconnection.

3.4 Investigation of the Impact of Load Tap Changers and Automatic Generation Control on Cascading Events

3.4.1 Procedure Overview

In order to identify the impact of LTCs and AGC on the appearance of cascading events, the same simulations with and without the action of these mechanisms are performed and compare the cascading event patterns that appear. In this study, dynamic RMS

Table 3.5: Overview of a Cascading Events pattern.

Time of Event (s)	Event Description
1	Three – Phase Fault on 90% of Line 5-7
1.07	Line 5-7 trips by Line Distance Protection Relays
122.26	NSG2 disconnected by UV Relay
141.30	NSG3 disconnected by UV Relay
152.80	NSG1 disconnected by UV Relay
155.61	G2 disconnected by UV Relay
156.34	40% of Load C disconnected by UF Relay
156.34	40% of Load A disconnected by UF Relay
156.34	40% of Load B disconnected by UF Relay
156.62	G3 disconnected by Under – Speed Relay
156.65	G1 disconnected by Under – Speed Relay

simulations are performed on a test system with RES penetration, including protection devices to capture the cascading events that appear after an initial applied contingency. An initial screening of a wide range of possible operating conditions is performed to identify which set of pre-fault operating conditions and contingencies lead to the appearance of cascading events. For the same operating conditions, the simulations are performed again, defining a scenario in which the LTCs are not considered and a scenario enabling AGC, to investigate how this impacts the cascading event sequences that appear.

Wind generation output and system loading values are discretized within a certain step, taking into account also stressed network conditions. Following the sampling of wind generation and system loading values, an AC OPF [51] problem is solved to determine the dispatch and the amount of disconnection of the SGs, as described in Section 3.2.2. Each SG is allocated a cost curve, as in [63]. In this case study, each SG is assumed to consist of 4 identical machines which can be set on or out of service. According to operating point of each SG, given by the solution obtained from the OPF, the number of machines for each generator that are needed and assumed to be connected is calculated.

In this study, three phase faults on lines are considered as an initial contingency. The fault occurs at t=1s and gets cleared by disconnecting the faulted line after 70ms. It should be noted that the initiating fault and line disconnection on the IEEE-39 bus

model can lead to cases where parts of the system become islanded due to the set-up of the test network. The cascading event sequences are consequently identified, describing how the events evolve in the system. The base case consists of the system modelled as described in detail Section 3.4.2 with LTCs activated and no AGC implemented. For the operating conditions that after the initial contingency (i.e. a three-phase fault causing the disconnection of a line) lead to at least one additional component to trip for the base case, two scenarios are defined. In the first scenario, the LTCs are not modelled and the simulations are repeated. In the second scenario, AGC is implemented in addition to the base scenario (i.e. with LTCs modelled). For each scenario the number and reason of cascading events, the average load loss and the time of events are presented in a comparative analysis.

3.4.2 Modified version of the IEEE-39 bus model

A modified version of the IEEE-39 bus model, as shown in Fig. 3.13, is used in this study to assess the impact of LTCs and AGC. The system is implemented using RMS simulations in DIgSILENT PowerFactory. The ten synchronous generators (SGs) in the network are represented by full detail four winding models (6th-order). equipped with Automatic Voltage Regulator (AVR), Power System Stabilizer (PSS), Governor (GOV), and Over-excitation Limiter (OEL). The network parameters can be found in more detail in [64]. The three wind parks in this study are connected to three different locations and are represented using International Electrotechnical Commission (IEC) type 4A wind turbines [57]. The installed capacity of wind generation is considered to be equal to 20% of the total installed conventional generation of the IEEE-39 bus system base case.

The protection devices as described in Section 3.2.3 without the distance protection, are implemented in the IEEE-39 bus system too. The distance protection scheme on the IEEE-39 bus system is later on added in Chapter 6.

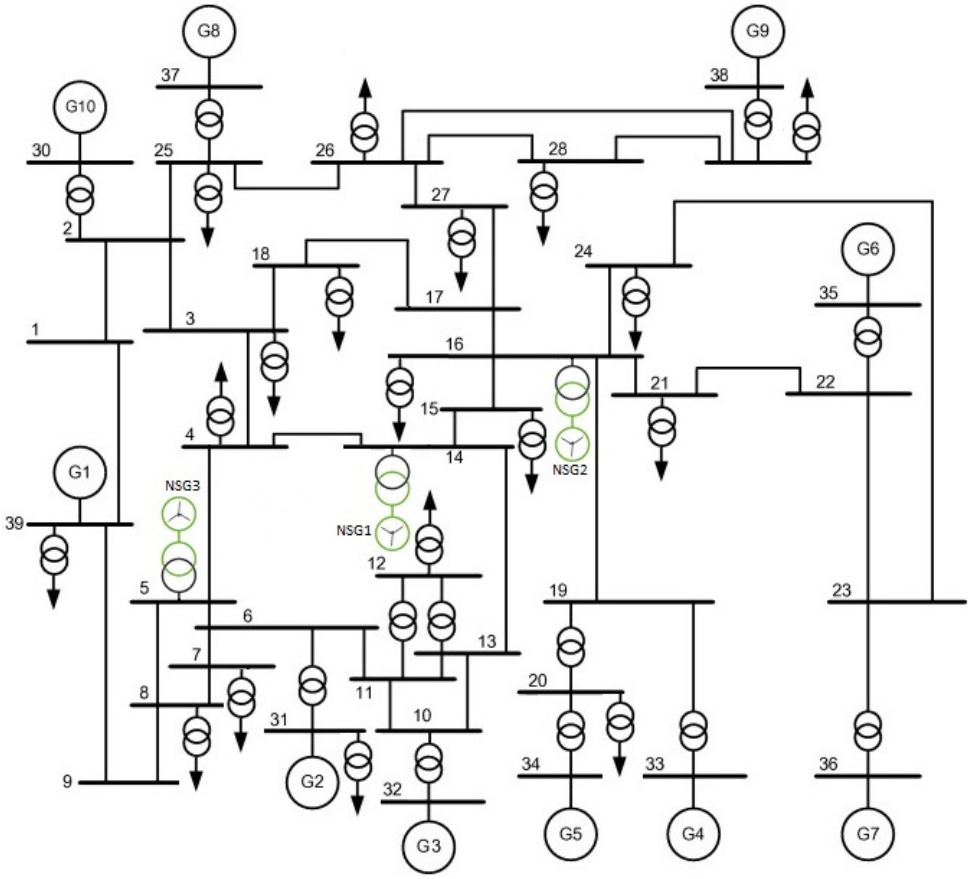

Figure 3.13: Modified version of the IEEE-39 bus model.

Load Tap Changers (LTCs)

The loads are modelled as balanced three-phase constant impedance loads and are connected to distribution voltage rated buses via step-down transformers. These transformers are equipped with LTCs, which adjust the transformer ratios keeping the distribution voltage within the deadband [0.99-1.01] p.u. The LTCs adjust the transformer ratios in the range [-10% to +10%] over 33 positions (0.625% ratio variation per step) [49] and act with an intentional delay of 10s. Without considering the action of LTCs, the distribution voltage may not be kept within this deadband. The step-down transformers tap positions will keep their initial value, as defined by the OPF solution. In some cases, not changing the tap positions can preserve the voltage stability of the network and prevent load shedding events [65]. On the other hand, as the loads are

voltage dependent a change in the voltage will change the power absorbed, which may in turn affect frequency.

Automatic Generation Control (AGC) in Power Systems

In order to realistically capture phenomena of frequency instability, it is significant to implement the main system components that take part in the power system frequency control. The power system frequency depends on the active power equilibrium of generation and demand. A sudden change in the power demand or generation leads to active power imbalance and consequently to the frequency deviation from the nominal value. The frequency deviation after the primary frequency control depends mainly on the droops of the synchronous machine governors. So, the primary frequency control can stabilize the frequency of the system at a value which differs from the nominal one. In order to restore the nominal frequency the secondary frequency control is used. The frequency is controlled by the generation power change that allows to reinstate the active power balance. Thus, a realistic frequency control requires the simulation of the AGC operation [66], that considers the different dynamic characteristics of the synchronous generators which are part of the frequency control.

A simplified AGC dynamic model is used to simulate the secondary frequency control in the network. The real-time system frequency measurement, obtained from a PLL at the bus connecting the generator to the network, is compared to the nominal reference frequency. Each generator acts independently, according to the local frequency measurement. The deviation of frequency is processed through a PI controller that transmits a signal to the governor adjusting the generator active power accordingly (Fig. 3.14) [67]. The gains of the PI controller adjust the active power rate of change. It is considered that synchronous generator G10, which represents a hydropower generator, a technology appropriate for fast acting varying power injection, contributes to the AGC scheme of the network. The action time of the AGC is 30s after the start of the simulation.

So, it is expected that in the scenario including the action of AGC, it will affect the cases in which frequency related trips appear after the first 30s in the simulation and it

will help to enhance the system frequency control and result in less trips and less load loss from the UFLS protection scheme.

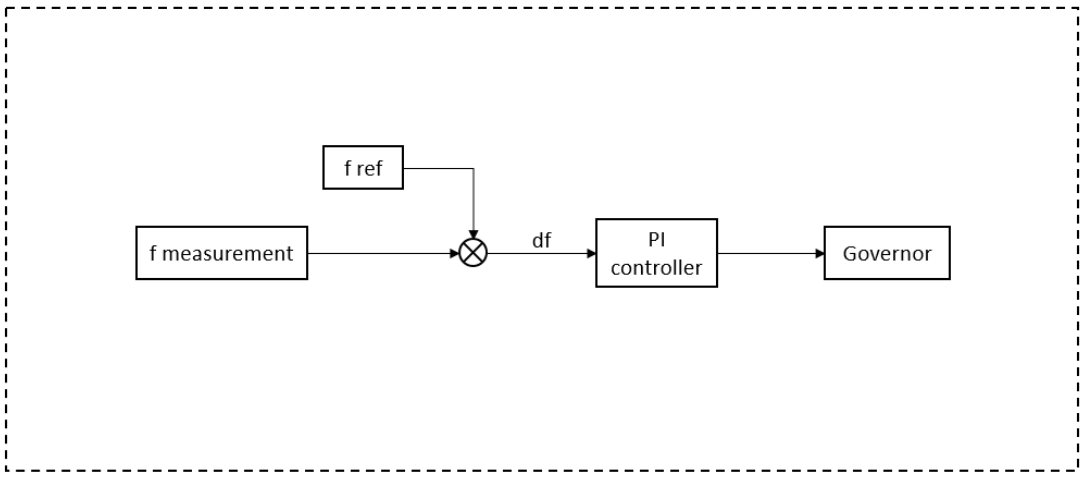

Figure 3.14: AGC Model Structure.

Example of the AGC operation

In this RMS example, a loss of load at t=1s is simulated to showcase the AGC response. This loss of load creates an imbalance between generation and demand, increasing the system frequency. In Fig. 3.15.a) without the use of AGC the governors action stabilize the frequency of the system at a new value, different than the nominal value (60.074Hz). As it is highlighted in Fig. 3.15.b), the use of AGC restores the frequency at the nominal value (60Hz). The secondary frequency response gets activated at t=30s.

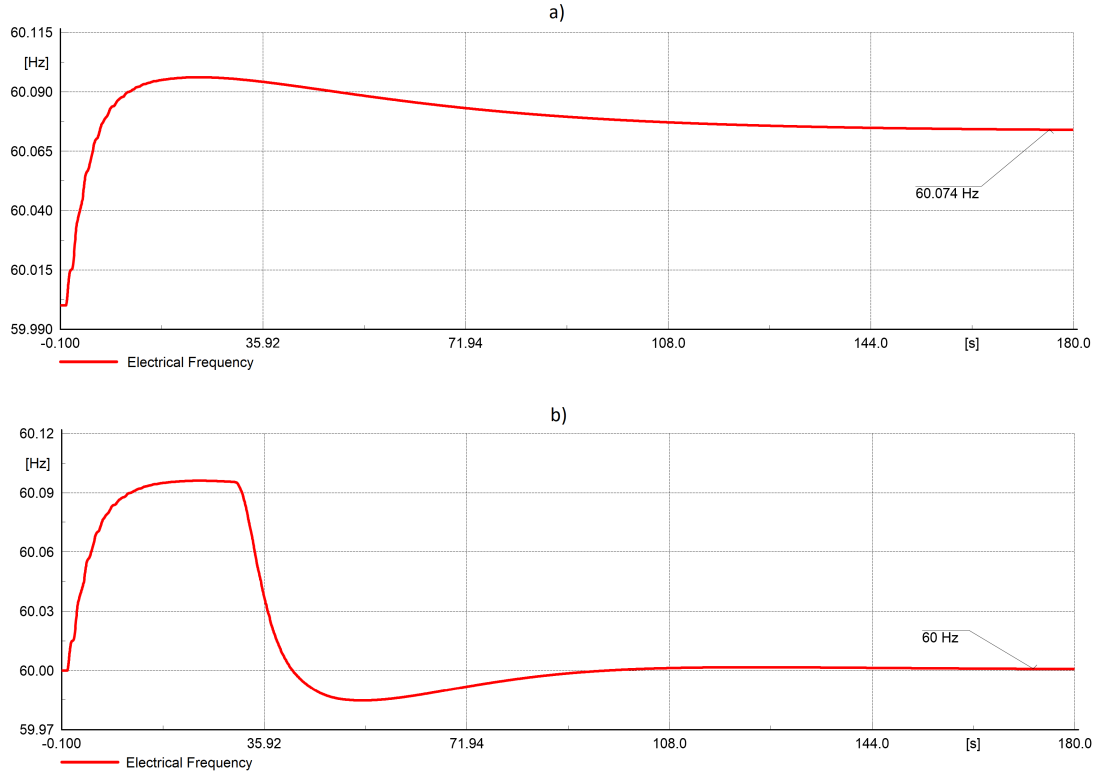

Figure 3.15: a) Frequency evolution without AGC. b) Frequency evolution with AGC.

3.4.3 Test Cases

In this study for the IEEE-39 bus model, the initial operating conditions, wind penetration and system loading, are sampled in a deterministic way, by discretizing these values within a certain step. The system loading is assumed to range from 70% to 120% of the total network demand (as calculated in the base case) in 10% steps. For each loading scenario, the wind generation output also varies. The output of each of the three wind generators in the network ranges from 0 to the 100% (of the nominal capacity of each wind generator) in 20% steps. For each system loading and wind generation output operating condition the amount of synchronous generation in the network is defined by the AC OPF solution.

In this study 44064 cases in total were simulated. Three phase faults in the middle (50% length) of each line are considered as initiating events. That gives 34 different

cases for each given network operating condition, multiplied by 6 different loading scenarios and by 36 different RES output scenarios (6 output scenarios for each one of the three WGs). For the cases that cascading events have appeared, the simulations are performed again for the same initial operating conditions to investigate the effect of a voltage related and a frequency related mechanism on the cascading event sequences that have appeared. For this reason, the following scenarios have been defined:

- Base case: LTCs on and AGC off
- Scenario I: LTCs out of service.
- Scenario II: LTCs on and AGC implemented.

The duration of the RMS simulations has been set to 120s. The interface between Python and DIgSILENT PowerFactory has been used to set up the dynamic simulations running multiple simulated cases in parallel in order to speed up the process.

3.5 Results of the impact of LTCs and AGC on the appearance of Cascading Events on the modified version of the IEEE-39 bus model

3.5.1 Scenario I: Not including Load Tap Changers

In the base scenario, cascading events appeared in 7131 cases, out of the 44064 that are simulated (16.2% of the simulations). In this Scenario, the 7131 cases where cascading events have been observed are simulated again, setting the LTCs out of service and comparing the results to the base scenario. In Scenario I, 71922 cascading events (i.e. counting as event every activation of a protection device) in total have appeared, a higher number than in the original scenario (69175), in 2271 different sequences. As it is shown in Fig. 3.16 fewer cases with only one event have appeared. Approximately the same number of sequences with 10-40 events appear for both cases. A reduction in the number of sequences with more than about 20 events is observed for both scenarios.

When the LTCs are not in action more sequences with a high number of events (70-80) have occurred.

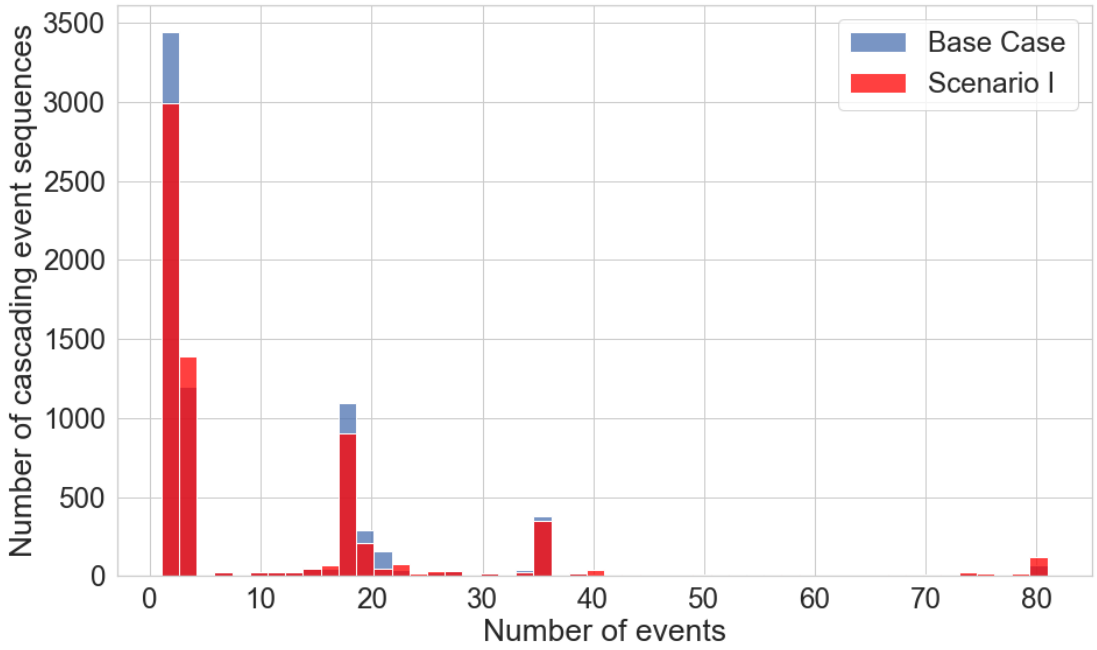

Figure 3.16: Number of cascading events per sequence for Scenario I.

The time of the events can also provide vital information regarding the evolution of cascading events. As it is shown in Fig. 3.17 during the early stages of the simulation (0-20s) more cascading events appear when the LTCs are not included. During the later stages (60s-120s) very few events are captured in Scenario I and fewer when compared to the base case, i.e. fewer events tend to appear in the later stages of the cascade. Both the mean value (0.95s) and the standard deviation (6.62s) of the time between cascading events in Scenario I are smaller compared to the base scenario (2.12s and 10.45s respectively) suggesting that events happen in quicker succession as highlighted in Table 3.6. This potentially leaves shorter time window for corrective measures that could be taken to prevent cascading events from spreading.

The UFLS scheme implemented in the model can disconnect an amount of the system load when the network frequency is low in order to restore the active power balance. For each sequence, the percentage of load loss is calculated, by dividing the amount of load that is disconnected to the total system loading at this case. In total

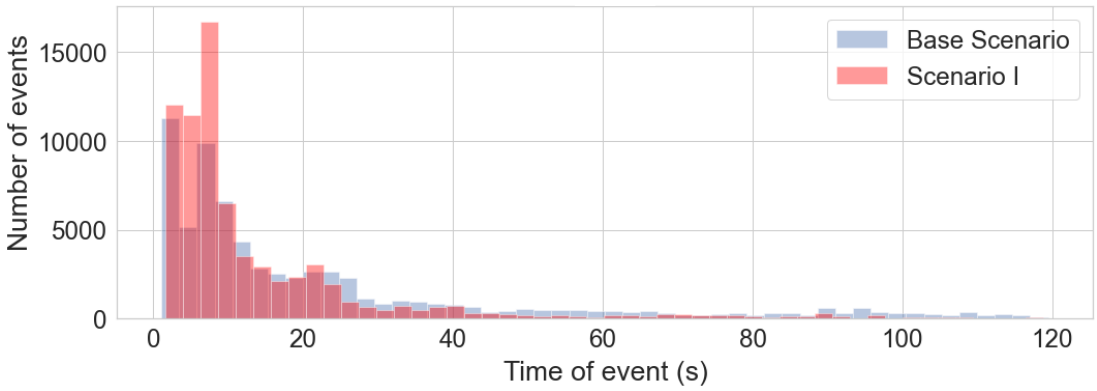

Figure 3.17: Time of events for Scenario I.

there are more load trips in Scenario I (57309) than in the base scenario (55711) and the total amount of load disconnection is higher. Fewer cases where observed in Scenario I when compared to the base scenario for load loss around 1% and around 7%. However, as highlighted in Fig. 3.18 in Scenario I more sequences have a large amount of load loss (30%), resulting in a larger total amount of load that gets disconnected. These additional sequences that resulted in a load loss of 30% in Scenario I, resulted in a load loss of 0% or 5-15% in the base scenario.

Table 3.6: Number and time of cascading events.

	Base Case	Scenario I	Scenario II
Number of sequences	2546	2271	2390
Total number of events	69175	71922	68236
Load events	55711	57309	54698
Mean/ Std Deviation time between events (s)	2.12/ 10.45	0.95/ 5.62	1.61/ 8.61

3.5.2 Scenario II: Impact of Activating Automatic Generation Control

The 7131 cases in which cascading events have occurred are simulated again, this time enabling the AGC implementation that provides secondary frequency response. The AGC model gets activated at t=30s. In the base Scenario, cascading events after the first 30s have appeared in 1597 cases, so the effect of AGC is expected to be seen in

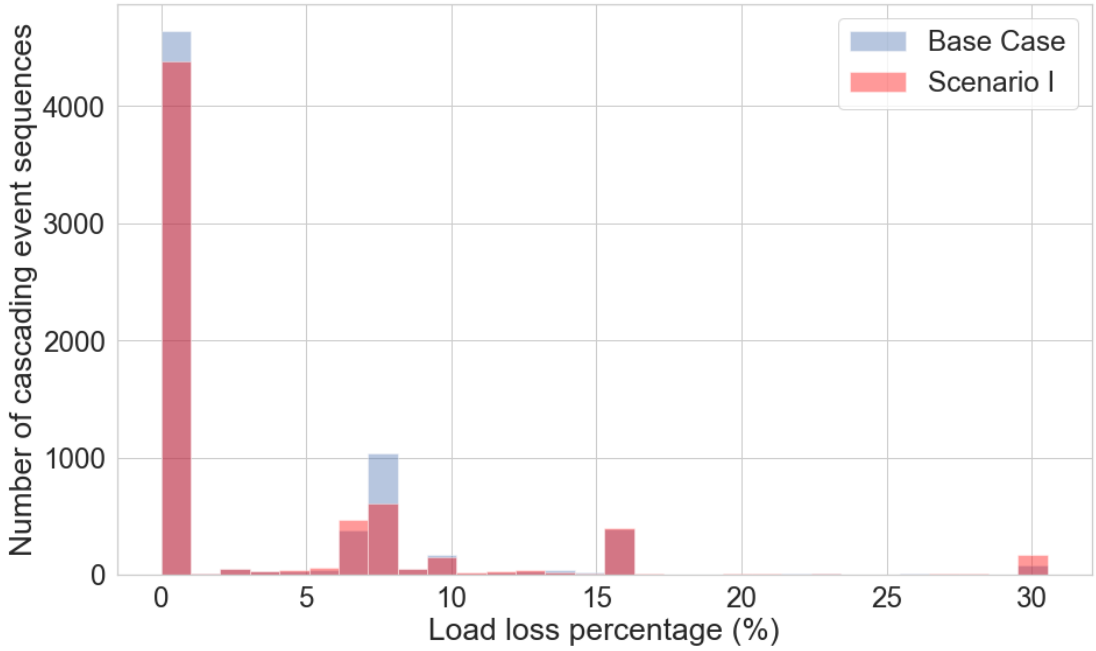

Figure 3.18: Load loss percentage for Scenario I.

these cases. The total number of trips for this scenario (68236) is smaller than in the base Scenario (69175). As it is highlighted in Fig. 3.19 the number of sequences with 10-40 events are fewer with the AGC.

The time of the cascading events is presented in Fig. 3.20. Until t=30s the time of the events for both scenarios remains the same, as the AGC gets activated after this point. In general, with the action of AGC fewer cascades appear during the earlier steps (30s-40s) and the last steps (80s-120s) of the simulation. More cascades appear compared to the original scenario at approximately t=50s. As the number of trips is in total smaller it can be concluded that secondary frequency response reduces the number of trips during the latter steps of the simulation. The mean value of the time between consecutive events, as shown in Table 3.6, is slightly shorter than in the original scenario and with a smaller standard deviation.

In this scenario fewer cascading events (54698) are caused by the disconnection of loads, and the total amount of load loss is also reduced compared to the original one. As shown in Fig. 3.21 there are fewer sequences that cause a 5-10% load loss. This can be attributed to a reduced number of under-frequency trips.

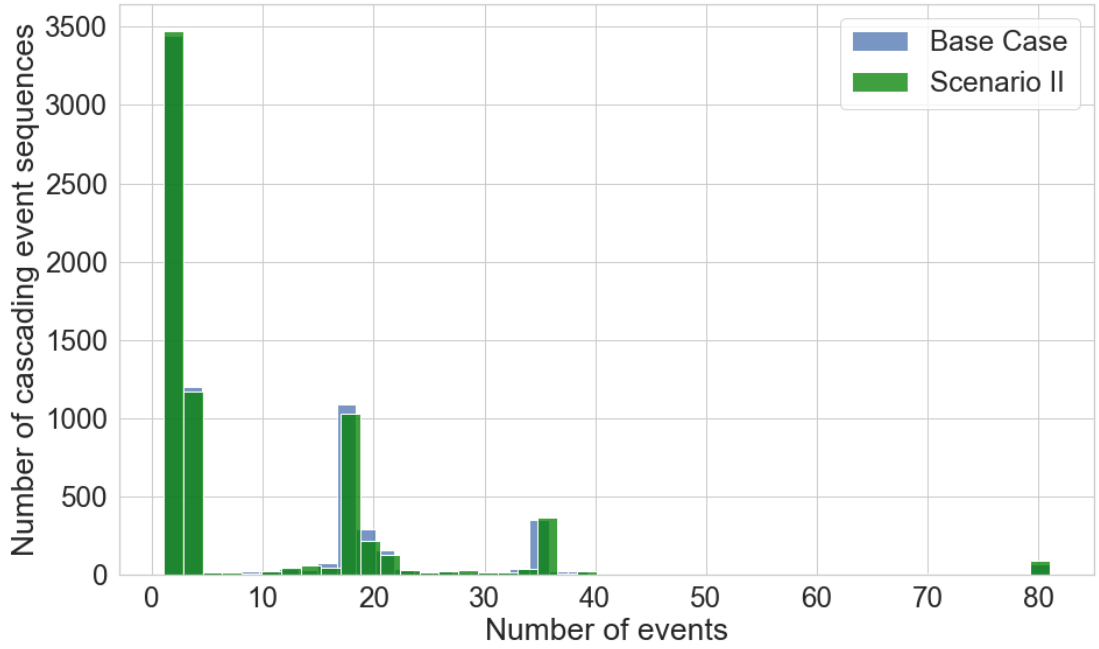

Figure 3.19: Number of cascading events per sequence for Scenario II.

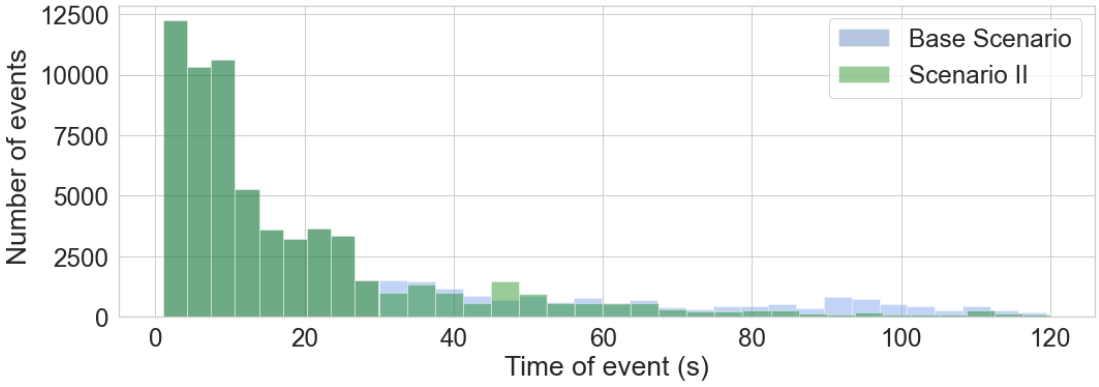

Figure 3.20: Time of events for Scenario II.

3.5.3 Reason for tripping and System Loading impact

In order to further investigate the impact of LTCs and AGC, the reason for tripping of the cascading events is displayed in Fig. 3.22. In Scenario I, slightly more trips due to voltage appear, and an increased number of trips due to frequency and transient instability. When there are no LTCs there are 6858 trips that involve the disconnection of SGs, while in the base scenario there are 6006 SG related trips. This increased num-

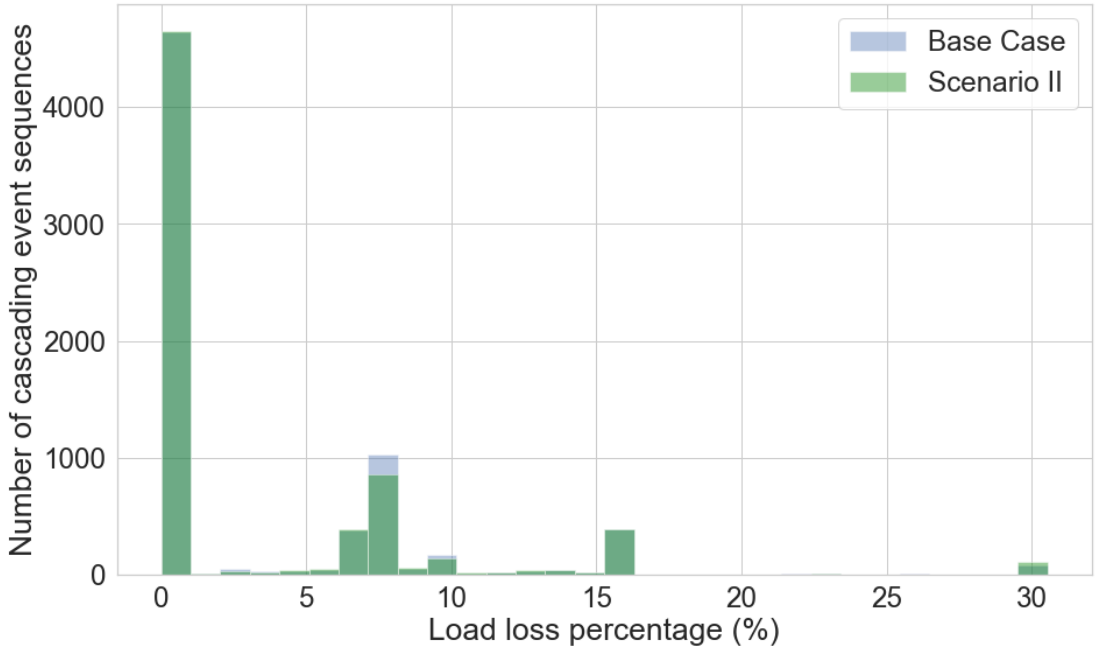

Figure 3.21: Load loss percentage for Scenario II.

ber of synchronous generation trips (852 more trips) leads to more cases of frequency instability, causing the disconnection of loads due to the UFLS scheme.

As expected in Scenario II, fewer trips occur due to under-frequency and under-speed. As the primary frequency response relies on the action of governors to restore the active power balance, the frequency may stabilise at a different value from the nominal. A further cascading event can deteriorate the frequency of the system and cause frequency related trips. The secondary frequency response, provided in this case by AGC, leads the SG through the governor to stabilise the frequency of the system to the nominal one, returning the power system operation to its secure state and reducing the number of frequency instability phenomena. It should also be noted that the number of trips due to voltage and pole slip in Scenario II (8236 and 1530 events, respectively) does not seem to be significantly affected compared to the base scenario (8261 and 1445 events, respectively). A slight reduction is however observed.

The number of cascading events across the system loading and the mean value of load loss for all scenarios are presented in Fig. 3.23.a) and Fig. 3.23.b) respectively. In general, it appears that most cascading events appear at the system loading nominal

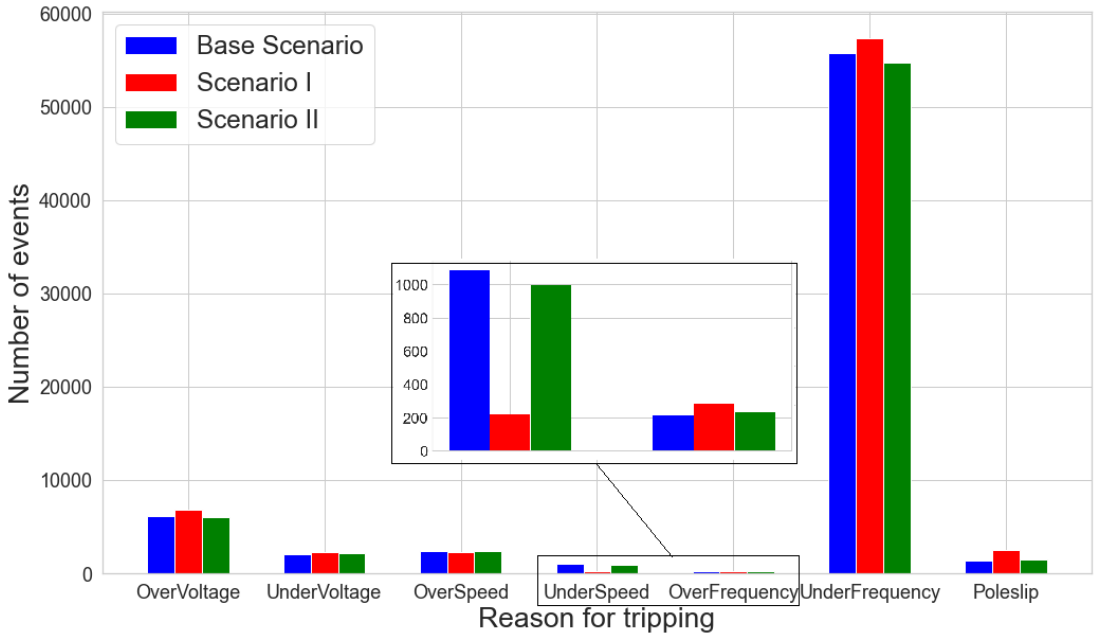

Figure 3.22: Reason for tripping for all Scenarios.

value (100%) for all three scenarios, however the cases with the most impact (higher load loss) appear at high system loading (110%-120%). In Scenario I, when the loading value is low (70%-80%) and at the nominal value more cascading events appear than in the original scenario and with a higher load loss. This increased number of trips (6585 events) appears due to more Under-Voltage, Under-Frequency and Pole-Slip events. On the other hand, when the system loading is at 90% and at 110% fewer cascades (1494 events) are captured when there are no LTCs, which are caused due to Under-Voltage and Under-Frequency. At 120% system loading, it is observed that although in Scenario I the number of events is slightly lower, the sequences of events have a higher impact, as a larger amount of load trips due to the UFLS scheme when there are no LTCs. Therefore, for this particular system and case study, the impact of including LTCs can have both a positive and negative impact in the evolution of cascades due to the complex interactions and dynamic phenomena.

In Scenario II, a reduced number of cascading events and reduced resulting load loss is observed across all system loading values, due to fewer Under-Frequency trips as expected from the action of AGC. Consequently, the use of AGC has a positive impact

on the appearance of cascades in all cases.

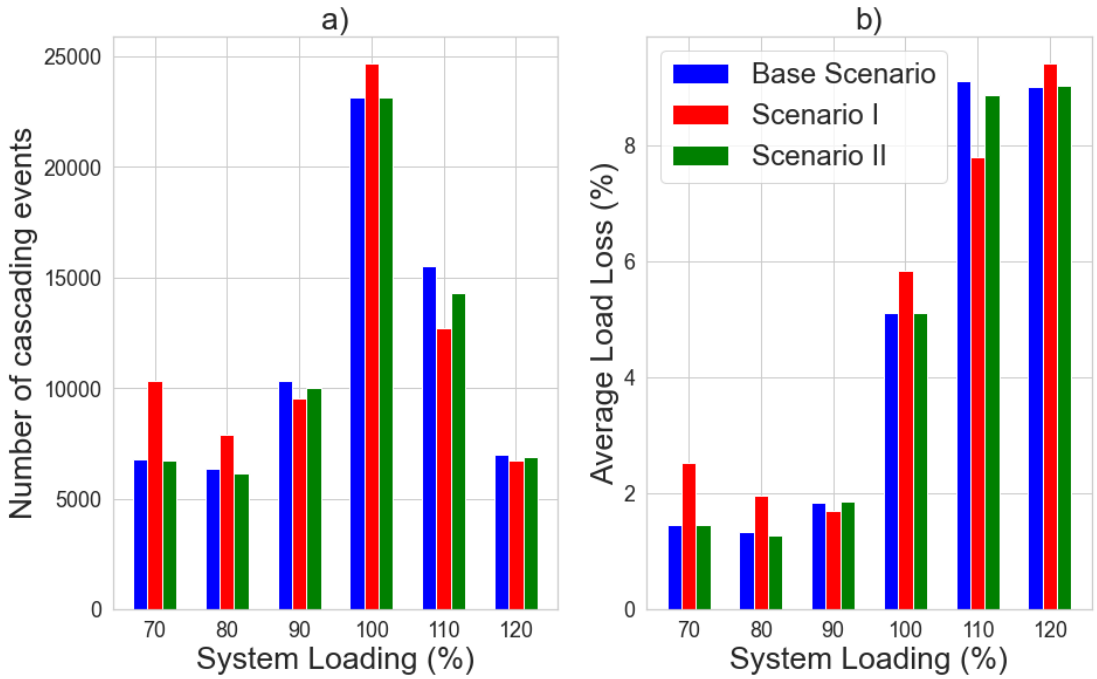

Figure 3.23: Number of cascading events and average load loss according to System Loading.

3.5.4 Cascading Events Patterns Comparison

In the base scenario, the cascading events have appeared in 2546 different sequences, with some of them appearing multiple times. In Scenario I, 2271 sequences have appeared and in Scenario II, 2390, so in both scenarios fewer sequences have appeared. In Table 3.7. the most common cascading event sequences are presented, sorted by times of appearance in the base scenario, and how many times these sequences appear in Scenarios I and II. A unique 'Id' number is assigned to each sequence in order to distinguish them. Each event is described by the component that trips and the reason for tripping. The time of each cascading event gives the sequence in which the events occur. For instance, the second most common sequence [(NSG-2-OverVoltage), (G01-Poleslip)] describes a sequence where the first event is the disconnection of wind generator NSG-2 due to Over-Voltage, followed by the disconnection of synchronous generator G01 due to pole-slip.

Table 3.7: Most Common Cascading Events Patterns

Times this pattern has appeared			Id	Cascading Event patterns
Base Case	Sc. I	Sc. II		
2219	1759	2221	1	[(NSG_2-OverVoltage)]
336	393	362	2	[(NSG_2-OverVoltage), (G01-Poleslip)]
279	403	292	3	[(NSG_2-OverVoltage), (NSG_1-OverVoltage), (G01-Poleslip)]
243	0	243	4	[(NSG_2-OverVoltage), (G01-UnderSpeed), (NSG_3-UnderVoltage)]
196	238	196	5	[(G05-OverSpeed)]

The most common sequence appearing in all the scenarios includes only a single event, the disconnection of NSG-2 due to Over-Voltage. This event is also the first cascading event in the most common cascading sequences. In Scenario I, sequence '1' appears fewer times than in the base case, and sequences '2' and '3', which include the disconnection of NSG-2 followed by other events appear more frequently. Sequence '5' appears more times in Scenario I as well. The disconnection of G05 due to Over-speed is a common event because when the initial fault is applied on Line 16-19 a part of the system becomes islanded. It should be noted that sequence '4', which includes the disconnection of G01 due to under-speed has not appeared in Scenario I. In this scenario, sequence [(NSG-2-OverVoltage), (G01-Poleslip), (NSG-3-UnderVoltage)] has appeared 249 times, indicating that the tripping of G01 after NSG-2 happens most commonly due to poleslip instead of under-speed. In Scenario II the most common patterns appear in general in the same manner as in the base scenario. Sequences '2' and '3' appear more times in Scenario II, as in some cases of the base scenario the disconnection of G01 in these sequences causes further cascading events due to under-frequency, which are prevented in Scenario II by the use of AGC.

3.6 Chapter Conclusions

In this chapter a framework for the characterisation of cascading events in power systems with renewable generation is proposed. It employs time domain dynamic simulations including protection device modelling and investigation of various operating

conditions and contingencies to identify cascading events patterns. The specific components that trip along with the reason for tripping are analysed. The impact of changes in system loading, wind generation output and synchronous generation disconnection (following an OPF solution) on cascading event patterns is systematically analysed. This information could be vital in understanding network vulnerabilities in terms of weak areas and instability mechanisms and attempt to avoid cascading events from spreading. Additionally, the effect of LTCs, a voltage related mechanism, and AGC, a frequency related one, on the evolution of cascading events in power systems with RES penetration is investigated. The main aim is to suggest a way to identify the importance and highlight potential impact from different level of modelling detail, i.e. including, or not, certain mechanisms. The impact of LTCs and AGC on the dynamic behaviour of the network is analysed by investigating the number and reason of cascading events, the average load loss, the time between consecutive events and the most common cascading event patterns. This investigation can provide significant information about how the different mechanisms affect the power system reaction to a contingency.

The proposed framework is initially applied on a modified version of the Anderson-Fouad 9 bus model, including RES units. The results highlight the most vulnerable area of the system and the reason for most cascading events. A number of possible cascades for the given network have been identified and characterized and changes in the patterns with system loading and wind penetration have been investigated. The results highlight the complexity of system dynamics as the impact of wind penetration in different areas and at different loading can drastically affect the potential cascades that might appear in the system in very different ways. The impact of LTCs and GC on cascading events is showcased on a modified version of the IEEE-39 bus model. The results from this specific test network showcase that not including LTCs in the network model results in more frequent appearance of cascading events and greater amount of load shedding. In the scenario that AGC is included, the frequency stability in the network is enhanced and a reduced number of load disconnection events due to under-frequency is observed.

Chapter 4

Explainable predictions of the appearance of cascading events using Initial Operating Conditions and Machine Learning

4.1 Introduction

4.1.1 Motivation

As it has become evident, the increasing penetration of RES at different locations of a power system makes the system response to a contingency unpredictable. In cases when network conditions activate protection devices, cascading events might appear, the propagation of which can lead to large scale blackouts, with severe consequences on society. As discussed in Chapter 2, ML approaches that focus on the prediction of cases of instability can have a significant impact on improving power system security and providing information to system operators about taking mitigation measures. However, many of the ML models are considered as black-box models, with their internal workings

to be difficult to be understood, especially by non-experts in this field.

This lack of understanding related to ML models can potentially lead to a lack of trust in such applications in power systems, negatively affecting the introduction of ML based models in practical scenarios. In industrial applications, it is not uncommon for stakeholders to prefer models that have poorer performance, but are interpretable, over models with higher performance that their internal workings can not be understood [68]. In critical domains where the actions of a ML model can affect society, such as the operation of power systems, there is a need for trustworthy and accountable ML models. Based on the above points, it is critical for system operators to be able to understand on which grounds the ML model makes a prediction, and how the input features affect the model output. That way, they can trust more this model's decisions and the range of applications can be further extended.

To address this issue, Explainable Artificial Intelligence (XAI) techniques have been developed. The field of XAI attempts to develop methods that can provide explanations about the decision-making process of ML models. As a result, XAI techniques can help to overcome the trade-off between interpretability and performance, by providing ways to explain the decision making process of accurate but complex black-box models [69].

For the aforementioned reasons, recent research studies have been focusing on the potential of XAI applications for power systems [70]. Howevern, as the analysis in Chapter 2 has revealed, the methods that utilize XAI techniques focus on stability assessment, without considering the action of protection devices. So far, no XAI applications for the prediction of cascading events have been identified.

4.1.2 Contribution

The main contribution of this chapter is the use of supervised ML algorithms and the SHAP XAI technique to predict the appearance of cascading events using initial operating conditions and explain the predictions of these models. The key difference from similar approaches is that in this case the explainability mechanisms are applied on the problem of cascading events prediction. The dataset consists of dynamic simulations that are performed on a test network with renewable penetration and protection devices,

that capture the appearance of cascading events and not just cases of instability. More specifically, the main contributions of this Chapter include:

- A ML-based method for the prediction of the probability of cascading events, knowing only the initial operating conditions during steady-state available from PMU measurements, which is formulated as a regression problem.
- A ML-based method for the prediction of the appearance of cascading events in a particular simulation, considering also information about the line on which the initial contingency appears, added to the input data of the model as a categorical feature. In this case, the problem is formulated as a binary classification task (predicting if a cascading event will appear or not).
- The identification of the the most important features when predicting cascading events by applying a feature importance technique using the permutation method to the best performing pre-trained ML model.
- The SHAP method, an XAI technique, is utilized to provide more detailed explanations on how the model reaches a certain prediction given the input data, aiming to improve the interpretability of the utilized model and enhance the trust of system operators on ML model applications for the prediction of cascading events.

4.2 Methodology

4.2.1 Detailed Procedure

The method presented in this chapter utilizes various ML algorithms for the prediction of cascading events and consists of two different problem formulations. Initially, the prediction of the probability of cascading events is formulated as a regression problem, and the input data consist of the initial operating conditions, as measured during the steady-state of the network, before the application of the initial fault. The dataset consists of the measurements gathered from the IEEE-39 network simulations as detailed in Chapter 3. Next, the network line on which the initial fault happens is added as a

discreet event to the input data, and the problem is formulated as a binary classification problem, meaning that the output of the model is if in this specific case a cascading event will appear or not. In this study, a cascading event is considered a deterministic event, resulting from the disconnection of a network component due to the intentional action of protection devices.

A variety of linear and non-linear ML algorithms and ensemble methods are tested and validated to determine the model that achieves the best performance. After training, the best performing model is evaluated on the unseen test dataset. To better understand the model decision-making process, a permutation feature importance analysis is performed to identify the features that have the highest impact on the model prediction. Finally, a SHAP analysis on the pre-trained model identifies the contribution that each feature has on the prediction of cascading events, providing explanations about both individual predictions, as well as the whole set of predictions on the test dataset. The main steps of the described procedure are showcased in Fig. 4.1.

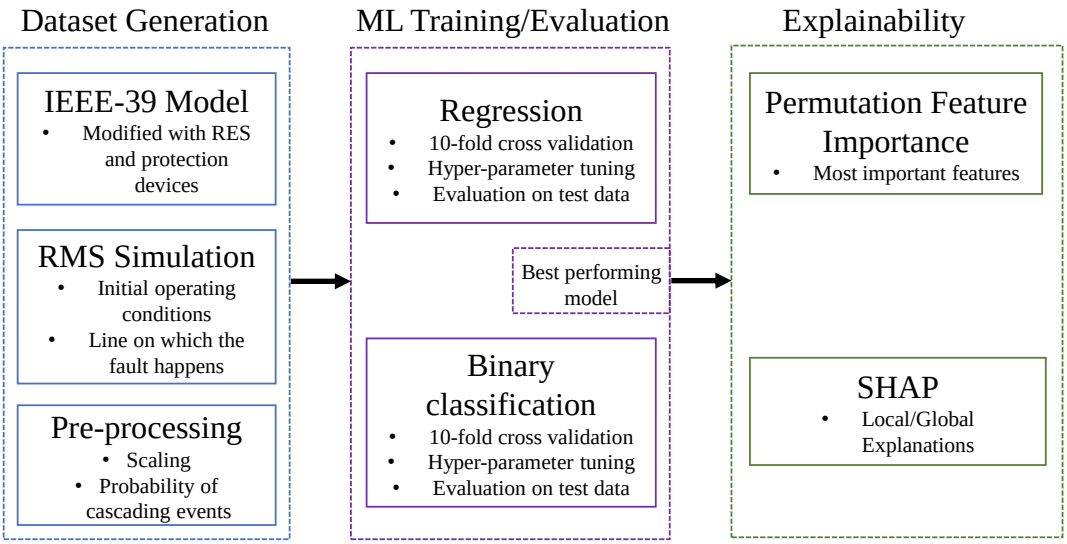

Figure 4.1: Flowchart highlighting the main steps of the procedure.

4.2.2 Machine Learning methods

Machine learning is a method for developing models that can learn from data, without being analytically programmed. Supervised machine learning is when the model is

provided with labeled data for training, and the goal is to make predictions about new, unseen data, known also as test data. These predictions are based on the patterns that the model extracts from the training data. The labeled data consist of input-output pairs, where the input is a set of features, and the output is a label or a target variable. In regression tasks the goal is to predict a continuous value variable for a given input, where in classification tasks the goal is to predict a categorical label for a given input.

According to the "No Free Lunch Theorem" [71] in machine learning, there is no single universally best algorithm for all problems. Machine learning algorithms have varying levels of complexity, making it challenging to predict beforehand which one suits the problem of predicting cascading events better. For this reason, a wide range of linear, non-linear and tree-based supervised machine learning algorithms are trained and evaluated, in order to identify the algorithm that showcases the best performance. The machine learning algorithms that have been chosen are suitable for tabular data and can be applied to either regression or classification tasks.

The following machine learning methods have been investigated and applied in this Chapter:

Linear Regression

Linear Regression is a statistical method used for modelling the relationship between a dependent variable and one or more independent variables [72]. This method is based on the assumption that the relationship between the variables is linear. In a linear regression model, the output value is a continuous variable and it is modeled as a linear combination of the input variables, with the coefficients representing the strength of the relationship between each independent variable and the dependent variable. The predicted value $\hat{y}$ of the model is represented as:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p \tag{4.1}$$

The coefficients $w_1, ..., w_p$ are estimated using a method such as least squares, which minimizes the difference between the real values and the values predicted by the model. The minimization problem is described by:

$$\min_{w} ||xw - y||_2^2 \tag{4.2}$$

where $x = (x_1, ..., x_p)$ and $w = (w_1, ..., w_p)$.

Lasso Regression

Lasso Regression (LR) is a type of linear regression that adds a regularization term to the model to prevent overfitting, by adding a penalty term to the loss function of the model [73]. The $L1$ regularization term is a penalty term that is proportional to the absolute value of the coefficients. This results in some coefficients of the models being equal to zero. With the added $L1$ regularization term Equation 4.2 gets transformed into:

$$\min_{w} \frac{1}{2n_{\text{samples}}} ||xw - y||_2^2 + \alpha ||w||_1 \tag{4.3}$$

where a is a constant value and $||w||_1$ is the $L1$-norm of the coefficient vector.

Elastic Net Regression

Elastic Net (EN) is another type of regression, that combines the $L1$ regularization term of Lasso regression, with the $L2$ regularization term. $L2$ regularization adds a penalty term to the loss function that is proportional to the square of the coefficients. This combination of regularization terms helps Elastic Net to prevent overfitting and handle more efficiently correlated variables in the dataset. Adding the $L2$ regularization term to Equation 4.3 gives:

$$\min_{w} \frac{1}{2n_{\text{samples}}} ||xw - y||_2^2 + \alpha\rho ||w||_1 + \frac{\alpha(1-\rho)}{2} ||w||_2^2 \tag{4.4}$$

where $||w||_2^2$ is the $L2$-norm of the coefficient vector and ρ a parameter that controls the convex combination of $L1$ and $L2$.

Logistic Regression

Logistic Regression (LR) is a similar statistical method to linear regression, only in this case it is applied to classification tasks. The logistic function maps the linear combination of the independent variables to a probability in the range of $[0, 1]$. The algorithm uses the logistic function to model a binary dependent variable based on the independent variables. The probability of the positive class is mathematically described by:

$$\hat{p}(x) = \frac{1}{1 + \exp(-xw - w_0)} \tag{4.5}$$

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a method that can be used for classification tasks. It is a linear method used for finding a linear combination of features that separates two or more classes [74]. This is achieved by projecting a dataset into a lower-dimensional space while maximizing the separation between classes. The dimensionality reduction results in a new set of features that can be used for classification. The class conditional distribution of the data for each class k is can be calculated by:

$$\log P(y = k|x) = -\frac{1}{2}(x - \mu_k)^t \Sigma^{-1}(x - \mu_k) + \log P(y = k) + Cst \tag{4.6}$$

where Cst is a constant term that corresponds to $P(x)$, in addition to other Gaussian constant terms.

Naive Bayes

Naive Bayes (NB) is a probabilistic algorithm which can be used for classification tasks. This method uses the Bayes Theorem in order to calculate the probability of a given input to belong to a particular class, assuming that all the input features are independent of each other [75]. The algorithm calculates the probabilities for each class and selects the one with the highest probability as the prediction. The probability

density of x given a class C_k can be calculated by:

$$P(x \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right) \tag{4.7}$$

K-nearest neighbors algorithm

K-nearest neighbors algorithm (KNN) is a non-parametric supervised learning method [76] which can be used for classification or regression problems. For classification problems, a class label is assigned based on the majority vote. This means that a data point is classified by a plurality vote of its neighbors, with the data point being assigned to the class that is most common among its k nearest neighbors. The parameter k is a positive integer, typically a small and odd number to avoid ties in classification problems. In order to determine which data points are closest to a given data point, the distance between the given point and the other data points is calculated, most commonly by using the Euclidean distance. For points p, q in the n-dimensional Euclidean space, the Euclidean distance is given by:

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + ... + (p_n-q_n)^2} \tag{4.8}$$

Decision Trees

Classification and Regression Trees (CART) is a term introduced in [77] that refers to Decision Tree algorithms that can be used for classification or regression problems. The representation of the CART model is a binary tree, where each root node represents a single input variable and a split point on that variable. The leaf nodes of the tree represent an output variable which is used to make a prediction. To determine the split points, the values are lined up and different split points are tried and tested using a cost function. The split that achieves the lowest cost is selected. All input features and all possible split points are evaluated and chosen in a greedy manner. This means that the optimal split point is chosen each time. A stopping procedure is needed to stop splitting, as the splitting procedure moves its way down the tree. A common stopping procedure is to use a minimum count on the number of training instances assigned to

each leaf node. If the count is less than the minimum count then the node is taken as a final leaf node.

Mathematically, the partitioning of data Q_m with n_m samples at node m into left and right datasets is described by:

$$Q_m^{left}(\theta) = \{(x, y)|x_j \leq t_m\} \tag{4.9}$$

$$Q_m^{right}(\theta) = Q_m \setminus Q_m^{left}(\theta) \tag{4.10}$$

where $\theta = (j, t_m)$ is a candidate split of a feature j and threshold t_m.

The quality of the candidate split of node m is computed by minimizing the impurity recursively until the maximum depth is reached ($n_m < \min_{samples}$):

$$\theta^* = \text{argmin}_\theta \left(\frac{n_m^{left}}{n_m} loss(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} loss(Q_m^{right}(\theta)) \right) \tag{4.11}$$

Support Vector Machines

Support vector machines (SVM) [78] is a supervised ML method that can be applied to classification or regression problems. When applied on regression problems, the algorithm is also mentioned as Support vector regression (SVR). The objective of the SVM algorithm is to find a hyperplane in a N-dimensional space, where N is the number of features, that classifies the data points. This can be represented as a separating line between two data classes in classification tasks. In regressions tasks, this line is used to predict the continuous output. In order to find a hyperplane in the higher dimensional space, kernel functions are used. The kernel functions are mathematical functions that can take the input data and transform it to a higher dimensional space. Support vectors are the data points that are closer to the hyperplane. These are the points that influence the orientation and position of the hyperplane, and help to construct the SVM model. The output of an SVM model is given by:

$$\hat{y} = \sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{4.12}$$

where α_i are the double coefficients and $K(x_i, x)$ is the kernel.

Ensemble Methods

Ensemble methods are ML models that construct a model from more than one base models, like decision trees. The final result is a combination of the individual models outputs. Boosting is a technique that involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models classified incorrectly [79]. This is achieved by adapting the distribution of the training data at each iteration so that the next model instance is more focused on the data that is hard to classify. The prediction $\hat{y}$ is given by:

$$\hat{y} = \sum_{m=1}^{M} h_m(x) \tag{4.13}$$

where h_m are the individual estimators, also called weak learners, and M is the number of estimators.

A gradient boosting algorithm is built in a greedy way, by minimizing the sum of losses of the previous ensemble F_{m-1}. This is mathematically described by:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{4.14}$$

$$h_m = \arg\min_h loss(y_i, F_{m-1}(x) + h(x)), \tag{4.15}$$

AdaBoost (Adaptive Boosting) is a simple boosting algorithm that assigns a weight to each training sample and adjusts the weight at each iteration. A more advanced boosting algorithm is Gradient Boosting, that uses gradient descent to minimize the loss function and improve the accuracy of the model. XGBoost (Extreme Gradient Boosting) is an optimized version of gradient boosting that utilizes a more regularized formalization algorithm to control over-fitting [80].

Bagging, also known as bootstrap aggregation, is an ensemble method that builds multiple decision trees by re-sampling from training data with replacement, and then

voting the trees for a majority prediction [81].

Random Forests (RF) and Extra trees (Extremely Randomized Trees) are ML models related to bagging. The main difference is that Random forests sub-sample from the input data with replacement [82], while Extra Trees use the whole data sample [83]. Another difference is that regarding the selection of split points for nodes, Random Forest chooses the optimal split point depending on the selected feature subset, while Extra Trees chooses the split point randomly. Once the split points are selected, both algorithms choose the best split point between all the subset of features, so Extra trees introduce some randomization but while still keeping the optimization principle.

These ensemble techniques can be used for either regression or classification type problems. For regression problems, the final prediction is obtained by averaging the predictions of all the consisting models and for classification problems the final prediction is obtained by taking the majority vote.

It should be noted that deep learning methods have not been considered in this specific chapter, as recent studies have shown that classical ML methods achieve more easily good predictions, with less computational cost and hyper-parameter tuning, than deep learning methods on tasks with tabular data [84], [85]. Tabular data refer to data that are structured within a table and each sample is represented by a row, and its features are represented among columns.

4.2.3 k-fold cross validation

The considered ML models in this study are trained and evaluated using a stratified k-fold cross validation. In this approach, the whole dataset is split into a training and a testing dataset. The training set is split into k smaller sets, with each set containing the same percentage of safe cases and cases with cascading events. For each of the k folds, the model is trained using $k-1$ folds, and validated on the the remaining 1 fold using a suitable evaluation metric, depending on the task. This process is performed k times for each set, with a different part of the data used for training and validation each time. This method is suitable when limited data are available, and helps to validate the performance of the model on different data splits. In addition, it can be used to

optimize the model hyper-parameters. The final model evaluation is performed on the unseen test data. A schematic showcasing the method is presented in Fig. 4.2.

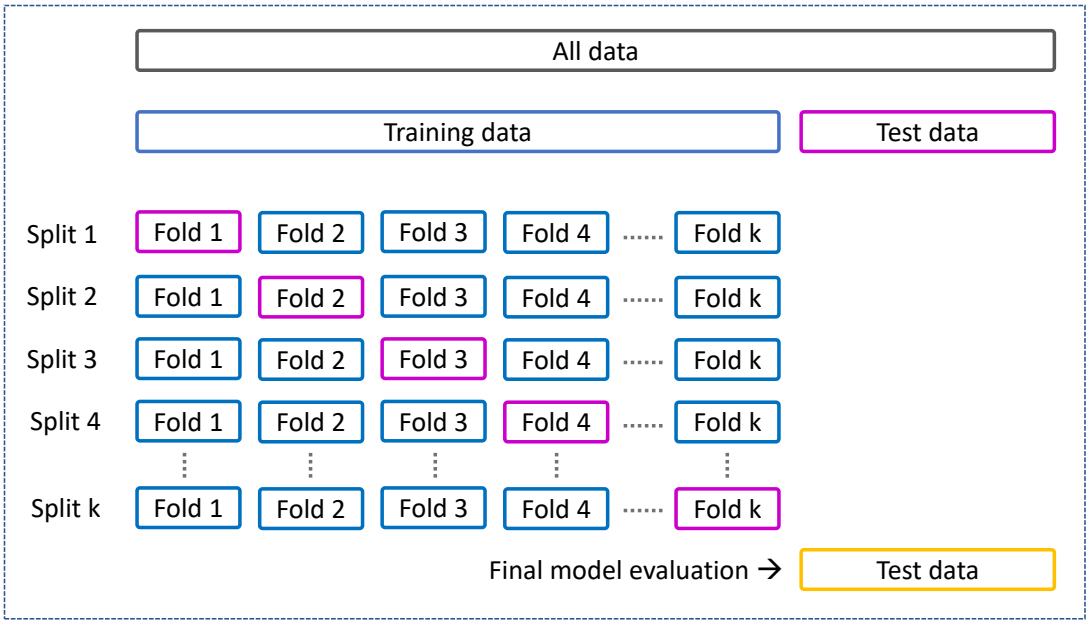

Figure 4.2: Schematic of the k-cross fold validation.

4.2.4 Evaluation Metrics

Evaluation Metrics for the regression models

Mean Absolute Error (MAE) is a metric commonly used to evaluate the performance of models trained for a regression task. This metric expresses the average of the absolute errors, that measure the absolute difference between the predicted and the real value. MAE uses the same scale as the data being measured, and is given by:

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{4.16}$$

Root Mean Squared Error (RMSE) is another commonly used metric for the evaluation of regression models. This metric emphasizes more on larger errors than smaller ones because the calculation involves taking the square root of the mean of the squared differences between the predicted values and the actual values. RMSE is given by:

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.17}$$

Coefficient of Determination (R^2) is a statistical measure used in regression analysis that represents how well the regression line fits the observed data. R^2 takes values in the range of [0,1], with higher values indicating a better fit of the model to the data, and is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y_i})} \tag{4.18}$$

In addition to these metrics, two more metrics that describe the maximum errors, Maximum Over-Estimate (MOE) and Maximum Under-Estimate (MUE), are calculated. An over-estimation of the probability of cascading events gives a false view on the operational status of the system, while under-estimating the probability of cascading events can have detrimental consequences to the secure operation of the power system. The described metrics are given by:

$$MOE = max(\hat{y}_i - y_i) \tag{4.19}$$

$$MUE = max(y_i - \hat{y}_i) \tag{4.20}$$

where in equations 4.16-4.20, n is the number of data samples, i is the particular observation, y_i is the actual value, $\hat{y}_i$ is the predicted value and $\bar{y_i}$ the mean of all observations.

Evaluation Metrics for the binary classification models

Accuracy, Precision, Recall and F1 score are typical measures used in ML that capture different aspects of the performance of a binary classifier [86].

Accuracy measures the number of correct predictions made by the model in relation to the number of total predictions, and is given by:

$$Accuracy\ (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN}} \tag{4.21}$$

Precision describes the number of positive class predictions that actually belong to the positive class. It is given by:

$$Precision\ (\%) = \frac{n_{TP}}{n_{TP} + n_{FP}} \tag{4.22}$$

Recall represents the number of positive class predictions made out of all positive samples, and is given by:

$$Recall\ (\%) = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{4.23}$$

F1 score takes both Precision and Recall into consideration, and is defined as the harmonic mean of these two metrics:

$$F1\ Score\ (\%) = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.24}$$

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

These metrics can provide valuable information about the task of classifying whether or not a cascading event will occur: *Accuracy* describes the percentage of correct predictions. *Precision* describes the percentage of the cases predicted to include cascading events that is actually correct and *Recall* the percentage of actual cases with cascading events that is predicted correctly. *F1 Score* is a metric that combines *Precision* and *Recall*, and it is defined as the harmonic mean of these two metrics.

4.2.5 Permutation Feature Importance

A feature importance analysis is performed to investigate the effect of each feature on model performance, with the goal of identifying the most important features and consequently system variables corresponding to them. These features represent the initial operating conditions that describe the measured electrical variables of the system

and can be acquired by PMUs in practical applications. As in large real-life power systems a certain number of PMUs is installed and in certain locations [87], it is of great importance to investigate in which way and to what extent these measurements affect the prediction of cascading events. This can also contribute to better understanding the mechanisms involved behind cascading events by identifying important system variables that might affect the appearance of cascades.

The concept of permutation feature importance is permuting randomly each time a single feature, while keeping all the other features stable, and calculating the model performance. A permutation of an important feature would cause a drop in model performance, as described in [88]. For the feature importance analysis, each input feature is randomly permuted one by one and the performance of the model with the permuted input dataset on the test labels is compared to the performance of the original model. As this method is applied on the pre-trained model, it does not require training the model again, thus being computationally efficient. The permutation feature importance for each feature j in the dataset is calculated by:

$$FI_j = s_{or} - s_{j,perm} \tag{4.25}$$

where s_{or} is the score of the model with the original dataset, and $s_{j,perm}$ is the score of the model when feature j is permuted.

4.2.6 SHAP- SHapley Additive exPlanations

SHAP (SHapley Additive exPlanations) is a post-hoc XAI method for the interpretation of the predictions of ML models [89]. The way that SHAP provides explanations is by assigning an importance value to each feature for an individual prediction. This is achieved be calculating the average output for every possible combination of feature values and comparing it to the output of the model with the inclusion of the specific feature in question. The resulting difference represents the contribution of this feature to the prediction given the presence or absence of other features. The estimated SHAP value for a certain set of features, is the contribution of a feature to the difference between the actual model prediction and the average model prediction. Due to their

additive nature, the SHAP values for all features can be added together to obtain the model prediction for this specific instance.

The SHAP method has a strong theoretical background which is based on the calculation of Shapley values from game theory. Furthermore, the SHAP values for individual predictions can be aggregated to provide global interpretation insights, which are consistent to the local explanations. An overview of the application of SHAP method for explaining the predictions of a ML model is presented in Fig. 4.3.

Shapley values, first introduced in [90], describe the importance of each player in a cooperative game. In the context of ML, the "players" are the input features and the "game" is the model prediction. Using SHAP, the contribution of each feature to the model prediction can be quantified.

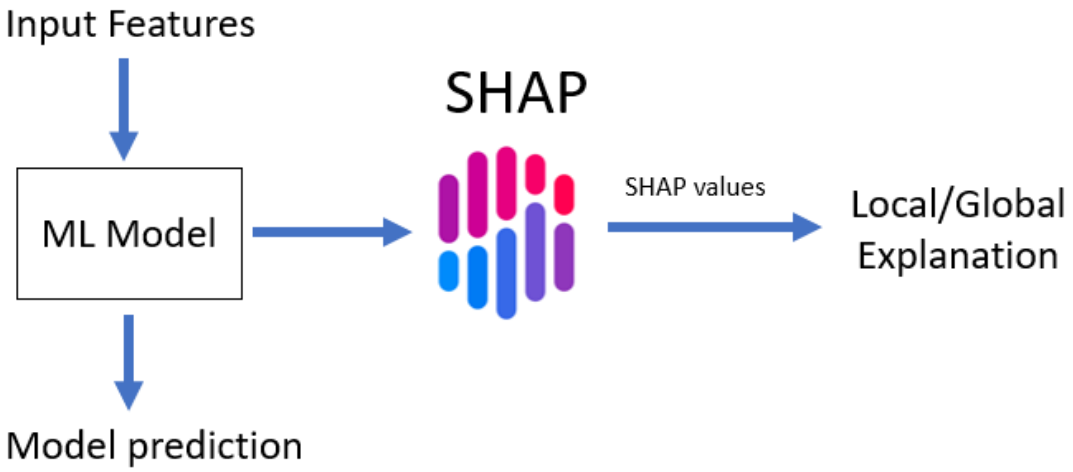

Figure 4.3: Explaining model predictions with SHAP.

The Shapley value ϕ_i, for the black-box model f at the input point x, can be calculated by:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z|!\,(M - |z'| - 1)!}{M!} \left(f_x(z') - f_x(z' \setminus i)\right) \tag{4.26}$$

where $|z'|$ is the number of non-zero entries in z', $z' \subseteq x'$ are all the $|z'|$ vectors where the non-zero entries are a subset of the non-zero entries in x', $z' \setminus i$ denotes setting $z'_i = 0$ and M is the number of simplified inputs.

In a ML context, calculating the exact Shapley would require retraining the model on the missing features, which would be very challenging computationally. To overcome this, SHAP expresses the explanation model as a conditional expectation function of the original model. The SHAP values are calculated by setting $f_x(z') = E[f(z)|z_S]$, where S is the set of non-zero indices in z', and represent how to move from the base value $E[f(z)]$, which would be the output if no features are known, to the current output $f(x)$. In other words, the SHAP value attributed to a feature when conditioning on that feature is the change in the average model prediction.

In order to calculate the exact SHAP values, it is needed to go through every possible coalition of features, which grows exponentially to the number of features. Due to the large computational effort that is needed, methods that combine SHAP with other XAI techniques have been proposed in order to efficiently approximate the SHAP values.

KernelSHAP, proposed in [89], is a model-agnostic approximation that combines LIME and SHAP, and calculates the SHAP values by using a weighted linear regression. The key difference to the LIME method is the way that the linear regression model is weighted. In LIME, the randomly sampled instances are weighted depending on how close they are to the original sample [91]. KernelSHAP assigns the weights to the sample instances according to the weight that this feature coalition would get in Shapley value estimation.

TreeSHAP is a faster method for the calculation of SHAP values of tree-based models, introduced in [92]. This method utilizes the tree structure and computes the SHAP values in polynomial, instead of exponential, time. This is achieved by pushing all the possible subsets of features down the tree at the same time. In addition, because of the additive characteristic of Shapley values, the SHAP values of a tree ensemble method can be calculated by aggregating the SHAP values of individual trees.

DeepSHAP is a method that combines SHAP and DeepLIFT [93], that can be used to explain the predictions of deep learning models [89]. This method calculates the SHAP values by utilizing the gradients of the network output with respect to the input features. This is achieved by recursively passing the multipliers defined in DeepLIFT, but in this case expressed in terms of SHAP values, backwards through the network.

In this study, the TreeSHAP method is used to identify the effect attributes of the initial operating conditions values on the prediction of cascading events. TreeSHAP will be used to analyze specific model predictions (local explanation), but also provide explanations across all the model predictions (global explanation). This information would be particularly useful for better understanding how the model reaches a prediction about the appearance of cascading events, and it would increase the trust of system operators on using ML-based models in practical applications.

4.3 Using Machine Learning methods to predict cascading events

4.3.1 Predicting the probability of cascading events

Predicting the probability of cascading events can be formulated as a regression problem. In this case the output, which represents a probability, is a continuous variable, and the input or independent variables, are the electrical system variables during steady-state that can be provided by PMU measurements. For each unique operating condition, the probability of cascading events is calculated by the percentage of the simulations that cascading events have appeared into, for this initial operating conditions case. This is mathematically described by:

$$P_{casc,i} = \frac{n_{i,casc}}{n_{lines}} \times 100\% \tag{4.27}$$

where i is the index of an operating conditions case, $n_{i,casc}$ is the number of simulations with cascading events for this operating condition, and n_{lines} is the number of lines in the network model.

An illustration that showcases the ML model application for the regression problem is provided in Fig. 4.4.

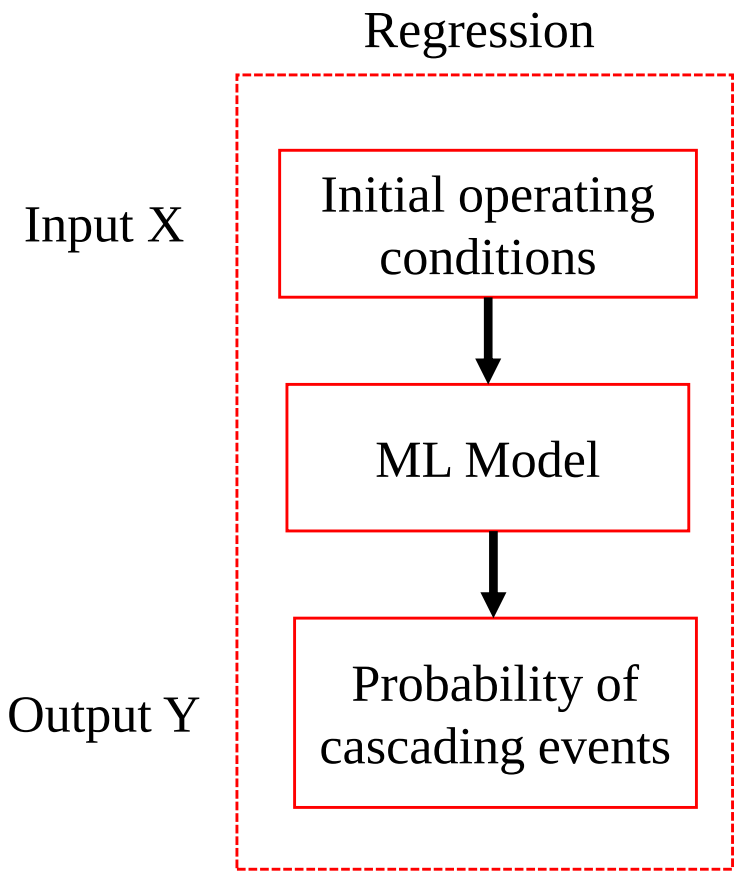

Figure 4.4: Predicting the probability of cascading events as a regression problem.

4.3.2 Predicting the appearance of cascading events in individual cases

Predicting the appearance or not of cascading events in a certain case can be formulated as a binary classification problem. In this case, as 0 are labelled the cases with no cascading events and as 1 the cases with cascading events. These labels are the output of the model, and the input includes the initial operating conditions and additional information about the line on which the initial applied fault happens. The ML model application for the binary classification problem is highlighted in Fig. 4.5.

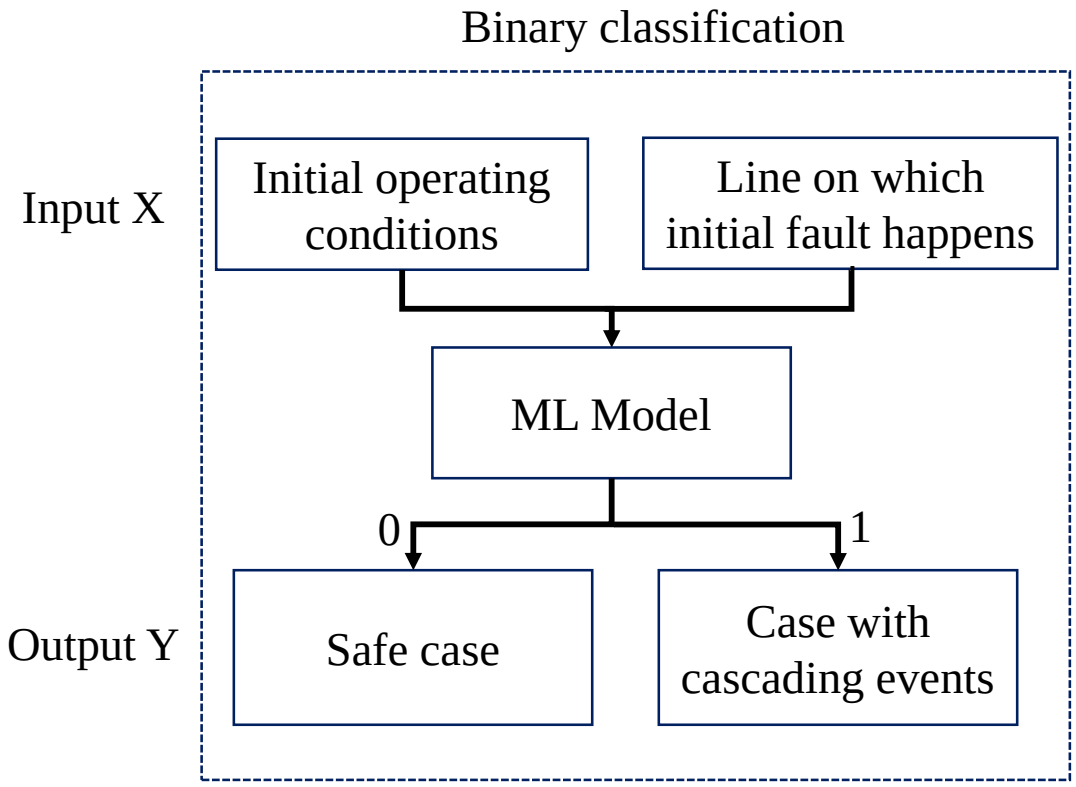

Figure 4.5: Predicting the appearance of cascading events as a binary classification problem.

4.3.3 Machine Learning methods application

In this study the ML models considered are applied using the k-fold validation, as described in Section 4.2.3. The value of k is set to 10, as this value has been found through experimentation to result in a estimate of model performance with low bias and modest variance [94]. For the regression task, the average and standard deviation of the MAE metric across the 10 folds are calculated for validation. Accordingly, for the binary classification task the average and the standard deviation of accuracy are calculated and compared.

The models are initially trained and validated using the default parameters. More specifically, the number of the neighbours for the KNN model is set to 5, the Radial basis function layer is used for the SVM with $C = 1$, the Gini index is used for the splitting of the CART model, and the number of estimators (trees) for the ensemble tree-based

models (Random Forest, Extra trees, AdaBoost, Gradient Boosting, XGBoost) is set to 100. The goal of this step is to quantify the suitability of each algorithm for the problem of predicting cascading events using measurement data from the steady-state conditions. Next, the model that showcases the best performance is further tuned to improve its performance by applying a grid-search optimization, as described in Section 4.4. All the ML models are implemented using the Scikit-learn Python library [95].

4.3.4 Study case and Dataset

The proposed framework is applied on a modified version of the IEEE-39 bus model, which includes the action of protection devices and RES penetration. The modified network model is described in detail in Section 3.4.2. The dataset consists of the results of the dynamic RMS simulations performed for this network as defined in the Base case of Section 3.4.3. Based on these test cases, 44064 simulations in total have been performed, with cascading events appearing in 7131 simulations (16.2% of the simulations). These simulations have been performed for 1296 different initial operating conditions and three-phase faults on each of the 34 lines. When the line on which the fault happens is not considered, and the problem is formed as a regression problem with the output being the probability of cascading events, the dataset consists of the 1296 cases, split into 80% (1036 cases)-20% (260 cases), for the 10-fold cross-validation and test set respectively.

When the fault location is added to the input features of the models, and the problem is formed as a binary classification problem, then the dataset consists of all the 44064 simulations (1296 different operating conditions $\times$ 34 lines). The whole dataset is again split into 80% (35451 simulations) - 20% (8613 simulations), for the 10-fold cross-validation and test set respectively. It should be noted that the split is performed in a stratified manner, meaning that the same percentage of cases with cascading events and safe cases exists in both splits.

For each simulation 178 features that represent the electrical variables of the system during steady-state are obtained over various network locations. These features include the voltage and frequency of every bus element, and the current, active and reactive

power of every line of the network. A pre-processing scaling step is applied on the features, so that all the features values are in the range of [0,1] before being input to the ML models. For the binary classification problem, if cascading events appear at a simulation then it is labelled as 1, and if no cascading events appear then it is labelled as 0.

4.4 Results

4.4.1 Predicting the probability of cascading events

After performing 10-fold cross validation, the Mean and Standard Deviation value of MAE over the 10 folds of the ML models for predicting the probability of cascading events are presented in Table 4.1. It should be noted that a smaller MAE score means better performance. As it can be observed, the KNN algorithm achieves the lowest MAE, in terms of mean and standard deviation, 2.09% and 0.20% respectively. The two algorithms that also showcase a good, but slightly worse, performance are the CART algorithm and Linear Regression (LR), despite the simplicity of this algorithm. The LASSO and Elastic Net (EN) regression algorithms have the poorer performance scores. The performance of the models is also illustrated using boxplots in Fig. 4.6.

Table 4.1: ML regression models 10-fold validation results.

Algorithm	MAE Mean (%)	MAE Standard Deviation (%)
LR	2.82	1.06
LASSO	6.07	0.42
EN	5.96	0.39
KNN	2.09	0.20
CART	2.38	0.36
SVR	3.07	0.28

Since the CART algorithm showcases a good performance, ensemble methods that consist of multiple Decision Trees have been also trained and tested. In a similar manner, the results of the ensemble methods after performing 10-fold cross validation are presented in Table 4.2 and with boxplots in Fig. 4.7. The model that showcases the best performance is the Extra Trees (ET) model, which performs with MAE mean of

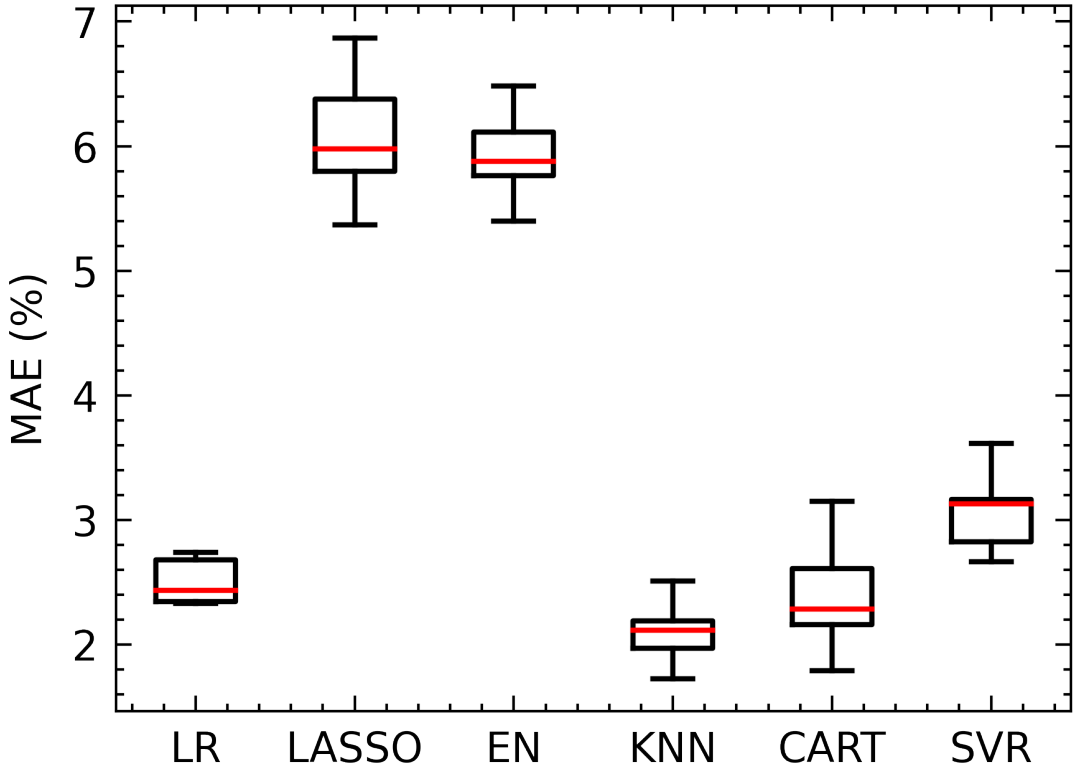

Figure 4.6: Boxplots illustrating the ML regression models 10-fold validation results.

1.66% and MAE standard deviation of 0.25%. The Gradient Boosting (GBM), Random Forest (RF) and XGBoost (XGB) models have overall a similar performance, with the XGB model having the largest MAE standard deviation of 0.29%. The AdaBoost (AB) model has the worst performance of the ensemble methods, with a MAE mean of 2.92% and MAE standard deviation of 0.29%. It can be concluded that the extra randomness of the Extra Trees algorithm fits better the problem of predicting the probability of cascading events.

Table 4.2: Ensemble regression models 10-fold validation results.

Ensemble Method	MAE Mean (%)	MAE Standard Deviation (%)
AB	2.92	0.27
GBM	1.77	0.20
RF	1.77	0.23
ET	1.66	0.25
XGB	1.81	0.29

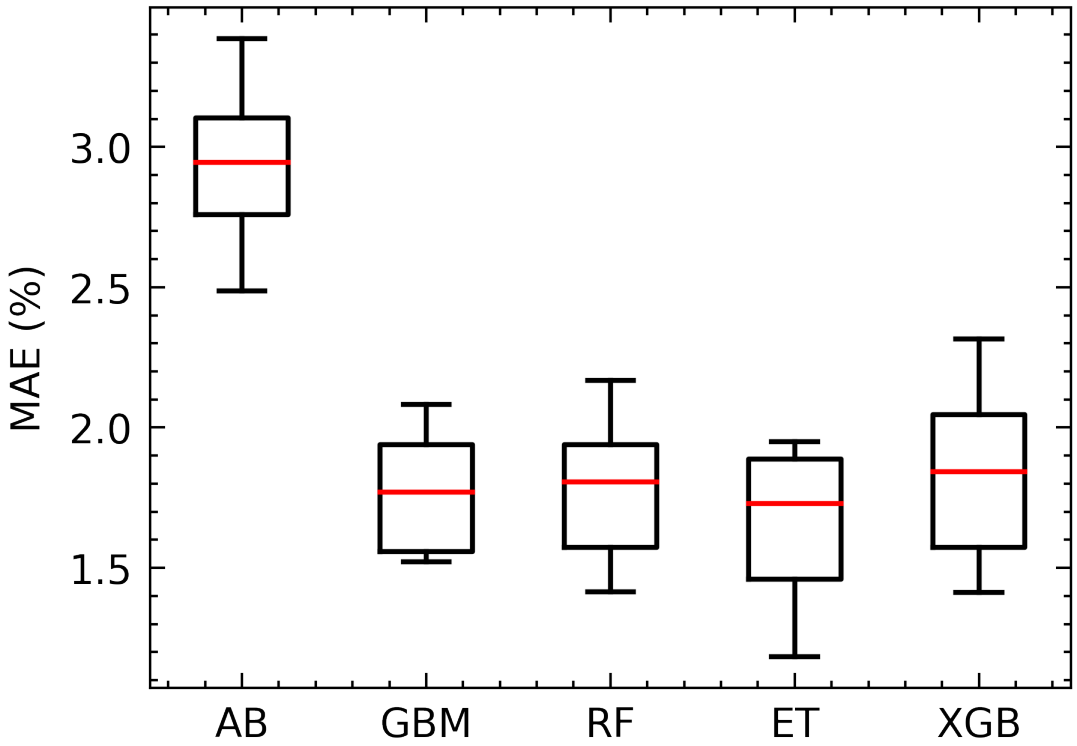

Figure 4.7: Boxplots illustrating the Ensemble regression models 10-fold validation results.

For the algorithm with the best performance on the previous experiments, Extra Trees, a grid search optimization is performed in order to improve the model performance. The grid search is performed for number of estimators in [50, 100, 150, 200, 250, 300, 350, 400], using the same 10-fold cross validation as applied before. After application, the number of trees is set to 200, as this leads to the lowest MAE score.

The best performing model, Extra Trees with 200 trees, is tested on the unseen by the model test dataset. The evaluation results are presented in Table 4.3. The model performs with a MAE score of 1.65% and a RMSE score of 2.57% on the test dataset. It should be noted that the optimization in this case leads to an improvement in MAE score of only 0.01%. Also, the R^2 score of 0.95 indicates a good fit of the regression model to the input data. However, the MOE and MUE metrics have significantly larger values, of 9.57% and 14.10% respectively. The high MUE value of 14.10% is particularly concerning, as under-estimating the probability of cascading events can potentially lead

to severe consequences.

Table 4.3: Extra Trees model performance results on the test dataset.

MAE (%)	RMSE (%)	R^2	MOE (%)	MUE (%)
1.65	2.57	0.95	9.57	14.10

To better understand the model performance, a scatter plot illustrating the predicted model values against the actual values is presented in Fig. 4.8. The diagonal red regression line shows where ideally all the points should be, as on this line the predicted values match the actual values. As it can be observed, for smaller probabilities of cascading events (10%-20%), the model tends to over-estimate the probability of cascading events, as the prediction values are higher than the actual values. On the other hand, when the probability of cascading events is higher (40%-60%), the model under-estimates the probability value. Although fewer instances appear at this range, predicting lower probability than the actual one could lead to a lack of trust on the application of this model in real-life scenarios.

4.5 Predicting the appearance of cascading events in individual cases

Based on the previous results, the regression ML model can under-estimate the probability of cascading events. For this reason, the line on which the initial fault happens is added as a categorical feature to the input features. In this case, the problem is formulated as a binary classification problem, and the output of the ML model is if a cascading event will appear or not, as described in Section 4.3.2.

After performing 10-fold cross validation, the ML models mean accuracy and standard deviation are presented in Table 4.4. For comparison purposes, the results are also visualized using boxplots in Fig. 4.9. The KNN model has the best performance, with mean accuracy of 98.20% and standard deviation of 0.22%. The CART model has a slightly worse performance, with a slightly lower mean accuracy of 97.82% and higher standard deviation of 0.29%. The Naive Bayes (NB) model showcases the worst performance, with a mean accuracy of 80.67% over the 10 folds. According to the

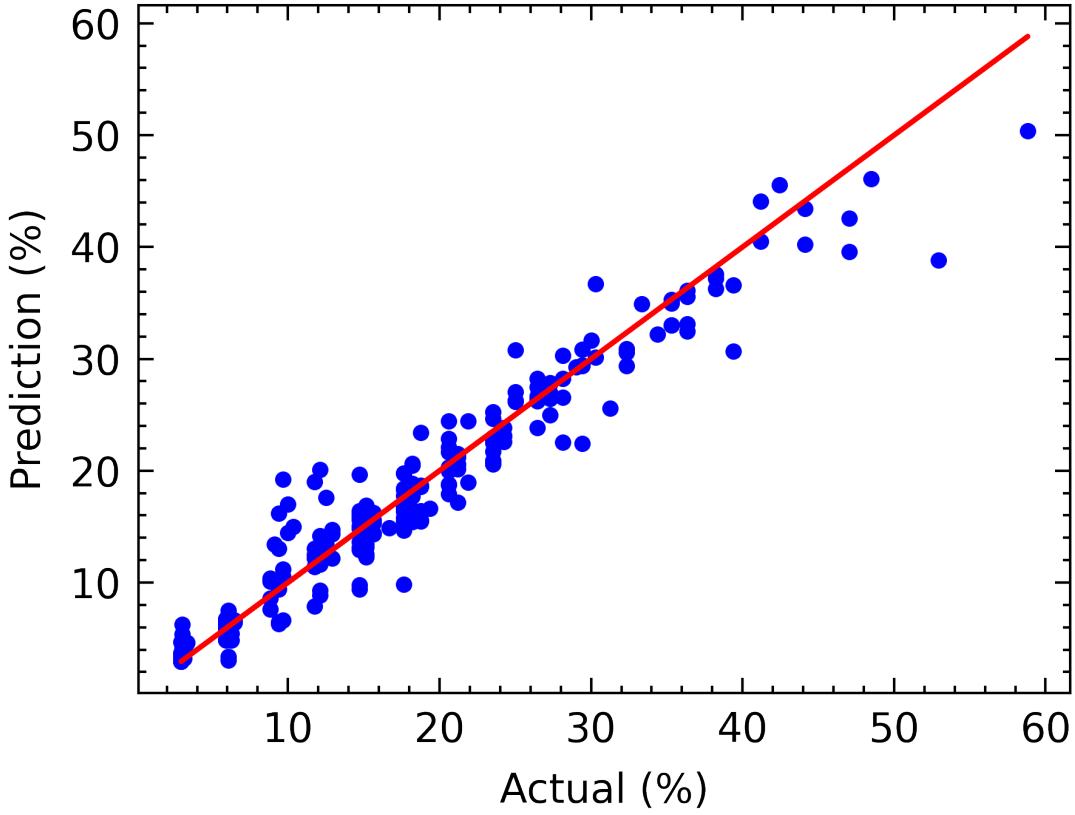

Figure 4.8: Model predictions compared to the actual values.

theoretical background of the Naive Bayes algorithm, it assumes that the features are independent, which assumption in the case of power system features is not valid.

Table 4.4: ML classification models 10-fold validation results.

Algorithm	Accuracy Mean (%)	Accuracy Standard Deviation (%)
LR	95.51	0.30
LDA	91.23	0.40
KNN	98.20	0.22
CART	97.82	0.29
NB	80.67	0.94
SVM	97.11	0.30

As the CART algorithm showcases also in the binary classification problem a good performance, the ensemble methods are trained and validated in the same way on 10-fold cross validation. Similarly, the mean and standard deviation of accuracy are presented in Table 4.5, and using boxplots in Fig. 4.10. In this case, the XGBoost

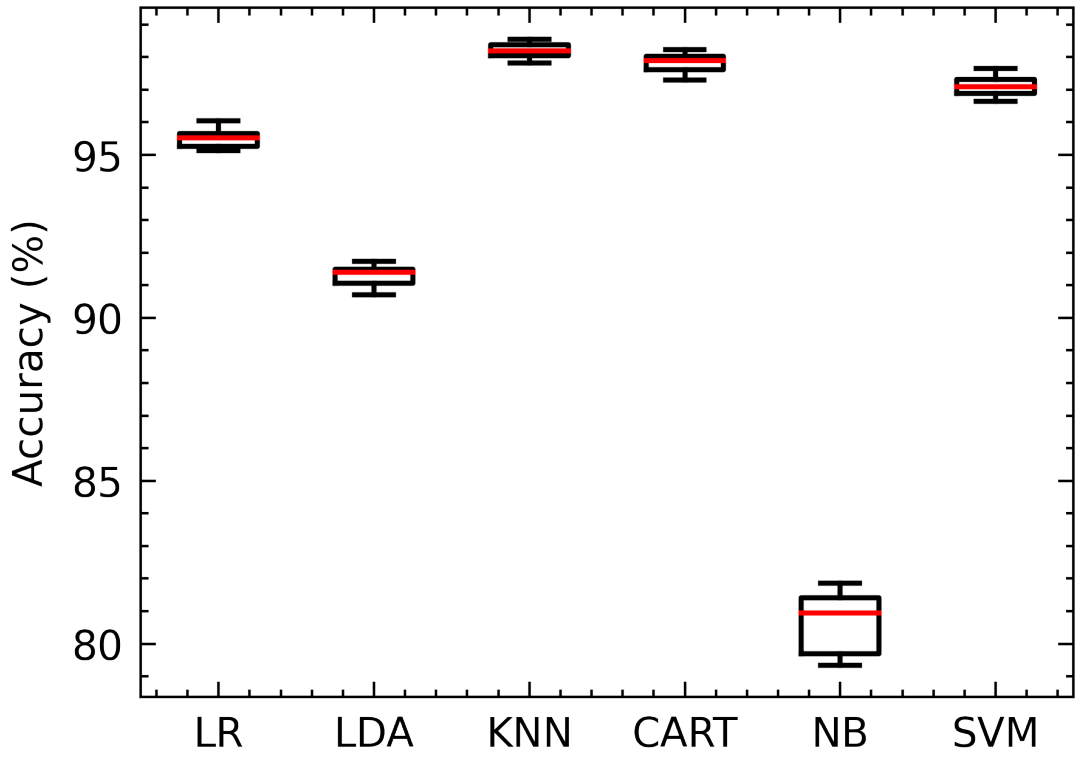

Figure 4.9: Boxplots illustrating the ML classification models 10-fold validation results.

(XGB) model has the highest mean accuracy of 98.68% and standard deviation of 0.20%, while the Extra Trees (ET) model has a slightly worse accuracy of 98.26%, but lower standard deviation of 0.17%. According to the boxplots, the XGBoost model showcases overall the best performance.

Table 4.5: Ensemble classification models 10-fold validation results.

Ensemble Method	Accuracy Mean (%)	Accuracy Standard Deviation (%)
AB	94.61	0.25
GBM	97.27	0.23
RF	97.63	0.20
ET	98.26	0.17
XGB	98.68	0.20

As the XGBoost is the best performing model, a grid search optimization is next performed in order to improve the model performance. The grid search is performed for number of estimators in [50,100,150,200,250,300,350,400], using the same 10-fold

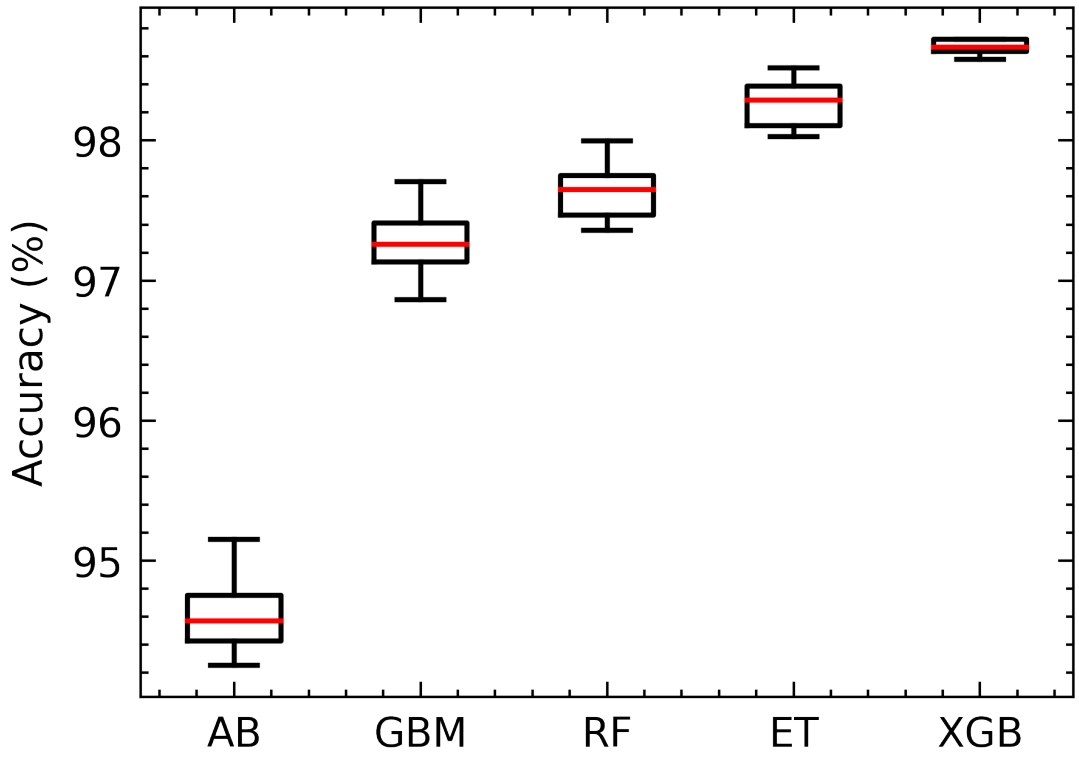

Figure 4.10: Boxplots illustrating the Ensemble classification models 10-fold validation results.

cross validation. The number of estimators (trees) is set to 350, as this is the value that leads to the highest accuracy.

The XGBoost model with 350 estimators is evaluated on the test dataset, and the result metrics are presented in Table 4.6. The model achieves a 98.61% accuracy on the test dataset, showcasing a precision score of 97.46%, and a lower recall score of 94.04%. It should be noted that in this case recall is particularly important, as it represents the proportion of the actual cases with cascading events that are predicted as cases with cascading events.

To better understand the predictions of the classification model, the confusion matrix is presented in Table 4.7. The XGBoost model correctly classifies 1341 instances as cases with cascading events, however it mis-classifies 85 cases of cascading events as safe cases. Furthermore, it classifies 35 safe cases as cases with cascading events (false alarm).

Table 4.6: Trained models and result metrics.

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
XGB	98.61	97.46	94.04	95.72

Table 4.7: Confusion Matrix for the XGB model.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	1341	35
Predicted Negative (0)	85	7152

4.5.1 Most important Features through Permutation Feature Importance

In order to identify which electrical system variables have the most influential effect on the prediction of cascading events, a permutation feature importance analysis is performed. This technique is applied on the best performing model, based on the previous experiments, the XGBoost model for the binary classification task. The 10 features that when permuted lead to the biggest drop in accuracy, therefore are the most influential, are presented in Fig. 4.11. The feature that when is permuted leads to the biggest drop in accuracy, 23.5%, is the fault location. In other words, the line on which the fault happens is the most important feature for the prediction of cascading events. The second most important feature is the voltage magnitude of Bus NSG2. The drop in accuracy that causes the permutation of this feature is 1.48%, which is significantly lower than the effect of fault location. It should be noted that in this particular network and study cases, the disconnection of wind-farm NSG2 due to Overvoltage is the most common cascading event, so it is correctly identified by the model as an important feature. The rest of the features when permuted lead to a drop in accuracy of less than 0.5%.

4.5.2 Identifying the impact that initial operating conditions have on the prediction of cascading events with SHAP

The previous permutation feature importance analysis helps to identify which features the model identifies as most important, however it does not provide an indication about

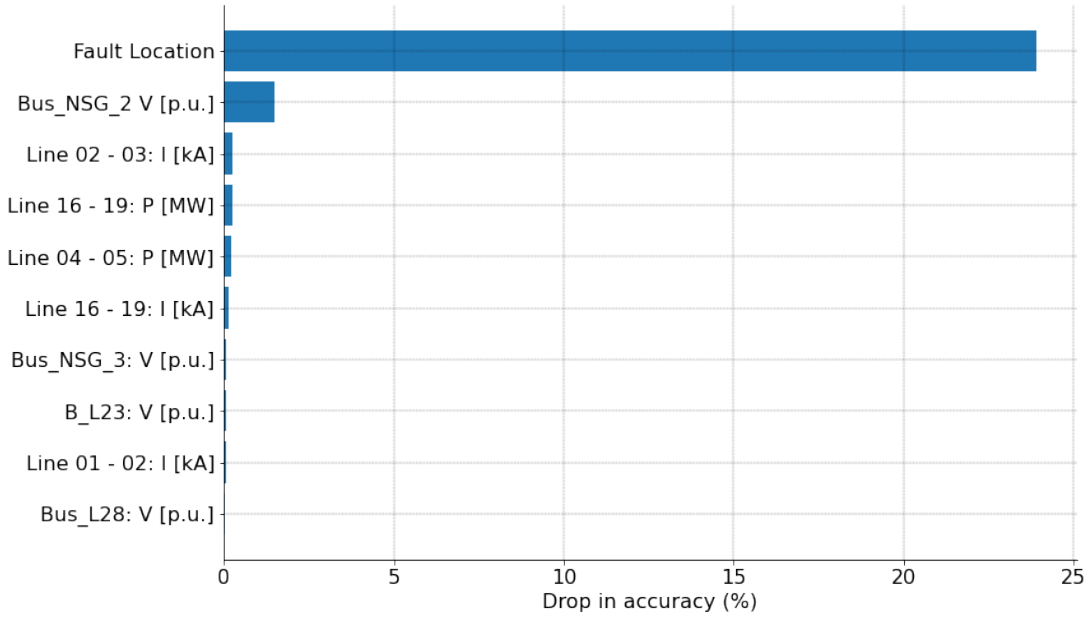

Figure 4.11: The 10 most important features according to Permutation Feature Importance.

how these features affect the model prediction. To address this, the SHAP values are calculated, as described in Section 4.2.6. As the best performing classification model is XGBoost, which is an ensemble method based on decision trees, the SHAP TreeExplainer method is applied to calculate the SHAP values on the test data and identify how the system steady-state measurements affect the prediction of cascading events. After calculating the SHAP values, plots are provided and analyzed for both local explanations, meaning individual model predictions, and global explanation, that describe the model predictions across the whole test dataset.

Explaining a True Positive prediction with SHAP

The plot in Fig. 4.12 illustrates the SHAP values as calculated for a single True Positive prediction. Each SHAP value can be interpreted as a force, that increases (red colour) or decreases (blue colour) the probability of the appearance of cascading events, which is in the range of [0,1]. The forces balance at the actual prediction value of the model, starting from the baseline value $E[x]$, which is the average of all model predictions. The sum of all the SHAP values equals to the difference between the baseline value and

the actual prediction value, as defined by the additive properties of SHAP values. The features are sorted by the absolute SHAP value, which shows how much each feature affects the model prediction. Here for presentation purposes the 20 most influential features, including their actual value at this instance, are shown.

In this case the model correctly classifies this instance as a case with cascading events. More specifically, in this instance a single cascading event appears, the disconnection of wind-farm NSG2 due to Over-voltage. The feature that has the highest impact is the initial fault location, at Line 08-09, which pushes the base prediction value $E[x] = 17\%$ by +66%. The increased voltage magnitude (1.038p.u.) on Bus NSG2 increases the probability by 5%. In this case the Over-voltage relay on this bus triggers the cascading event. The current magnitudes on Line 02-03 and Line 16-17 increase the predicted probability by 3% and 2% respectively. The rest of the features have a smaller impact (less than 1% each) on the prediction of this particular instance.

Explaining a True Negative prediction with SHAP

The SHAP values as calculated for a True Negative prediction are presented in Fig. 4.13. In this case the model accurately classifies an actual case with no cascading events as a safe case. The model prediction at this instance is $f(x) = 0$, so it predicts a 0% probability of the appearance of cascading events. The initial applied fault location on Line 07-08 lowers the base prediction value by 8%. In comparison to the previous instance, the voltage magnitude on Bus NSG2 in this case is lower at 1.009p.u., and decreases probability by 2%. Furthermore, the current magnitudes on Line 02-03 and Line 16-17 decrease the predicted probability by 1% each. The sum of all the other feature attributes decrease the base value by 5%.

Explaining a False Positive prediction with SHAP

Fig. 4.14 illustrates the SHAP values as calculated for a False Positive prediction. In this instance the model falsely classifies a safe case as a case with cascading events. The prediction of the model is 50.7%, only slightly higher than the threshold value of 50% for the binary classification of an instance as 0 or 1. In this case the base prediction

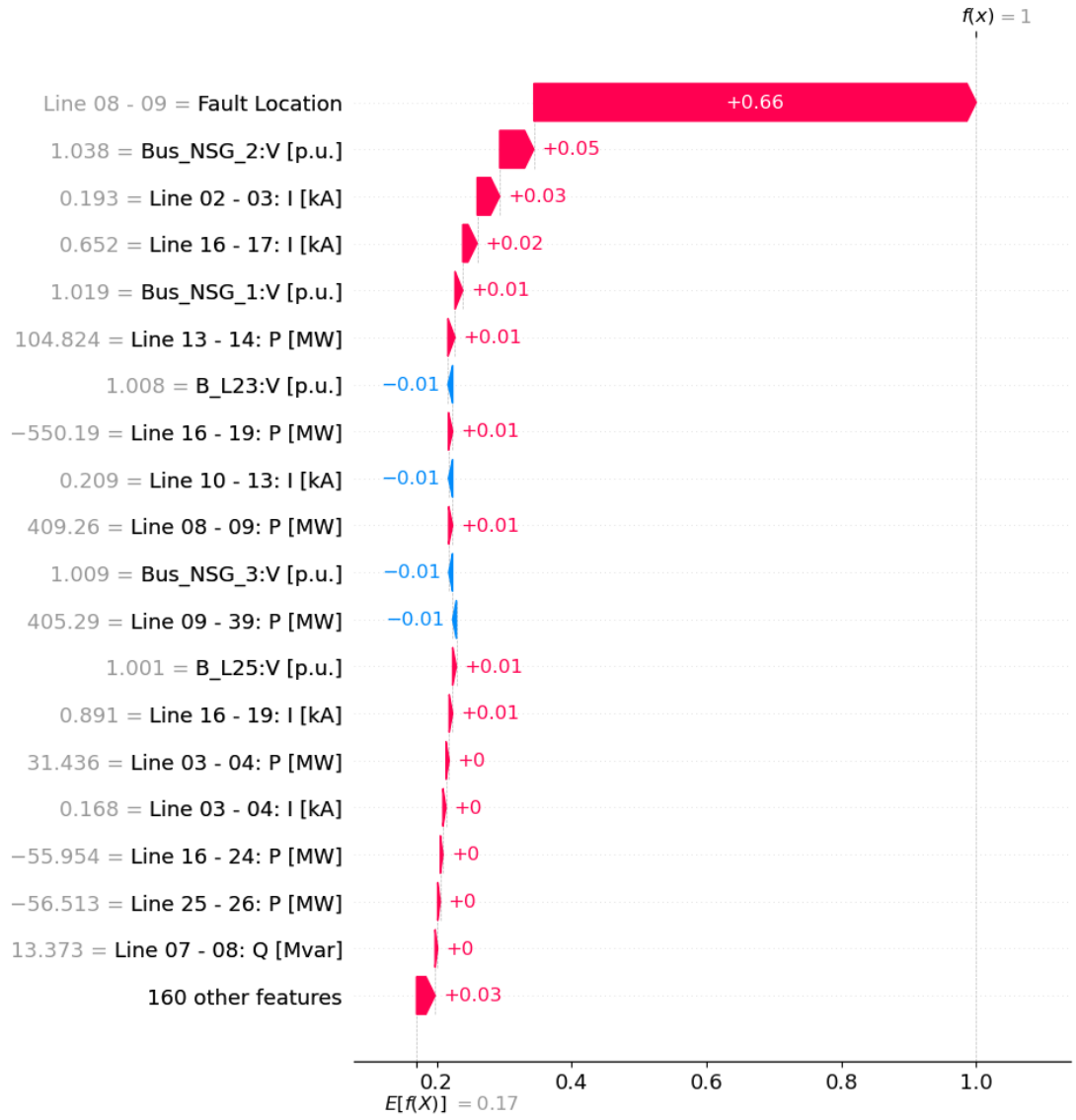

Figure 4.12: SHAP values for a True Positive prediction.

value is pushed higher by the fault location on Line 04-05, meaning that the model considers a higher probability of cascading events because the initial fault appears on this specific line. As the voltage magnitude on Bus NSG2 is 1.05p.u. and on Bus NSG1 is 1.033p.u., which are towards the upper limit, the base value is pushed higher by 4% and 3% respectively. This information can be useful to stakeholders that intend to use such models in real-life scenarios, as it enhances the accountability of the model and provides indications on what causes the model to make a wrong prediction.

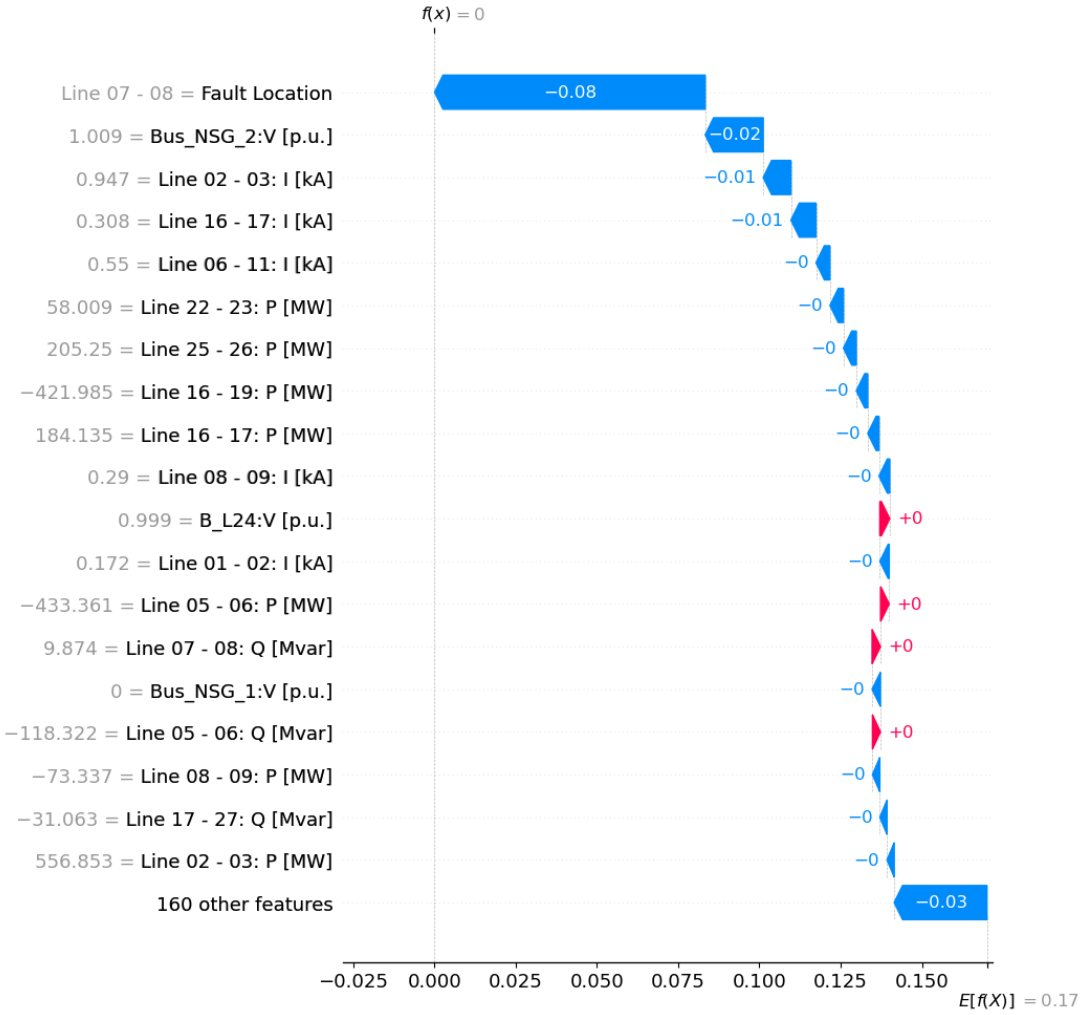

Figure 4.13: SHAP values for a True Negative prediction.

Explaining a False Negative prediction with SHAP

In a similar manner, the SHAP values for a False Negative prediction are presented in Fig. 4.15. In this case the model classifies incorrectly a case with cascading events as a safe case. The single cascading event that appears in this case is the tripping of wind-farm NSG2 due to Over-voltage. According to the calculated SHAP values for this prediction, the voltage magnitude of 1.034p.u. on Bus NSG2 increases the base value probability by 7%. However, the initial fault location on Line 26-29 decreases the predicted probability also by 7%. Most of the rest of the feature values increase the model prediction probability for the appearance of cascading events. The final

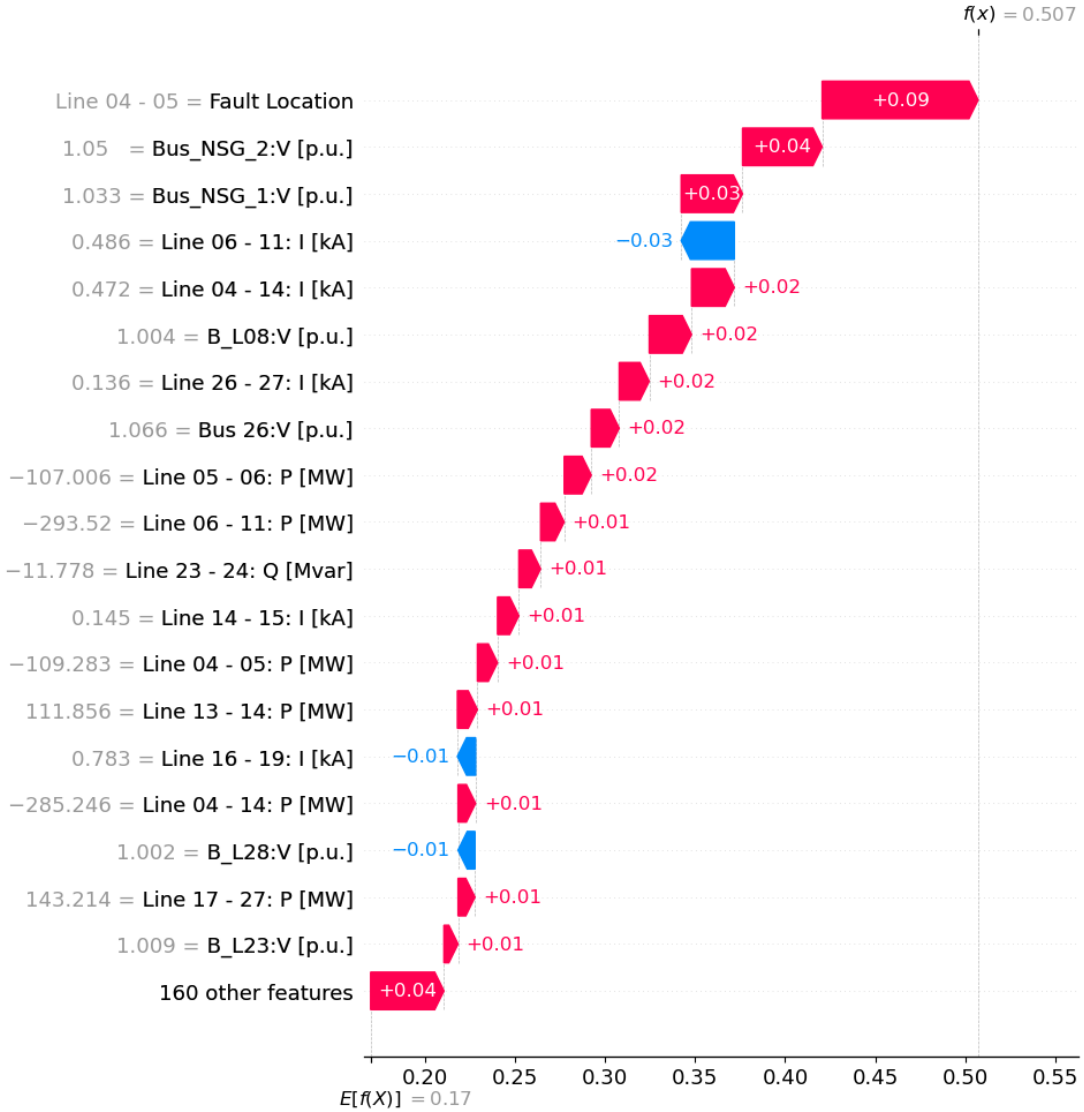

Figure 4.14: SHAP values for a False Positive prediction.

prediction is 48.8%, only slightly lower than the threshold value of 50% for classifying an instance as 0 or 1. This specific case highlights that although the initial fault location feature has a high impact on the model prediction, it can mislead the predicted probability.

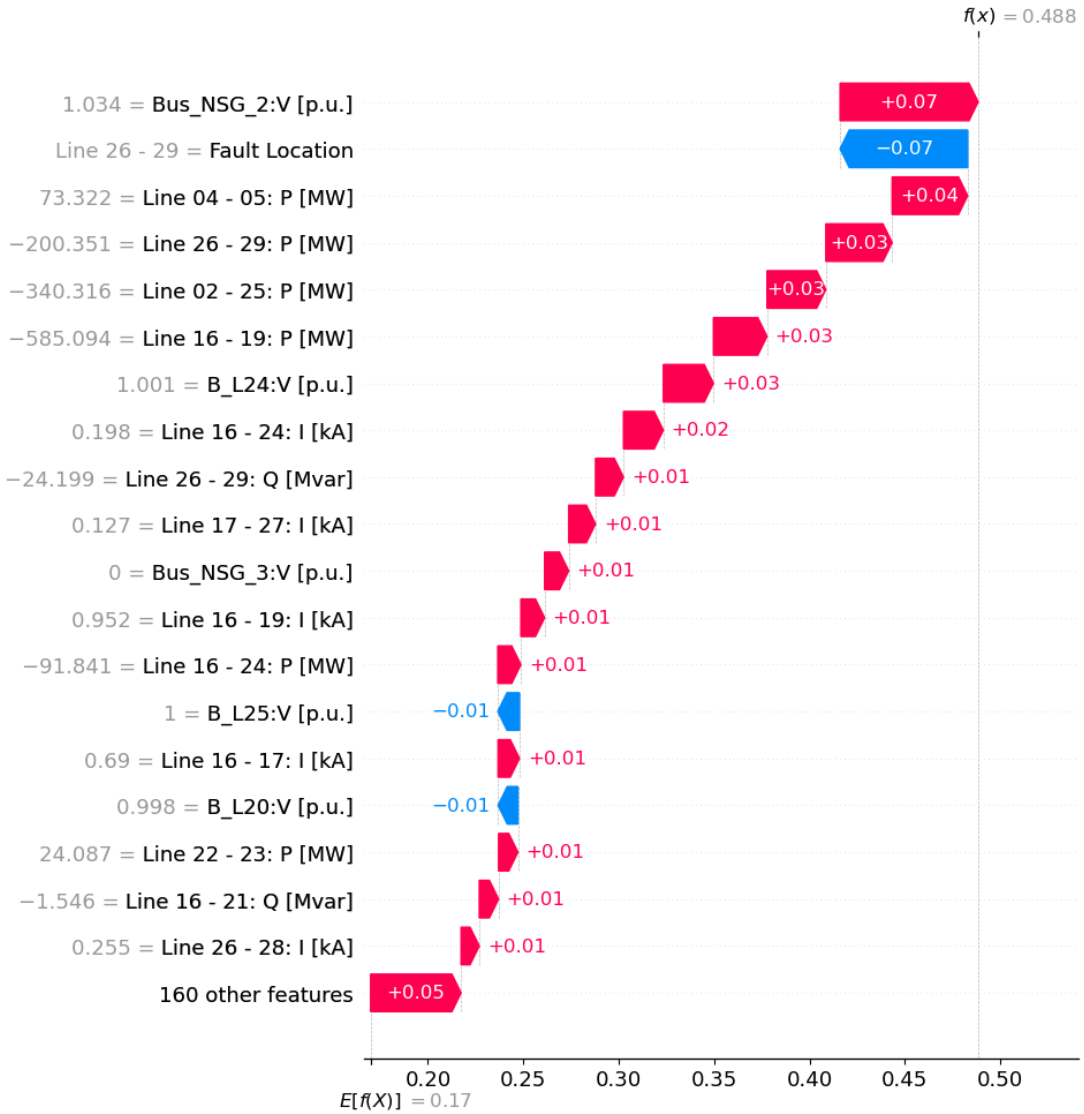

Figure 4.15: SHAP values for a False Negative prediction.

Global Explanation with SHAP

A SHAP summary plot can provide a global explanation about all the instances in the test dataset. This helps to acquire an understanding on how the features affect the model predictions globally, instead of individual instances. Each point on the plot represents an instance and the respective SHAP value. The features are sorted on the y-axis according to their importance. The colour represents the value of the electrical system variable feature, ranging from low values (blue colour) to high values

(red colour). The over-lapping points are positioned across the y-axis direction, giving an indication of the distribution of the SHAP values per feature. This SHAP summary plot can provide indications of the relationship between the value of the features and the impact on the prediction of the appearance of cascading events.

The summary plot presented in Fig. 4.16 showcases the SHAP values for the 20 most important features across all the instances in the test dataset. The most important feature is the fault location, followed by the voltage magnitude on Bus NSG2. As previously mentioned, the tripping of wind farm NSG2 is the most common cascading event for this test network and set of simulations. The rest of the features are related to active power measurements on lines (6 features), current magnitudes on lines (6 features), voltage magnitudes on lines (3 features) and reactive power measurements on lines (2 features). The colour range gives an indication on how the SHAP values are attributed according to the feature values. However, it should be noted that a negative measurement value (and a correspondingly low value after the scaling step) indicates the opposite power flow direction.

By comparing the SHAP summary plot results to the permutation feature importance analysis presented in 4.5.2, it can be highlighted that both methods identify the same two features, that is fault locations and Bus NSG2 voltage magnitude, as the most important ones. Furthermore, the following five most important features as identified by permutation feature importance are also included in the SHAP summary plot, but in a different order. It can be concluded that both methods identify similarly the features that have the highest impact on the model prediction, however SHAP analysis can provide more information on the model prediction process due to the calculation of feature attributions.

4.6 Chapter Conclusions

This Chapter presents a framework for the prediction of cascading events and explaining how the initial operating conditions during steady-state affect the ML model predictions. The problem is first formulated as a regression problem, where the output is the

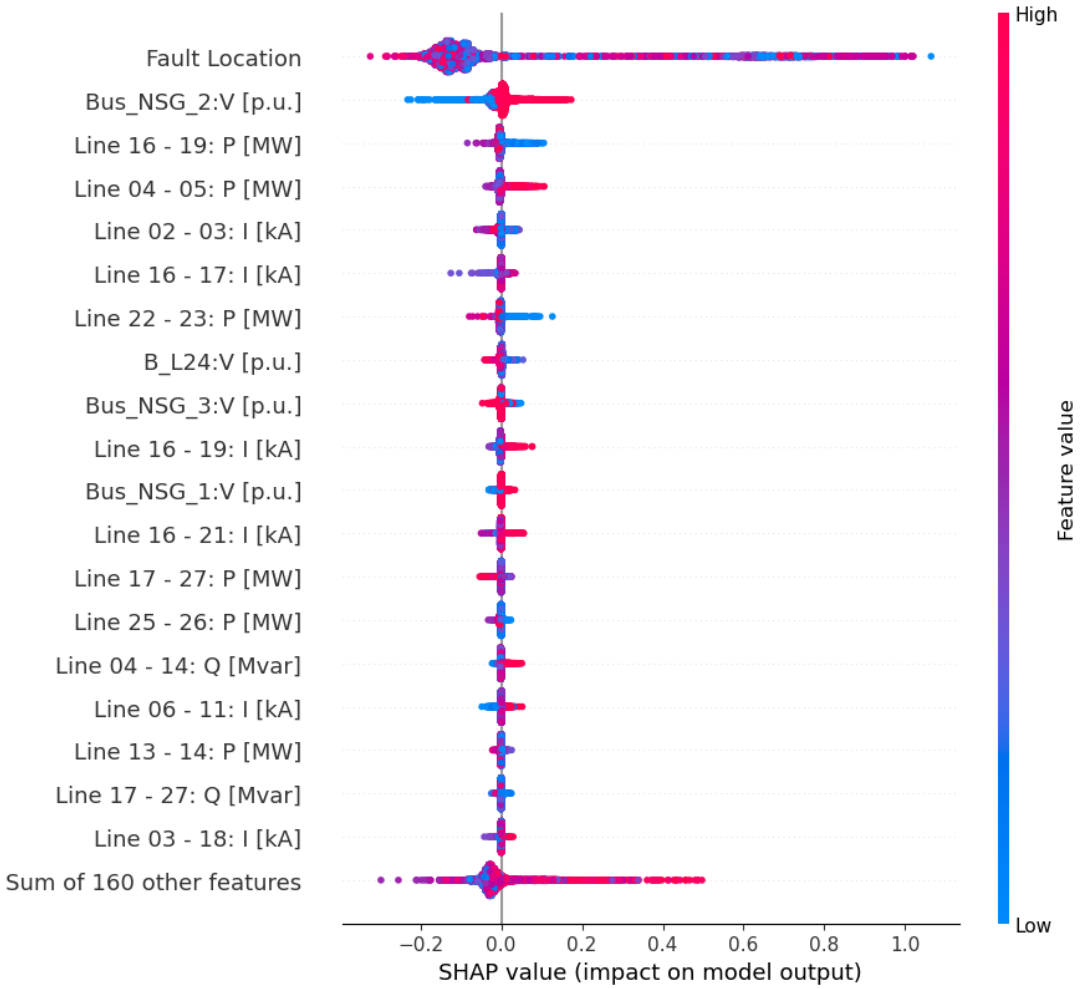

Figure 4.16: Global explanation with SHAP summary plot.

probability of appearance of cascading events, without knowing the line on which the initial fault happens. A wide range of linear and non-linear ML algorithms is trained and validated through 10-fold validation. The framework is applied on a modified version of the IEEE-39 bus system, augmented with wind farms that represent the network RES generation and with protection devices. The best performing regression model is an Extra Trees model that performs with a MAE score of 1.65% on the test dataset. However, the MUE score of 14.1% might not be sufficient, as there is a significant risk of under-evaluating the probability of cascading events. Next, the line on which the initial fault happens as a discreet event is added to the input features and various ML classification algorithms are trained and evaluated in a similar way. The model with

the best performance is XGBoost, which performs with 98.61% accuracy on the test dataset.

A permutation feature importance analysis is performed on the XGBoost model, to identify the most important features for the prediction of the appearance of cascading events. The most important feature as identified by this analysis is the fault location, followed by the feature whose values cause the most common cascading event, the voltage magnitude of Bus NSG2. To gain more insights about how each feature affects the individual model predictions, the SHAP technique is implemented and applied. Instances of both correct and incorrect model predictions are explained and analyzed using SHAP plots. Furthermore, a global explanation SHAP plot reveals the most important features across the whole test dataset. Providing explainable ML models can enhance the trust of system operators on data-driven methods and lead to more widespread use of ML based models in power systems practical applications.

Chapter 5

Predicting the Onset of Cascading Events using Long-short Term Memory Networks and Time-series Measurement Data

5.1 Introduction

In modern power systems, the uncertainty that comes with the integration of RES penetration, makes the online dynamic security problem a challenging task. The highly complex, non-linear behaviour of electrical power systems is not yet well understood, creating the need to re-establish stability definitions [1]. In some occasions, the complex response of a system to a contingency can cause the appearance of cascading events, compromising its secure operation. For this reason, intelligent approaches that are able to predict unstable behaviour by using real-time measurement data, coming from PMUs that are nowadays available, are being investigated to ensure the secure operation of modern power systems with increasing renewable penetration.

The accurate representation of protection devices is a key element in capturing the cascading events that might appear in a system following a contingency [2], [4]. In some cases, protection devices might activate before actual instability limits are reached. Their action can also cause subsequent events, leading to the appearance of cascading event sequences. A method to predict this behaviour can provide valuable information about the online system state, enabling system operators to take corrective actions in order to prevent cascading events from spreading.

As the literature review presented in Chapter 2 has shown, most existing methods mainly aim at identifying specific stability or security related phenomena and do not investigate the possibility of cascading events. So far, methods predicting cascading events have been focusing on static simulations. However, using dynamic models to capture, and consequently predict cascading events can provide a more realistic real-life representation of power systems operation. Combined with the need for dynamic simulations, [2], [4] highlight the importance of representing protection devices in capturing the evolution of cascading events, an approach the proposed method of this Chapter is following. More importantly, [54], [96] highlight the fact that not including protection devices in dynamic studies might result in inaccurate assessment of system behaviour.

So far, existing methods have been focusing on online dynamic security, or individually on transient, small-signal or voltage security of power systems as defined by stability limits. Compared to the work presented in Chapter 4, this methodology focuses more on the online application of the method, and on practical considerations related to PMU measurements. Based on this, the model is trained and evaluated using time-series measurement data, starting from the time of the initial fault clearing, as opposed to data during steady-state (that was the input in Chapter 4). Moreover, the line on which the initial fault happens is not specifically given as an input feature to the model, but the model learns to identify it by the time-series data. The prediction of cascading events has only been examined using data from static simulations. Based on the literature review, an approach for the online identification of cascading events defined by the activation of protection devices using time domain measurements has not yet been proposed.

5.1.1 Contributions

The main contribution of this chapter is the use of supervised machine learning and measurement data from dynamic simulations to predict the appearance of cascading events, as defined by the actions of protection devices. The accurate modelling of system dynamics (capturing phenomena related to voltage, frequency and transient instability) as well as the discrete action of protection devices are considered, which play a significant role in the realistic representation of the appearance of cascading events. This approach for online identification of cascading events including the action of protection devices makes the proposed method distinct to stability/security assessment methods mentioned before, with the key reason being that the actual limit where an event might propagate is more accurately represented. The method is applied on a power system incorporating renewable generation, as the uncertainty that comes with it has shown to affect the power system operation after a contingency. As the prediction takes place in close to real-time, this information could be vital in taking corrective control actions in time and preventing cascading events from spreading. Other contributions include:

- The investigation of the impact that the time window length, used for the online prediction, has on the model performance.
- How the performance of the prediction model differentiates for individual operating conditions. As it is concluded from the results, the model performance can vary for different system loading and wind penetration, which can offer useful information to system operators, related to the level of confidence when using the method.
- A feature importance analysis is also performed to identify which of the features play a significant role in the prediction of the onset of blackouts. This can offer interesting information on the parameters affecting cascading events as well as identify the specific PMU measurements that have the highest impact on the model performance, which can inform measurement infrastructure decisions.
- The model performance is evaluated considering limited availability of PMU measurements and noisy data, which can be found in practical applications.

5.2 Methodology

The proposed framework aims to the online identification of cascading events followed by an initial disturbance. A schematic illustrating the main steps of the framework online and offline stages, which are described in detail below, is presented in Fig. 5.1.

5.2.1 Detailed Procedure

The method presented in this chapter consists of two main stages: i) the offline generation of the dataset and the training of the appropriate supervised machine learning model, and ii) the online binary classification using the pre-trained model to predict the appearance of cascading events. A cascading event in this study is defined as the trip of a component caused by the activation of protection device after (and in addition to) the initial disturbance. During the offline stage, a number of dynamic RMS (Root Mean Square) simulations for various initial operating conditions and contingencies is performed, taking into consideration the increased uncertainty that comes with RES penetration and the reduced network inertia caused by SG disconnection as dictated by economic dispatch. The initial applied fault may cause the appearance of cascading events, as dictated by the discrete action of protection devices that have been implemented in the system. The time-series data obtained from the dynamic simulations are pre-processed to represent a typical PMU sampling rate and are subsequently used for training the model. For this method, Long-short term memory networks (LSTM) have been used, because of their ability to store information and to learn from time series dependencies. This approach is compared to the performance of a regular feed-forward neural network, as a baseline model, and to the performance of a simple recurrent neural network (RNN).

In practical applications, the time domain measurement data during the online phase can be obtained from PMU measurements [97] and used as input to the pre-trained machine learning model to predict the appearance of cascading events. Another potential application of the proposed method is as a screening method for planning studies. In this case, a simulation can be performed for only a short duration, and

then the ML model can be utilized to evaluate if a cascading event will appear or not. This way, the fast inference time of the ML model is taken into advantage, instead of running computationally demanding detailed simulations for long timescales.

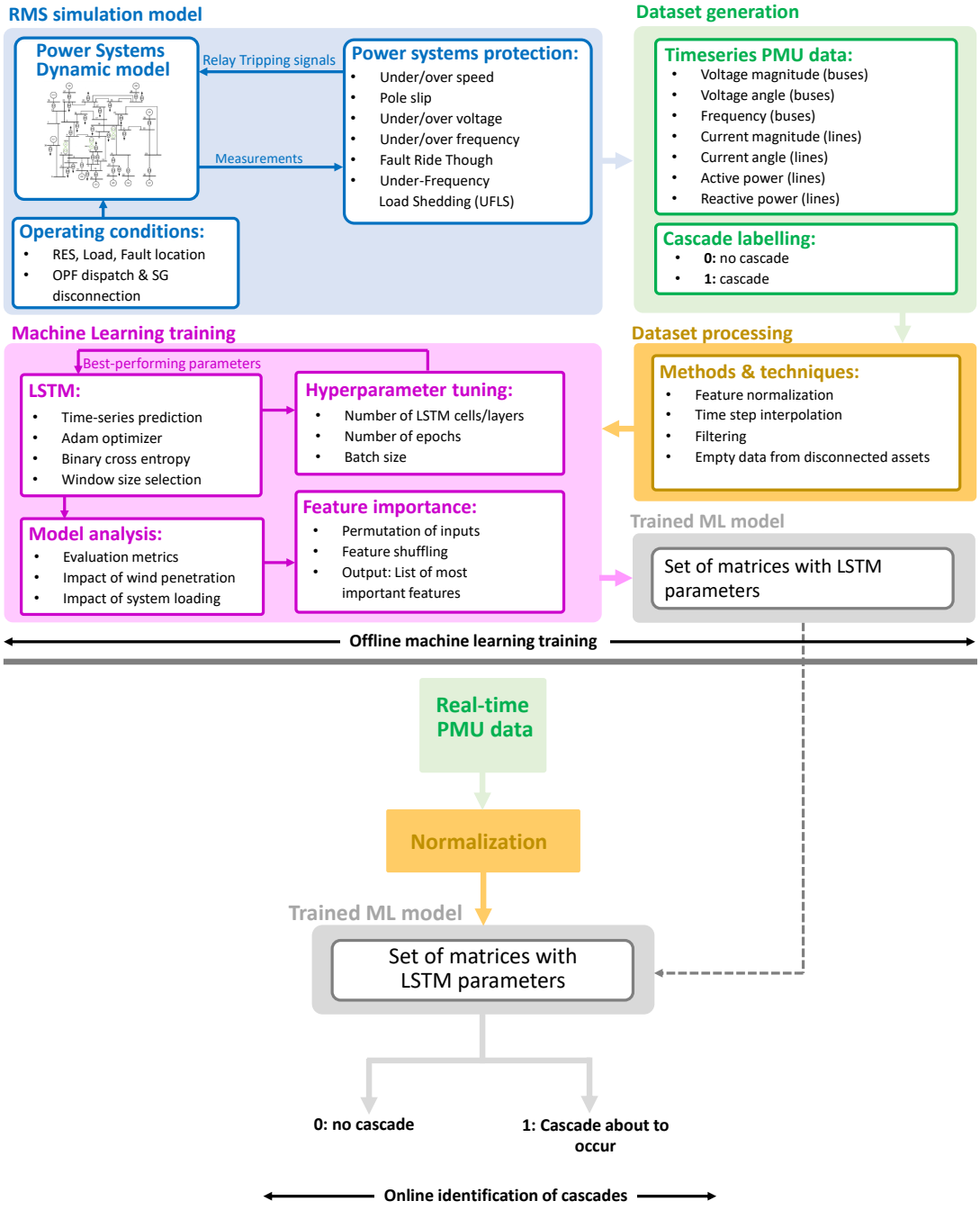

Figure 5.1: Flowchart illustrating the steps of the proposed framework.

It should be noted that in this study, measurement data from simulations have been used for training and testing the method. The analysis of the model performance is related to the window size and the performance for different loading and RES penetration levels. A feature importance analysis using the pre-trained model is then carried out to identify the most important features. Taking into account practical applications, the model performance is evaluated considering limited availability and noise of measurement data.

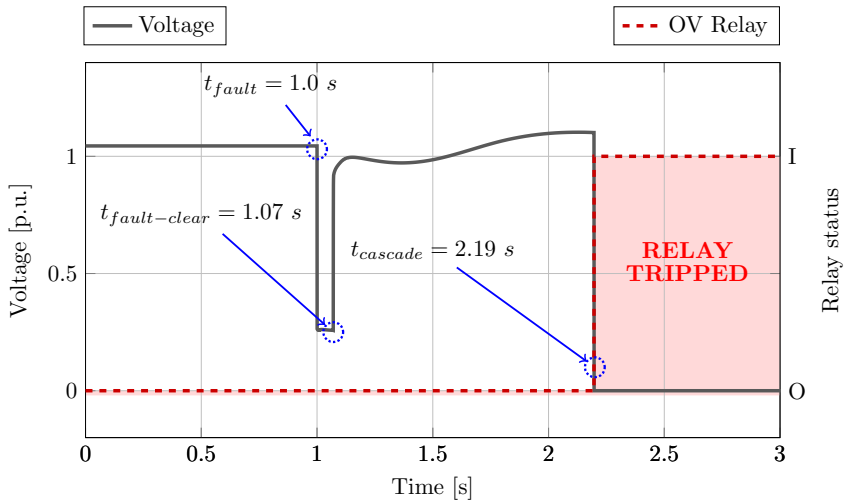

Figure 5.2: Example of a cascading event and the method application.

5.2.2 Example of a Cascading Event and Method Application

An example of a cascading event and the application of the proposed method is shown in Fig. 5.2. In this plot, the cause of a cascading event, the tripping of a wind generator (NSG2) due to over-voltage simulated in the test network used in this study, and the signal of the protection relay are presented. After the fault clearance (1.07s), as the bus voltage recovers it causes the violation of the over-voltage protection limits (in this case 1.1 p.u. for over 0.15s) that leads to the activation of the over-voltage relay and the disconnection of the wind generator from the grid (2.19s). The tripping of this wind generator may cause the violation of other protection device limits, causing the tripping of more components and creating a sequence of cascading events. Predicting the possibility of voltage instability in this case might not capture the tripping of this

element and any subsequent events. This highlights the importance of capturing, and subsequently being able to predict, the action of protection devices, and not just the instability mechanisms involved.

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

The action of Load Tap Changers (LTCs) and Over-excitation Limiters (OELs) has been also implemented within the model, in order to capture longer phenomena related to voltage instability [49]. The duration of the RMS simulations has been set to 120s to capture both fast and slower evolving dynamic phenomena.

In large power systems there is a wide range of uncertainties including load variation and RES, that can affect the dynamic behaviour of the system. In this study, the sampling of possible initial operating conditions is based on the discretization of the operating range of the variables of interest, towards creating a large data-set of operating scenarios and events. The parameters considered for this purpose are RES output generation (for each RES unit), system loading and line fault location.

5.2.3 Dataset Generation

The dataset generation procedure is described in detail in Section 3.4.1. The time series data of each simulation is obtained, with a total of 178 features describing the states in

various power system locations over time. These features represent the measurements that can be obtained from PMU devices found in real power systems and include the voltage and frequency of every bus element, and the current, active and reactive power of every line of the network. At the end of each simulation it is determined whether the system remains secure or if any cascading events occur, and each simulation is labelled as 0 or 1 respectively.

5.2.4 Preprocessing Data

In order to convert the dataset into the input format expected by the selected machine learning model, a number of pre-processing steps are performed, consisting of feature normalisation, time step interpolation and windowing. All features are normalized to enable more efficient and high performance model training. Normalization is a widely used pre-processing technique for data smoothing that aims to retain information related to within feature variance, while ensuring that all features are on the same scale. In this study, the scaling value for all quantities in per unit (p.u.) has been set to 1, and for all the other quantities it has been set to 100. After application all the measurements values are in the range of [-10,10].

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

5.3 Neural Network Models and Training

5.3.1 Artificial Neural Networks (ANN)

An Artificial Neural Network (ANN) is a type of machine learning algorithm that is inspired from biological neural networks, like human brains. A typical ANN is composed of multiple, directly connected layers, that consist of interconnected neurons which can

process and transmit information. These layers include an input layer, one or more hidden layers, and an output layer. The input layer receives the input data, and each subsequent layer processes and transforms the information, until the output layer, that gives the model prediction. An ANN that consists of an input, a hidden, and an output layer is illustrated in Fig. 5.3.

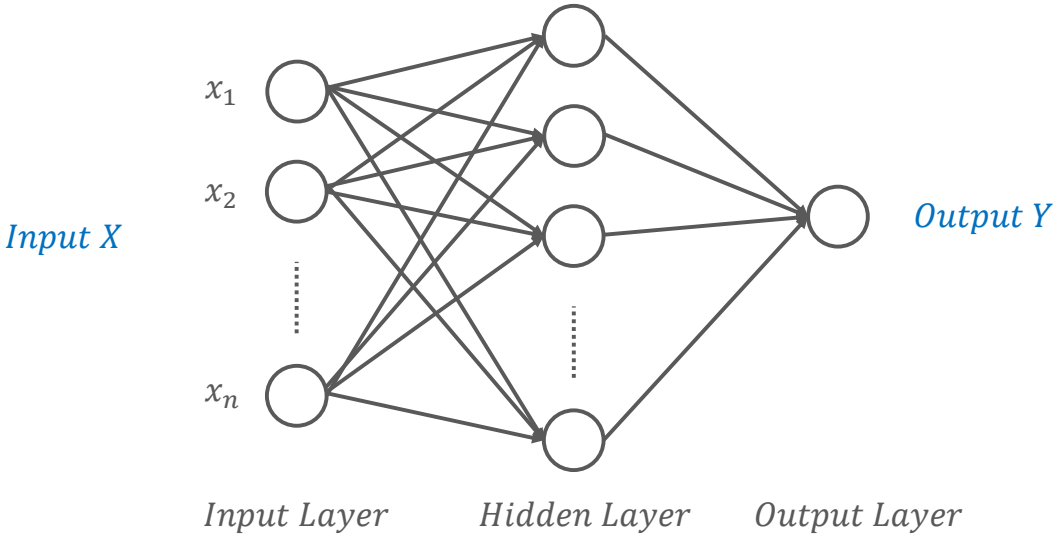

Figure 5.3: Schematic diagram of an ANN

Each neuron of the ANN, receives input from the neurons of the previous layer, and performs a mathematical operation. Subsequently, the output of this neuron is transferred to the neurons of the next layer. The connections between the neurons have certain weights, that are optimized during the ANN training process with the objective of increasing the network performance. As the ANN gets deeper, which means that the network has more hidden layers, it consists of more training parameters and can represent more complex relationships between the input data ad the output value. The output of a neuron is given by:

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{5.1}$$

where x_i are the inputs of the previous layer to the neuron, w_i are the weights associated

with each input, n is the number of inputs to this neuron, and β denotes the bias value that helps to adjust the output of the neuron. The f denotes the activation function, which transforms the input to the desired output of the neuron. Usually, in hidden layers the purpose of the activation function is to introduce non-linearity to the relationship of the neurons.

A feed-forward neural network, also known as a Multi-layer Perceptron (MLP), is a type of ANN in which the information flows in only one direction, from the input to the output layer [98]. The training phase of a feed-forward ANN consists of two stages: forward propagation and back propagation. During the forward propagation, the input data are passed through the interconnected layers of the network and the output value is generated. This process consists of passing the input data through the neurons of the network, calculating the sum of the inputs multiplied by the weights, and passing the value to the activation function. At the output layer, the prediction generated by the network is compared to the true label, and the error is calculated, according to the specified loss function. This is a deterministic process, meaning that for a given set of inputs, the network will produce the same output.

Back propagation is the process during which the error that is calculated at the end of the forward progation phase is propagated back through the network to update the parameters of the model. Starting from the output layer, at each layer the partial derivatives of the total error with respect to each weight are computed, which represent their contribution to the total error. The gradient is then used to update the weights of the connection. A typical optimization algorithm that is used at this step is the gradient descent. The model with the updated weights is then used for the next iteration of forward and back propagation. This training process is iteratively repeated until the loss function converges to a specified value, or the number of iterations reaches a pre-defined value.

5.3.2 Recurrent Neural Networks (RNN)

Recurrent neural networks, also known as RNNs, are a class of deep neural networks designed to process sequential data, where the order of data points is important. To

achieve this, these networks define a recurrent relationship over the input sequence [99]. Unlike feedforward ANNs, where the output depends only on the current input and the network weights, the RNNs also use hidden state h_t, which is a function of the current input and the previous hidden state, h_{t-1}. The transition function is given by:

$$h_t = f\left(Ux_t + Wh_{t-1}\right) \tag{5.2}$$

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

The previous equation describes an RNN with a single layer. However, multiple RNN layers can be stacked, to create a stacked RNN. In this case the hidden state h_t^l at level l and timestep t, takes as input x_t^{l-1}, from the previous level, and h_{t-1}^l, of the previous time step. The Equation 5.2 is transformed to:

$$h_t^l = f\left(Ux_t^{l-1} + Wh_{t-1}^l\right) \tag{5.3}$$

where l is the layer number.

Because of the recurrent relation, RNNs can have memory over time, as each hidden state contains information about the previous steps. RNNs can be used to model a many-to-one type problem, which means to take as an input a sequence of time steps and classify it to a label. The recurrence relationship can be observed by unfolding the structure of an RNN. In Fig. 5.4 an RNN that takes an input x, that consists of n samples, and classifies it to an output y and its internal structure is presented. The weight matrix U transforms the input x into hidden state h, the weight matrix W transforms the hidden state of the previous timestep into the current hidden state, and weight matrix V maps the hidden state h to the output.

RNNs are trained in a similar manner to feed-forward ANNs, using an extension of backpropagation, the backpropagation through time. To compute the gradients, the network is unfolded over time and each time step is considered as a hidden network layer. The weights of the network are shared across time steps. During the backpropagatio, the gradient loss is computed with respect to the network parameters and differentiation

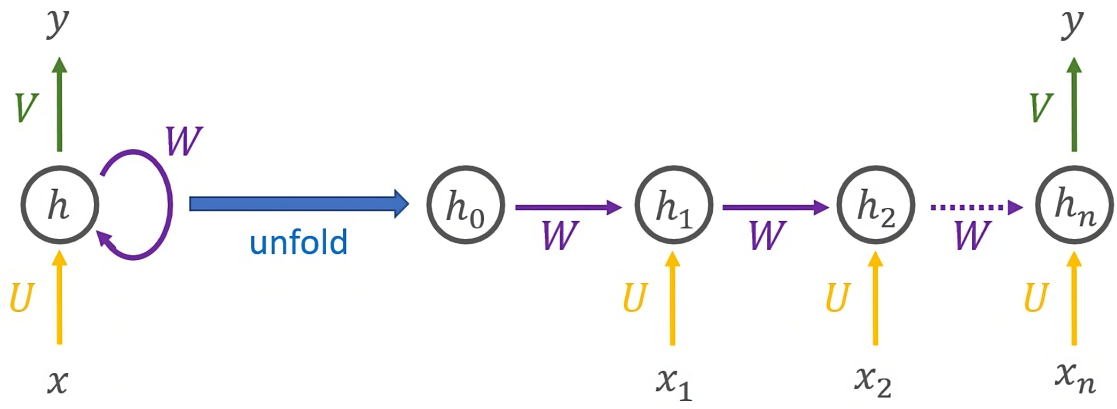

Figure 5.4: Schematic diagram of the RNN recurrence relation

through time. Then, the gradients are used to update the weights of the network using an optimization algorithm, such as gradient descent.

However, RNNs can suffer from a significant problem: vanishing or exploding gradients. The vanishing gradients problem refers to the phenomenon where the gradient values decrease exponentially over a number of time steps, resulting to extremely small values in the earlier states. As a result, the larger values of gradients from more recent time steps become dominant, hindering the ability of the network to retain information about earlier time steps. Exploding gradients refers to the opposite phenomenon, when the gradients take very large values. As the gradients are multiplied during back-propagation, if the resulted product is greater than 1, then the gradient values can increase exponentially. This can result to updating the network weights in a way that can decrease performance or lead to instability during training.

5.3.3 Long-Short Term Memory Networks (LSTM)

The LSTM networks [100], [101] are a kind of recurrent neural network that aims to resolve the vanishing/exploding gradient problem. The key characteristic of LSTMs is that they use a vector of memory cells $c_t^l \in \mathbb{R}^n$ with input, output and forget gates to maintain information for longer periods and regulate the flow of information. The gates are defined by a sigmoid function and element-wise multiplication. The forget gate decides whether parts of the existing cell state will be erased or not, and the input

gate decides what new information will be added to the memory cell. This gives the ability to LSTM to store more specific information in its cell state vector. The role of the output gate is to decide what the output of the cell will be. In a similar manner to the simple RNN, a LSTM network can be trained by back-propagation through time.

In this way, LSTMs can decide to overwrite the memory cell, retrieve it, or keep it for the next time step, hence maintaining both long and short term memory depending on the task and context. A schematic diagram of a memory cell is shown in Fig. 5.5.

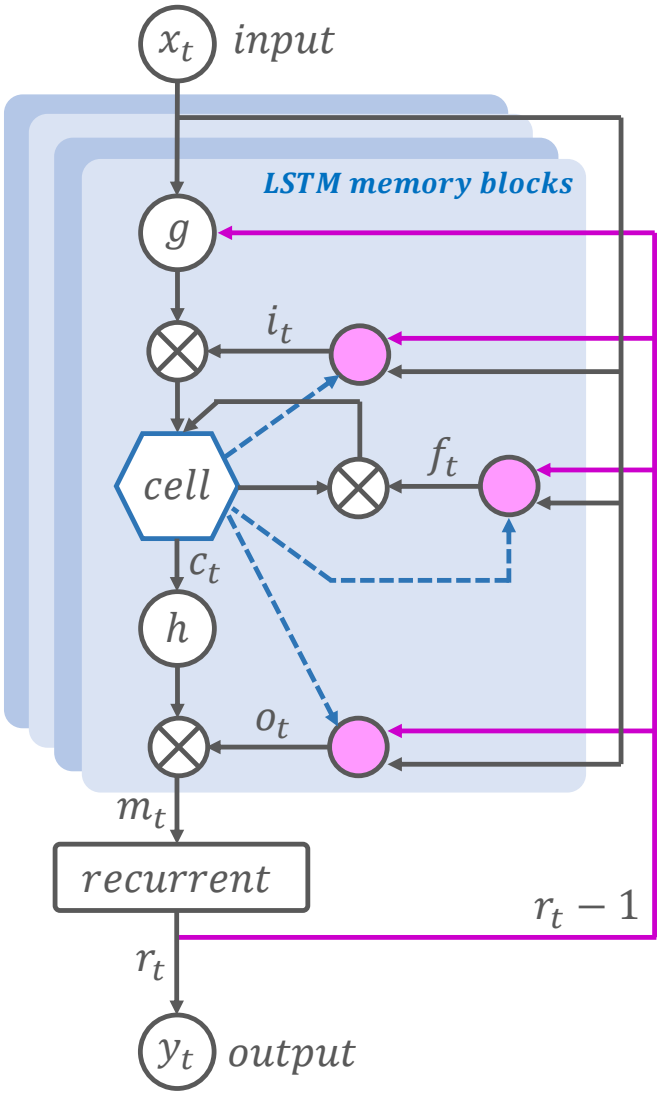

Figure 5.5: A schematic diagram of a LSTM memory cell.

RNNs, including LSTMs, can map one to many, many to many or many to one.

For example, given an input sequence $x = (x_1, \ldots, x_T)$ and target output sequence $y = (y_1, \ldots, y_T)$, the LSTM network unit activations can be calculated iteratively from $t = 1$ to T with the following equations [102]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \tag{5.4}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{5.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g\left(W_{cx}x_t + W_{cm}m_{t-1} + b_c\right) \tag{5.6}$$

$$o_t = \sigma\left(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o\right) \tag{5.7}$$

$$m_t = o_t \odot h\left(c_t\right) \tag{5.8}$$

$$y_t = \phi\left(W_{ym}m_t + b_y\right) \tag{5.9}$$

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

5.3.4 Using LSTMs to predict cascading events

Due to the particular properties of LSTMs explained previously, they offer a good fit for the problem of predicting cascading events. LSTMs can handle time-series data and their memory properties also fit well with the need to capture the evolution and inter-dependencies of the system variables as they evolve in time, an important aspect affecting cascading events. In order to predict the occurrence of cascading events, an LSTM model is trained using the pre-processed data, described in Sections 5.2.3 and 5.2.4, as input. The input X of the model is a $N_F \times N_T$ matrix, where N_F is the number of features and N_T is the number of time steps included in the selected time-window (input size). The time-window length is investigated in Section 5.5.1.

To pose the occurrence of a cascading failure as a binary classification problem, the final layer consists of a single neuron that is fed into a sigmoid activation function to output a value between 0 and 1, that represents the probability of a cascading event occurring or not. The sigmoid activation function is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{5.10}$$

The threshold is set to 0.5, if the output probability is higher than the threshold then Y is set to 1 (a cascading event will occur), otherwise Y is set to 0 (no cascading event). The binary cross entropy between the model predictions and real values (1 for failure cases, 0 for non-failures) is chosen as the loss function. The binary cross entropy is a typical loss function used for binary classification problems, and is given by:

$$BCE_{loss} = -(y \log(p) + (1 - y) \log(1 - p)) \tag{5.11}$$

where y is the actual label (0 or 1) and p is the predicted probability.

Each parameter's contribution to the total loss is computed via back propagation and batch gradient descent is performed to optimize the weights of the model parameters, as explained in more detail below. The structure of the proposed model is presented in Fig. 5.6.

5.3.5 Model Training

To train the LSTM models, we use the pre-processed dataset as outlined in Section 5.2.4 and perform a stratified split using a ratio of 80-10-10% to create training, validation and test sets. We use a single layer LSTM, where the number of hidden units/neurons is set to 150. The size of the hidden units is chosen based on model performance after performing a grid search for the following values: {50, 100, 150, 200, 250}. The Adam optimizer is used and the binary cross entropy as the loss function, a common choice for binary classification problems. To compare the performance of LSTM, we train additionally a feedforward Multilayer Perceptron (MLP) and a simple RNN network. The number of hidden units for the RNN is set after performing a similar grid search

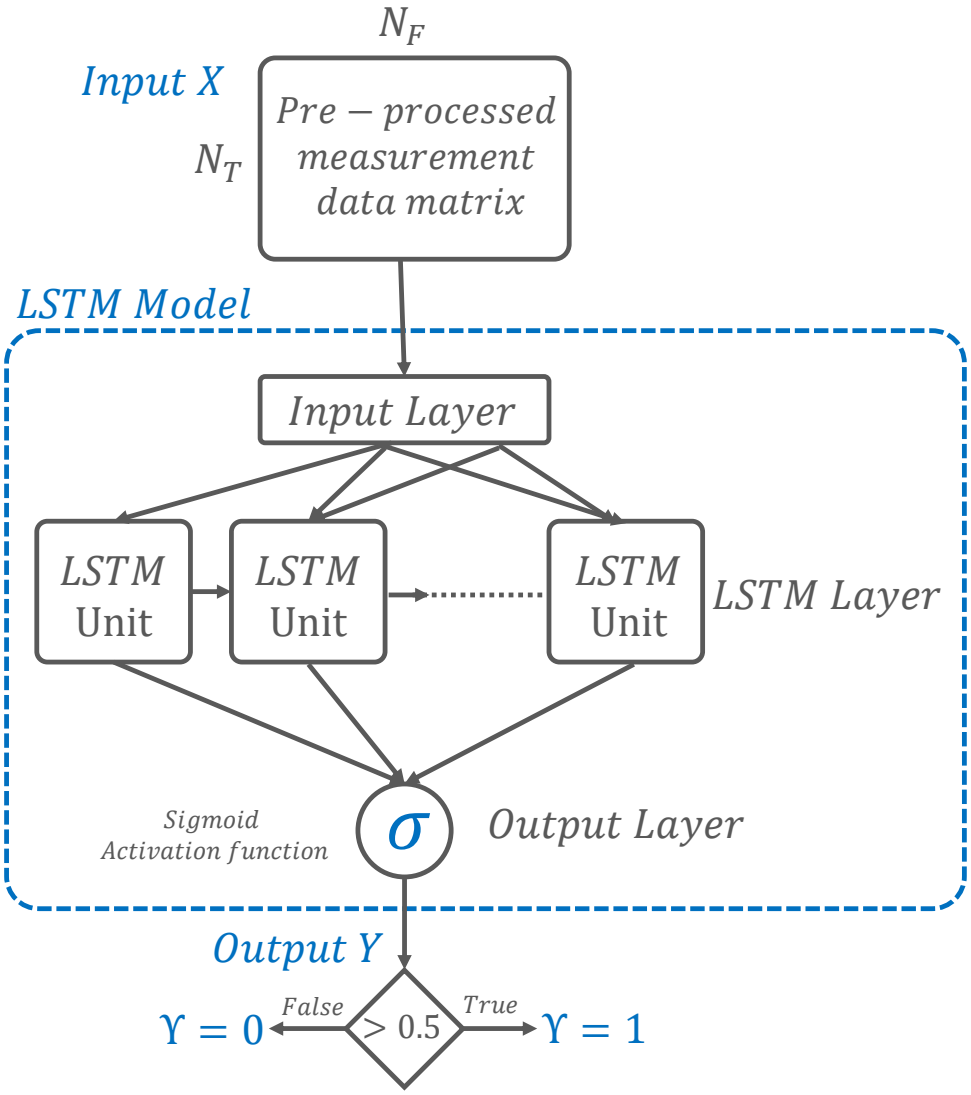

Figure 5.6: Structure of the LSTM model.

as for the LSTM model. As a baseline approach, the MLP consists of an input layer with the number of neurons set equal to the number of input data points (number of features × time steps) and a single hidden layer with 300 neurons, as set following a grid search. The rectifier linear unit (ReLU) activation function is chosen for the MLP, to capture the nonlinear behaviour. The ReLU activation function is given by:

$$Relu(z) = max(0, z) \tag{5.12}$$

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

Because of the randomness introduced during the initialization of weights and during the training of neural network algorithms, the same network trained on the same data can produce different results [103]. To ensure reproducibility, we set the model seed to 17 during the model training process. The feature importance and time window length experiments in Section 5.5 show that the implemented models perform well regardless of the data split.

As observed in Fig. 5.7, where the evolution of the training and validation loss is presented across the epochs, no over-fitting is observed and the model has converged towards the end of training. Moreover, we observed that models tended to overfit after 10 epochs (training loss decreases while validation loss increases). Once the model is trained, we perform inference on the test set and compare the predicted against the true labels.

5.3.6 Evaluation Metrics

To evaluate the performance of the proposed LSTM binary classifier, the metrics presented in (4.21-4.24) are used. Accuracy, Precision, Recall and F1 score are typical measures used in machine learning that capture different aspects of the performance of a binary classifier [86]. The mathematical definition of these metrics are given in

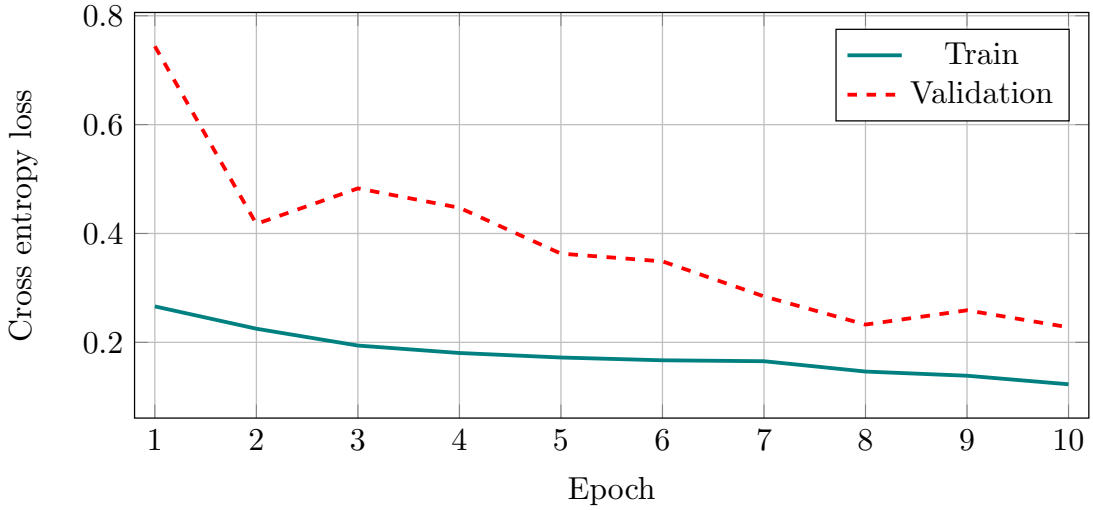

Figure 5.7: Train and Validation learning curves of the proposed LSTM model.

Section 4.2.4.

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

We note that there is almost always a trade-off between recall and precision with datasets of limited size. Models with high recall-low precision and low recall-high precision performances can be interpreted as over-fitting and under-fitting respectively. In this particular application, a false negative is more critical than a false positive as missing a real failure event might lead to subsequent cascading events or even a widespread blackout. Thus, a high *Recall* is more important in this case.

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

$$Accuracy' \ (\%) = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{FP} + n_{TN} + n_{FN} + n_{MC}} \tag{5.13}$$

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

5.3.7 Permutation Feature Importance

A permutation feature importance analysis is performed to identify the most important features according to the model prediction performance. In this case, these features represent time domain measurements that describe the measured electrical variables of the system, which in an online setting can be acquired by PMUs. To this extend, this analysis can help to identify which PMUs provide the most important information for the prediction of cases with cascading events.

The feature importance analysis is described in detail in Section 4.2.5. In this study, the same technique is adapted to features that represent time series windows. More specifically given a sequence of n timesteps, the time order of all features except the feature to be permuted remains the same while the selected feature column is shuffled, breaking the time-order. Since LSTMs are a type of RNNs that expect ordered time-series as input, a permutation of an important feature would cause a drop in accuracy, as described in [88].

5.4 Test System

5.4.1 Power Systems Dynamic Model

In this study, a modified version of the IEEE-39 bus model with RES units and protection devices is used, as described in Section 3.4.2.

5.4.2 Case studies

The dataset consists of the case studies as defined in the Base case of Section 3.4.3. In total, 44064 cases have been simulated in this study, with cascading events appearing in 7131 cases (16.2% of simulated cases). The percentage of cases with cascading events is higher compared to practical applications, as the lines that are disconnected

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

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

5.5 Results

5.5.1 Time window selection

In this study, a fixed length observation window approach is utilised, by training and testing the proposed model for various prediction times. In order to define this time constant, the time of the first cascading event needs to be investigated, as this defines the time window in which the prediction of whether a cascading event appears or not has to be made, i.e. before the first cascading event actually happens. Fig. 5.8 shows the time elapsed until the first cascading event occurs after the applied fault is cleared. After investigation, the first cascading event takes place at 0.5s-2.5s after the fault clearance in 98.8% of all cases. We observed that increasing the time window length to 0.6s leads to a significantly higher number of missed cases (98 cases - 6.34% of total cases). Hence, time windows longer than 0.5s are excluded from the model experiments.

To investigate the impact of time window length selection on $Accuracy'$, a single layer LSTM model with 150 hidden units has been trained for 10 epochs for different time window lengths and the results are presented in Fig. 5.9. Also, the number of the

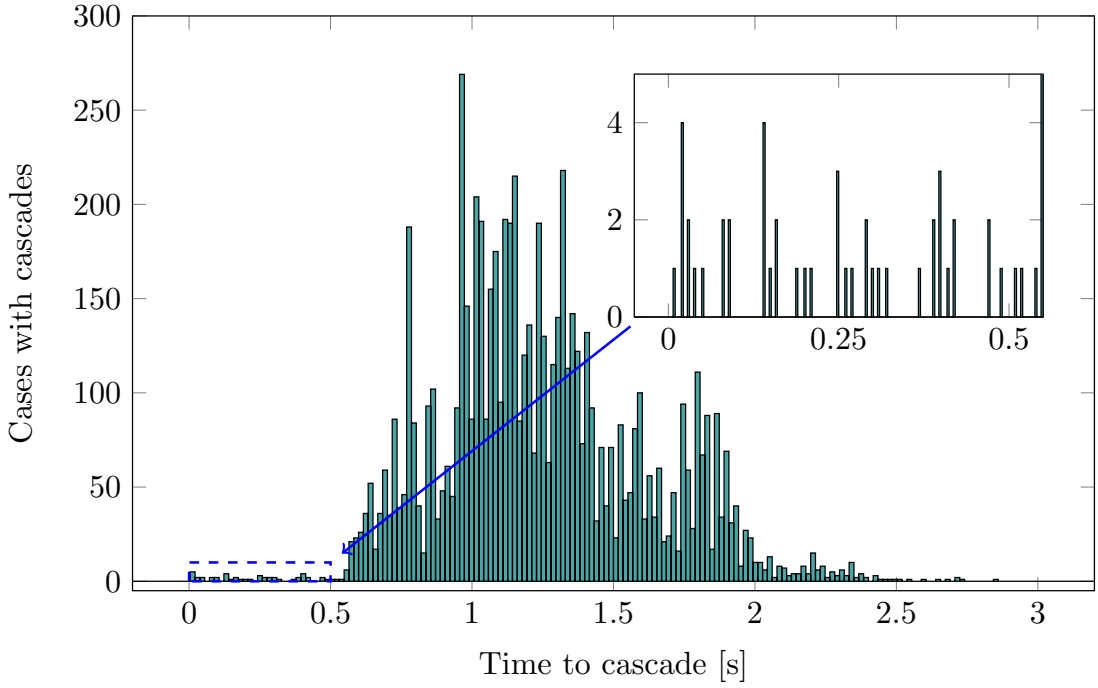

Figure 5.8: Time elapsed until first failure after fault clearance.

cases with cascades for which the first cascading event appears inside this time window (referred to as missed cases) is presented in Fig. 5.9.c). It should be noted that these missed cases have been excluded from the training and testing dataset, as the training of the model should not include measurement data during or after cascading failures. The results show that a time window of 0.1s results in the best *Accuracy*′, as driven by the low number of missed cases (13). Moreover, we observe that the number of false positives increase and the number of false negatives decrease as the time window length increases, leading to a loss in precision. Hence, the model exhibits a tendency to overfit when trained on data with window lengths of over 0.2s. We conclude that learning short-term trends and dependencies are more important for the predictive and generalization capabilities of the model.

5.5.2 Performance of online prediction

Following the previous analysis related to the time window selection, we use a time window of 0.1s to perform an online prediction analysis. The LSTM model performance

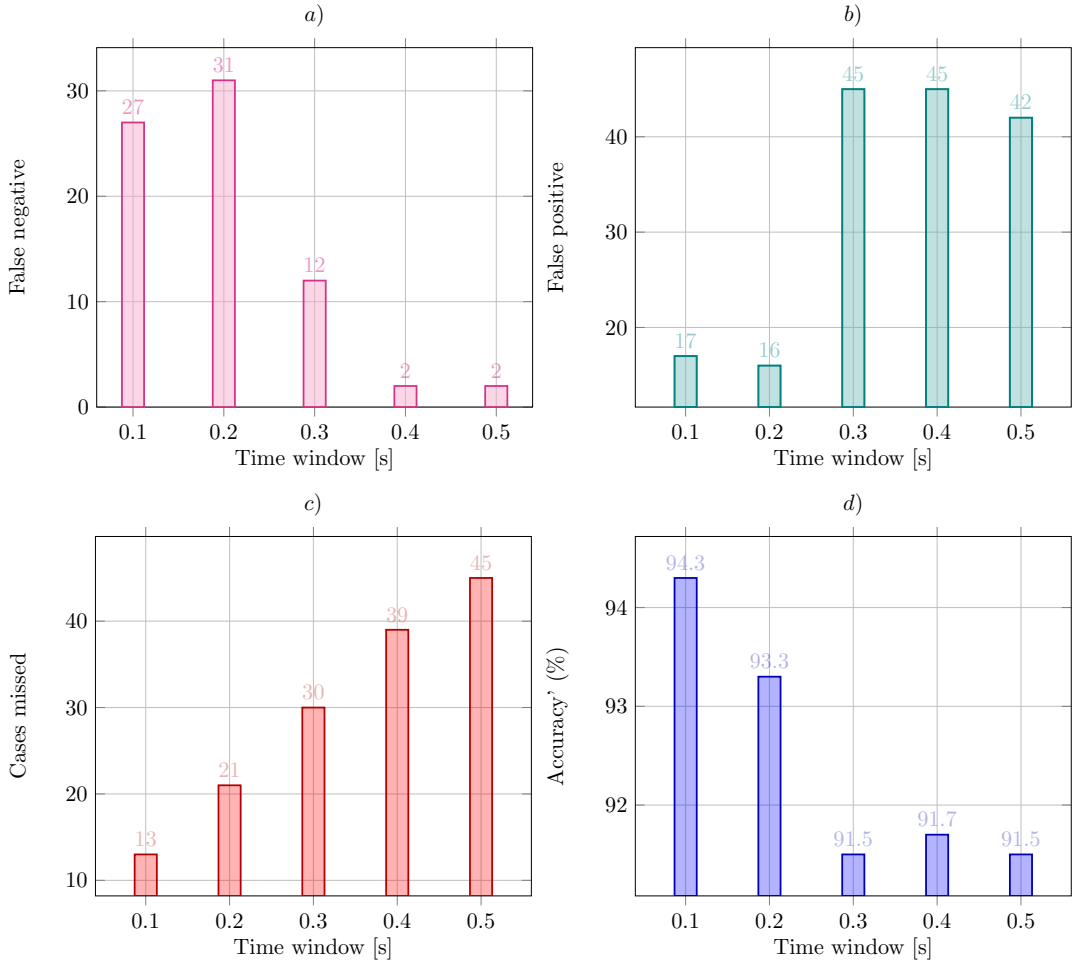

Figure 5.9: Impact of time window on online prediction.

is compared to the performance of a feed-forward MLP, as a baseline method, and a simple RNN model, which is another type of recurrent network configuration. The performance metrics of these models are presented Table 5.1. The metrics of the three models are calculated using the same test set of 1000 cases, which is pre-processed as described in Section 5.2.4 before being used as input to the trained models. The results show that the LSTM model exhibits the highest Accuracy Recall and F1 score. The MLP model shows a higher Precision than the LSTM model, but with a low Recall. As it is concluded, the LSTM shows overall the highest performance, and the following analysis is conducted for this model.

The confusion matrix (excluding missed cases) shown in Table 5.2 reveals that the

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

To further investigate the cause of false predictions, the boxplots of the Y output value of the model are presented in Fig. 5.10. This value represents the probability of whether a case includes cascading events or not that the LSTM model provides as output. For the false positive predictions, it is observed that the values are in the range of [0.5,1] and the median value is close to 0.8. For the false positive predictions, the range of Y values and the median value (0.4) are closer to the threshold (0.5). This indicates that the model predicts falsely a safe case as a case with cascading events with higher confidence than a case with cascading events as a safe case.

The tripping of a system element may cause the appearance of subsequent events, creating cascading event sequences of varying length. A summary of the cases with cascading events is presented in Table 5.3 in order to identify what is the impact of the correct or incorrect predictions on system security. The cases with cascading events that the model predicts correctly (true positive), have a mean value of 3.16 trips per sequence, and a mean value of 0.74% load loss. This percentage is calculated as the amount of load that gets disconnected because of the UFLS scheme to the total amount of system load at this case. In these cases, 239 SG units trip in total. These metrics showcase that the model is able to accurately predict cases with cascading events that have a high impact on system operation. All of the actual cases with cascading events that are falsely predicted as safe cases (false negative), include only one cascading event, the tripping of wind generator NSG2 due to over-voltage. So, as in this cases only one cascading event appeared and no amount of load is shed, the false prediction does not have a high impact on system operation.

Table 5.1: Trained models and result metrics.

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

Table 5.2: Confusion Matrix for the LSTM model.

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

Table 5.3: Cases with cascading events.

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

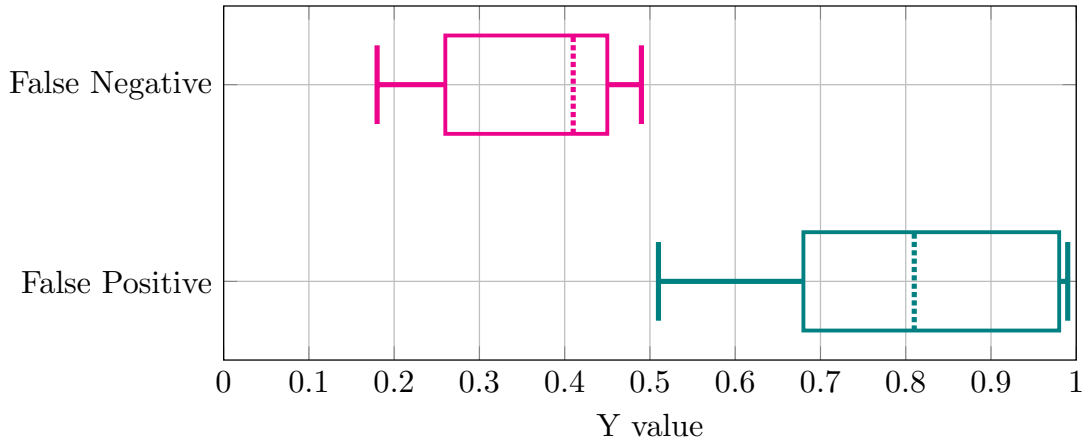

Figure 5.10: Boxplots of the model output Y for false predictions.

5.5.3 Impact of System Loading and Wind Generation on performance

The way that initial operating conditions affect the model performance can provide useful information about machine learning applications on power systems. In Table 5.4 the number of cases as correct or false predictions is presented for each system loading state appearing in the test dataset. Moreover, the accuracy that the model achieved at this system loading is also shown. In the test dataset there have been no case at

90% loading, represented by XX values in the Table. It is observed that as the system loading increases and reaches the nominal value (100%) the accuracy of the model improves. For this loading value, only cases without cascading events have appeared in the dataset, and the model predicts these cases more accurately.

Following a similar approach, an investigation on how the accuracy of the model is affected by the wind generation output is also presented. The percentage of wind generation can affect the amount of synchronous generation that is disconnected and the network topology. This consequently might affect the predictive power of the model due to changes in the appearance of particular cascading events, e.g. wind generator NSG2 has been shown to cause several trips related to voltage in this particular network and cases studied. The wind generation output percentage is expressed here as the percentage of the combined output of the three wind generators to the total nominal wind generation capacity (e.g. 100% wind generation output means that in this simulation the output of the three wind generators equals their nominal capacity). When the wind generation is lower (6.7%-26.7%, bars no. 2-6 in Fig. 5.11. there is a higher number of false predictions (41 cases in total). For these wind generation values, the appearance of cases that include only the tripping of wind generator NSG2 due to over-voltage are common, which the model falsely predicts as safe cases as explained previously. When the wind generation is higher (40%-100%) the model achieves a very high accuracy. The analysis in this Section highlights that machine learning model performance can vary for different operating conditions of the system and this is something that should be taken into consideration and could provide useful knowledge and potentially increased confidence when applying machine learning based methods.

Table 5.4: Impact of system loading on prediction performance.

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

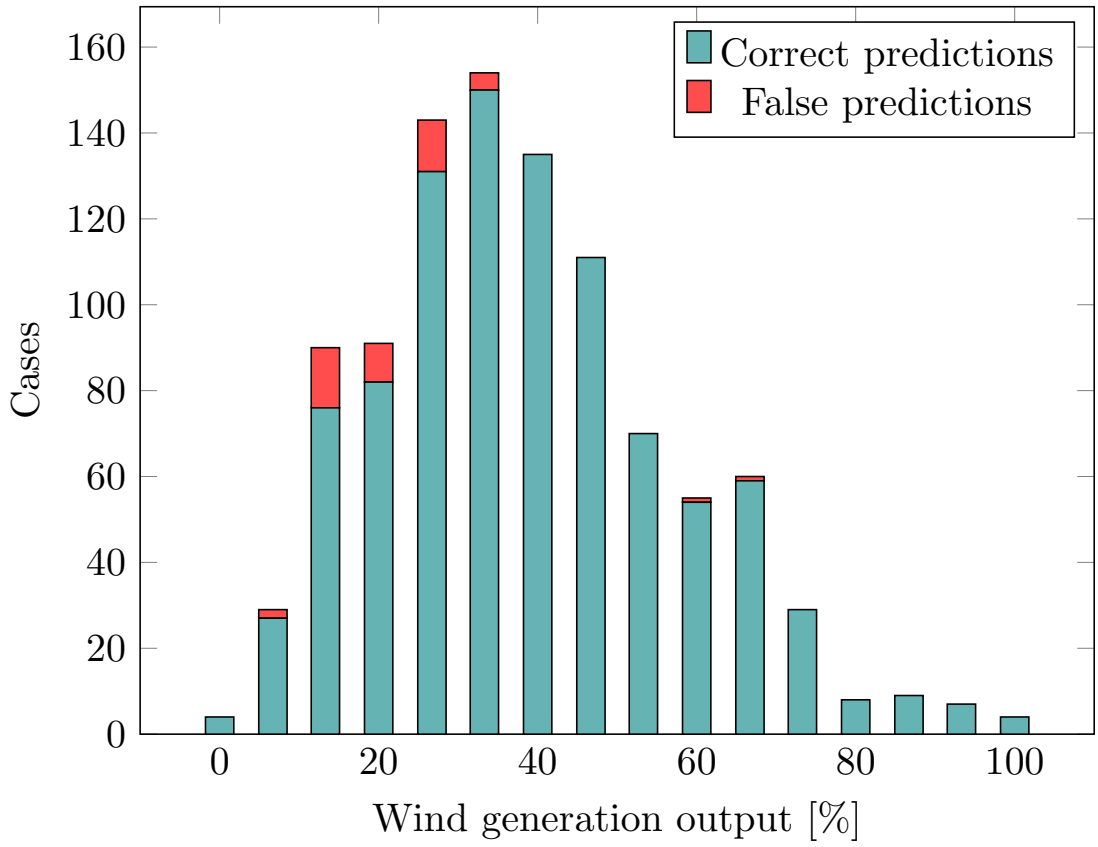

Figure 5.11: Wind generation impact on performance.

5.5.4 Feature Importance

After training and evaluating the model, a feature importance analysis using the permutation technique as described in Section 5.3.7 is performed, in order to identify which features, in this case representing PMU measurements, are mostly affecting the model performance. Because of the nature of neural networks, each feature acquires an individual weight and affects the training of the model differently.

In Fig. 5.12. the 20 features that when permuted result in the largest drop in the accuracy of the model are presented. These are the features that have the highest impact on the model performance, and therefore the most important ones. All but one of the most important features correspond to PMU measurements of active (14 features) and reactive power (5 features) on lines. The most important feature, which when permuted causes a 4.8% drop in the accuracy, is the active power measurement

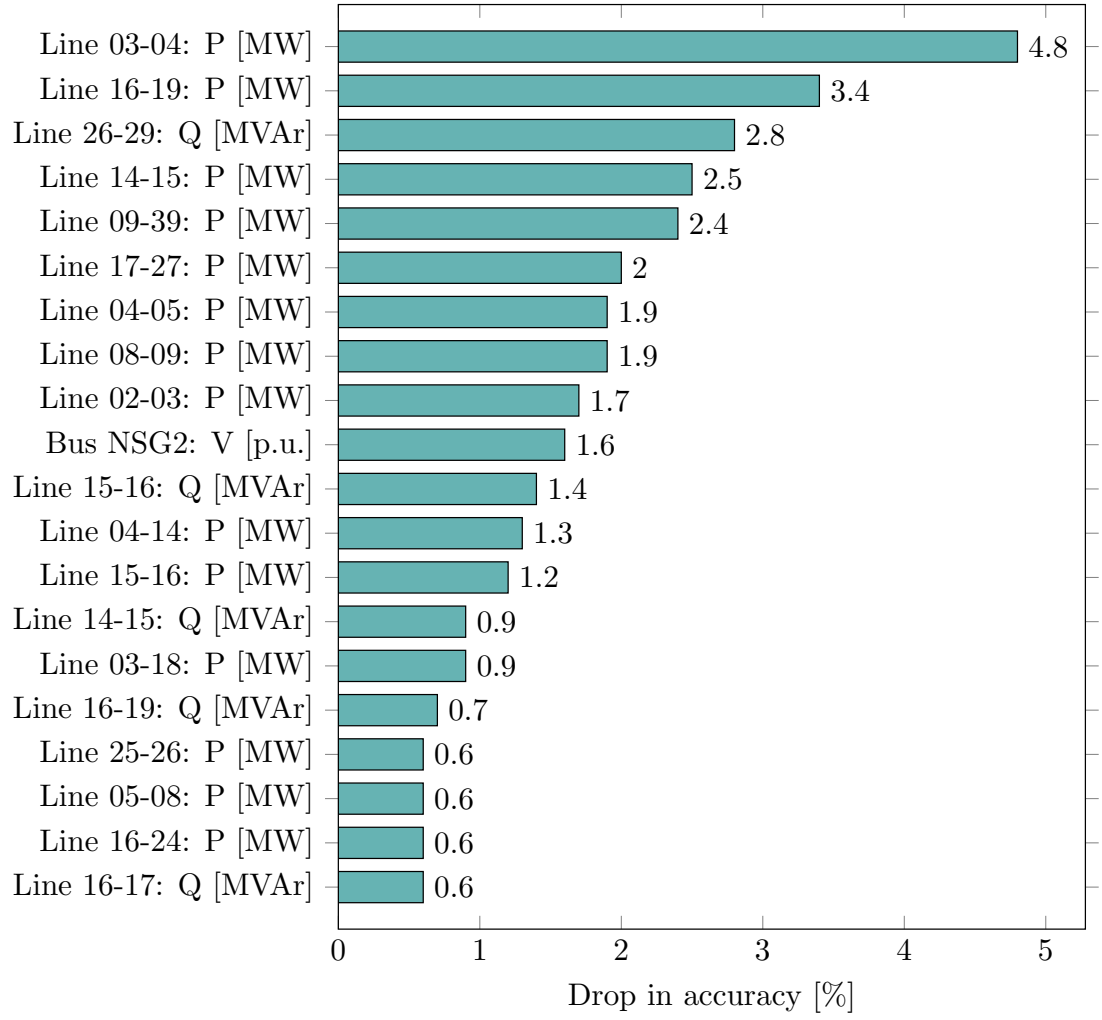

Figure 5.12: Permutation feature importance.

of Line 03-04, that connects two buses in the centre of the grid on which loads are connected. When disconnected, this line changes the network topology leading to an alternative flow of power. The second most important feature is the active power measurement of Line 16-19, which when is disconnected creates an islanded part of the system and causes the frequent appearance of cascading events. The only voltage measurement included in these features is that of the wind generator NSG2 bus, the tripping of which due to over-voltage is the most common appearing cascading event.

5.5.5 Considering availability and noise of PMU measurements

In large real-life power systems, the increased number of buses makes it infeasible to install PMUs at every bus of the system. For this reason, the performance of the model when limited PMU measurements are available is investigated. The proposed LSTM model is trained and evaluated using only the 10, 15 and 20 most influential features, as these have resulted from the feature importance analysis. The results in Table 5.5 show that when 10 and 15 features are considered, the model performs with 84% and 90.9% accuracy respectively (12.1% and 4.9% reduction in accuracy compared to the original LSTM model with 178 features). When the number of features is increased to 20, all of the model performance metrics improve, performing with 94.4% accuracy (1.26% reduction in accuracy). For this particular study it can be concluded that the model performance is satisfactory when including only the 20 most influential features. These 20 features can provide locational information about the buses at which the PMUs should be installed.

In practical applications, the PMU measurements may contain noise introduced by errors related to transducers and signal processing. The pre-trained LSTM model with 178 features is tested using test data measurements with added noise signal. The noise in PMU measurements is simulated by Additive white Gaussian noise (AWGN) with a standard deviation of 0.002 p.u. [104]. The results show that the added noise has no effect on the model performance, as the performance metrics, in Table 5.5, are identical to those of the original LSTM model without added noise. This highlights the robustness of the proposed method, as PMU measurements with noise do not affect its performance.

Table 5.5: Performance considering limited availability and noise of PMU measurements.

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

5.5.6 Computational time and Practical considerations

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

During the online stage, the average time that a single prediction takes after performing 1000 predictions on the GPU using the pre-trained model is 0.042s, which highlights the fitness of the method for real-time prediction as a fast model response at this stage is critical. Also, it showcases the ability of machine learning estimators to respond significantly faster, compared to the running time of a time-domain simulation.

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

5.6 Chapter Conclusions

This chapter introduces a framework for the online identification of cascading events in power systems with renewable generation using measurement data and supervised machine learning, namely LSTMs, a type of RNN. Dynamic RMS simulations on a model with protection devices included have been performed, in order to capture cascading events that appear, which are defined by the action of the protection devices.

Simulation data are pre-processed to represent typical PMU data and are used to train a LSTM based model. The pre-trained model is then used to predict online the appearance of cascading events and various aspects of its performance are analysed, including the time window selection, important features and how the performance is affected for different operating conditions. The framework is applied on a modified version of the IEEE-39 bus system, including wind generators and protection devices.

Results show that the proposed approach performs with a 95.6% accuracy within a short fixed-time window (in the order of 0.1s) following the initial fault clearance, showing improved performance compared to other neural network configurations (MLP and RNN). The model has the ability to predict the appearance of cascading events sequences, as opposed to only early instability violations that is a common approach in existing online prediction methods. After further investigation, the performance of the method appears to vary with the initial operating conditions, either improving or deteriorating. Such behaviour should be taken into account in order to inform the confidence to similar methods when considering real power system applications. Finally, the results of the feature importance analysis highlight important system variables that improve the model performance, with offering useful information in terms of monitoring requirements as well as system variables that are related to the appearance of cascading events. For this particular network, active and reactive power measurements of lines have a high impact on the prediction of cascading events. Also, the monitored power system variable that causes the most common cascading event is identified as an important feature. Tests considering limited available PMU measurements and noise in signals have little impact on model performance, verifying that the suggested approach is appropriate for practical applications.

Chapter 6

Predicting the Reason of Cascading Event sequences in Power Systems using Deep Learning

6.1 Introduction

6.1.1 Motivation

As the findings from the previous Chapter have highlighted, ML models that have the ability to learn dependencies from time-series data can utilize measurement data provided by PMUs and predict accurately and efficiently the appearance of cascading events in power systems with RES. However, the framework introduced in the previous Chapter is able to predict the appearance of cascading events in a binary fashion, meaning it can predict if cascading events will appear or not, after an initial fault.

Following the appearance of a disturbance, a series of subsequent cascading events in various timescales might appear, the propagation of which can lead to load shedding or even blackouts [2]. In addition, in a cascading events sequence multiple types of instabilities might be involved (e.g. a cascading event due to transient instability might

be followed by a cascading event due to frequency instability), as Chapter 3, that focus on the investigation of cascading events, has revealed. Taking the previous points into account, the framework introduced in this Chapter aims to approach the online identification of cascading events by a multi-class perspective, by predicting the reason of the cascading event that follows in a sequence. This can provide system operators with additional information to effectively implement targeted security measures in order to halt the progression of cascading events or minimize their consequences.

The approach presented in this Chapter is focused on the fast phase of cascading events sequences, during which, according to the analysis in [5], the fast succession of events makes it impossible for system operators to take any manual measures for the mitigation of cascading events. The proposed method aims to predict the reason of cascading events using measurement data in real-time, providing valuable information about the types of instabilities involved. This information can enable system operators to take preventive or corrective actions in order to mitigate or avoid the impact of cascading events.

The studies reviewed in Chapter 2 focus on predicting only a particular type of instability. In addition, the action of protection devices is not accounted for, which in reality might change the actual trajectory of state variables following a major disturbance and can even lead to subsequent trips. As it is highlighted in [96], including the action of protection devices in dynamic studies is of significant importance to accurately capture system behaviour. In [105] a method based on a Graph Convolution Network (GCN) that contains information about spatio-temporal properties is proposed for the prediction of cascading failures, in a dynamic network with renewable generation and protection devices. However, the method focuses only on the prediction of the appearance or not of cascading failures, and not on predicting the reason of cascading failure that follows in a sequence.

6.1.2 Contribution

The main contribution of this chapter is the use of supervised deep learning methods to predict the reason of upcoming cascading events as they appear in sequences capturing

detailed dynamic behaviour (limited to the RMS framework). According to the existing literature review, this is the first time that a method to predict the reason of upcoming cascades is being proposed. More specifically the contributions of this chapter include:

- The detailed dynamic behaviour represented in the RMS framework is accounted for, including mechanisms and controllers related to angle, voltage and frequency stability and more importantly the action of protection devices. The impact of renewable generation interfaced via converters is also captured. This leads to a more accurate and detailed representation of the events leading to potential load shedding or system collapse. The reason of cascading events that follow is predicted, and not just if a cascading event will appear or not, which can give to system operators more information about taking targeted corrective, or even preventive measures to stop the evolution of cascading events or mitigate the impact.

- The proposed method utilises Temporal Convolutional Networks (TCNs) combined with a moving window approach to predict the evolution of the sequence of cascading events. This is achieved by formulating the above as a multiclass classification prediction problem for the reason of the upcoming event. The prediction goes beyond to identify the reason of the violation, e.g. a trip related to over-voltage or under-frequency by defining seven classes.

6.2 Online prediction of the reason of upcoming cascading events

6.2.1 Detailed Procedure

The method proposed in this chapter aims to the prediction of the reason of cascading event sequences in an online manner, as they progress. The main steps of the framework are presented in Fig. 6.1. In this approach it is formulated as a multi-class classification problem, consisting of an offline and online phase. During the offline phase, the dataset generation and the training of the model take place. After using an efficient sampling

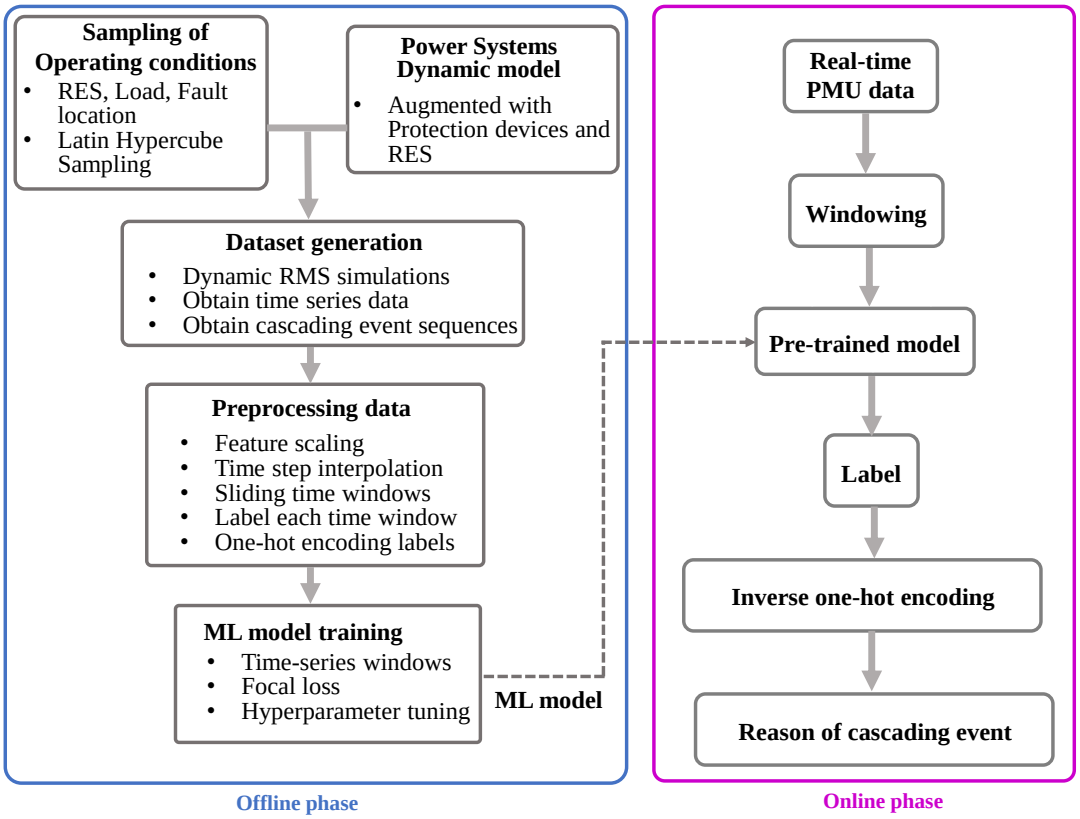

Figure 6.1: Main steps of the proposed framework.

method for selecting the initial operating conditions, a number of dynamic Root Mean Square (RMS) simulations is performed, taking into consideration different levels of RES penetration and system loading. The initial applied contingency may cause the appearance of cascading events, which in this study are dictated by the intentional action of protection devices when the network conditions violate their limits. In this case, the sequence in which these cascading events appear is obtained, and each cascading event is characterized by the reason of tripping, the component that trips and the time of the event.

The time-series data obtained from these dynamic simulations describe the electrical system variables as they evolve in time. The time domain features are pre-processed to represent typical PMU data and are formed into equal length moving time windows. Each time window is labeled according to the reason of the cascading event that follows. Then, the time windows and their labels are used for creating training, validation and

testing datasets for the machine learning model.

During the online phase in a real-life scenario the time-series data can be obtained from PMUs [97] and after forming them into time windows can be used as input to the pre-trained model to predict the reason of the next cascading event in the sequence. In this study, only simulated measurement data have been used for training and testing the model. The models have been evaluated on the test dataset using the accuracy metric and individual metrics for each class. Also, the performance of the model per cascading sequences is analyzed.

6.2.2 Sampling of Initial Operating Conditions

In order to sample the initial operating conditions, in this study wind generation and system loading, the technique of Latin Hypercube Sampling (LHS) [106] is used, as opposed to the discretization with a certain step approach used in Section 3.4.3. A square grid containing sample positions is a Latin square if there is only one sample in each row and each column. LHS is the generalisation of this concept to a higher number of dimensions. The sampling points are selected in such a way that the minimum distance between them is maximized in the multidimensional space, resulting in an effective coverage of the search space.

The advantage of this method is that LHS ensures that the set of samples is representative of the input variability whereas random sampling returns a set of random samples without any specific rule. The application of LHS in a two dimensional space, for illustrative purposes, and a comparison to random sampling is shown in Fig. 6.2. As it can be observed, compared to random sampling LHS can cover more effectively the search space defined by the two parameters x_1 and x_2.

For power system stability/security studies, it is important to sample evenly all the areas of the search space [107], [108]. In this case, LHS is applied on a n dimensional space to ensure representative sampling points. The $n-1$ dimensions represent the output of each one of the network wind farms and one dimension represents the loading of the system.

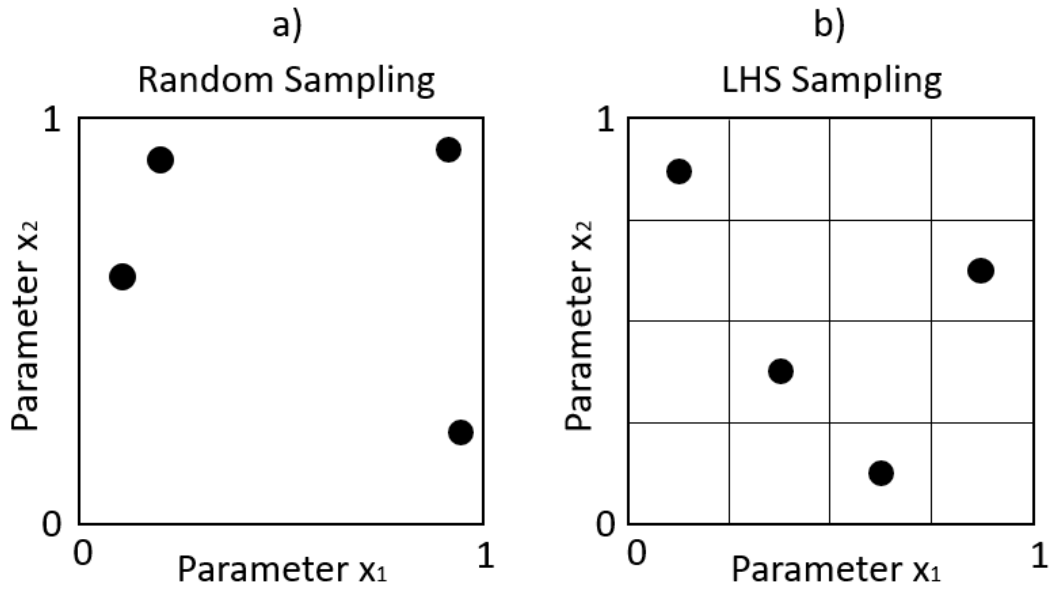

Figure 6.2: Sampling on a two dimensional space using: a) Random sampling, b) LHS sampling

6.2.3 Dataset Generation

After determining the initial operating conditions for renewable generation and system loading using LHS, an AC OPF problem is solved to determine the dispatch of the SGs and perform the dynamic RMS simulations, as described in Section 3.4.1. The time series data of each simulation is obtained, with time domain features describing the states in various power system locations over time, as in Section 5.2.3. If after the end of each simulation cascading events appear, the cascading event sequence is obtained and each cascading event is characterized by the component that trips, the reason of the event and the time that the event appears.

6.2.4 Pre-processing Data

Following the dataset generation from the dynamic simulations we perform scaling, time step interpolation and windowing on the time series data to ensure that the pre-processed data are appropriate for machine learning applications and representative of PMU measurements. After application of scaling in this study all the measurement values are in between 0 and 1. The scaling transformation for each individual feature

X_F is given by:

$$X_{Fscaled} = X_F - min(X_F)/(max(X_F) - min(X_F)) \tag{6.1}$$

where $min(X_F)$ and $max(X_F)$ are the minimum and maximum values respectively of individual feature X_F across all time steps.

Next, we perform interpolation to ensure evenly sampled time steps across all data samples, by applying first order spline interpolation, with the time interval δ set to 0.01 seconds.

After these pre-processing steps, the simulation time series data are divided in windows in order to generate evenly sized data samples. A major advantage of windowing is that it enables real-time predictions as model inference is performed on a small window, an important aspect due to the online nature of the proposed method that in a realistic setting requires quick predictions. In this case, a windowing function with a window length of 10 time steps is used. A diagram of the windowing process can be seen in Fig. 6.3.

Each time window is assigned a label, according to the reason of the next cascading event. As next cascading event it is defined the event in a sequence that appears after the last time step of the time window. In the example in Fig. 6.3. the label of the first time window, *Label* 1 is assigned according to the reason of the cascading event that appears in t_{12}. If cascading events appear later than the next time window then the appropriate label is still set. If no cascading event appears after the time window then the label equals to 0.

In this case, there are 7 possible cases of cascading event reasons, and each one of them represents a class. These classes are defined as: No cascading event, Out-of-step, Over-frequency, Over-voltage, Under-frequency, Under-voltage and Distance. The classes are one-hot-encoded to bring them to a suitable format for machine learning applications. One-hot-encoding is a common technique applied in machine learning problems to convert categorical data to binary vectors [109]. Following dataset generation and pre-processing, the training/testing sets for the machine learning models described in Section 6.3 are derived.

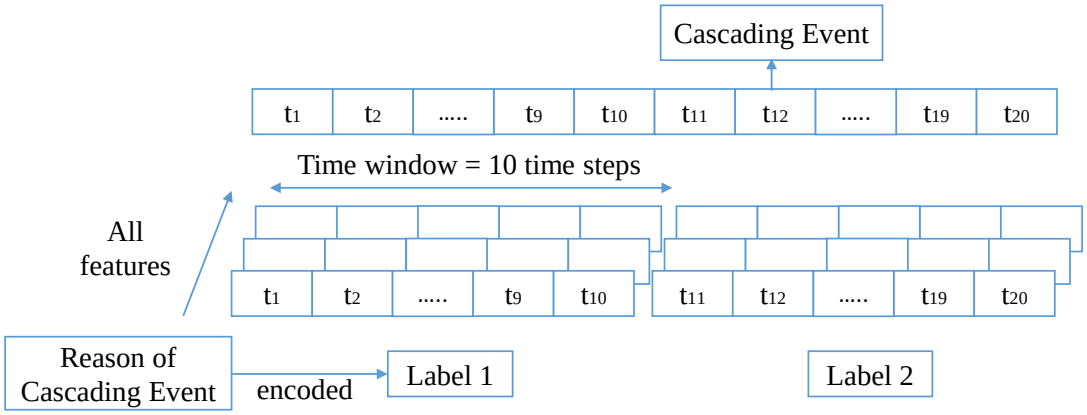

Figure 6.3: Time windowing process.

6.3 Using Deep Learning to predict cascading event sequences

6.3.1 Deep Learning Models

For the specific problem of the prediction of the reason of upcoming cascading events, machine learning models with appropriate architecture for sequential data and consequently time series, are proposed. This makes it possible for the models to capture the evolution and inter-dependencies of the network variables in time, which describe the reasons of upcoming trips as part of a cascading sequence. A comparison between possible appropriate architectures of RNNs, LSTMs and TCNs is provided, with TCNs being proposed due to theoretical and performance related aspects elaborated below. The proposed models are trained using the pre-processed data formed into time windows as input. The input size of all the models considered is a $N_F \times N_T$ matrix, where N_F is the number of features and N_T is the number of time steps included in the selected moving time-window.

The prediction of the reason of a cascading event is formulated as a multi-class classification problem, where each time window is assigned a label based on the reason for the upcoming cascading event. The time window is moving online as measurements are obtained. For all the deep learning architectures considered in this study, the output layer is a dense layer with 7 neurons with a softmax activation function. The softmax

function outputs a value between 0 and 1 for each one of the 7 classes, that represents the probability of the reason of the next cascading event. The class with the highest probability is selected using the arguments of the maxima (argmax) as the final model output. The output value is inverse one-hot encoded to convert it back to the reason of cascading event.

6.3.2 Recurrent Neural Networks

RNNs are a type of deep learning neural network designed to model data with temporal qualities where the order of data points is important. A detailed descriptions of RNNs and LSTMs is presented in Section 5.3.2. In this study three different RNN architectures are implemented and evaluated.

Stacked LSTMs consist of more than one layers with LSTM cells that are stacked on each other. In this architecture, the output of a LSTM layer is fed as an input to the next LSTM layer, with the final output of the model being obtained from the last layer. Stacking multiple LSTM layers allows for higher complexity models that can learn complex relationships of the input data, with each layer building on the representations learned by the previous layer.

Bidirectional LSTM (BiLSTM) is a type of LSTM network where the input flows in both directions. This network architecture consists of two LSTM layers, one processing the input sequence in a forward direction and the other one processing the input sequence in a backward direction. The output of the BiLSTM model is obtained by concatenating the output of the forward and the backward direction LSTM layers. Thus, it is capable of utilizing information from both directions of the sequence. In the context of cascading events, this would be useful as the model would have information about both directions of the time series measurement data. The forward hidden state h_t^f, that considers the input in ascending order (i.e. t = 1 to T), the backward hidden state h_t^b, that considers the input in descending order (i.e. t = T to 1), and the output y_t are calculated by:

$$h_t^f = \tanh(W_{xh}^f x_t + W_{hh}^f h_{t-1}^f + b_h^f) \tag{6.2}$$

$$h_t^b = \tanh(W_{xh}^b x_t + W_{hh}^b h_{t-1}^b + b_h^b) \tag{6.3}$$

$$y_t = W_{hy}^f h_t^f + W_{hy}^b h_t^b + b_y \tag{6.4}$$

where W denote weight matrices and b denote bias vectors.

Gated recurrent units (GRUs) are a type of RNN that use a gating mechanism to control the flow of information through the network [110]. The structure of GRU cells is similar to that of LSTM cells, but with two gates: an update gate and a reset gate. The update gate is a combined version of the forget and input gates of LSTM, and decides which information to forget and which to include, while the reset gate determines how much of the previous hidden state to retain. As a result, a GRU model has fewer parameters than a LSTM and is more computationally efficient. The reset gate r_t, the update gate z_t and the hidden state h_t are given by:

$$r_t = \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r) \tag{6.5}$$

$$z_t = \sigma(W_{xz} x_t + W_{hz} h_{t-1} + b_z) \tag{6.6}$$

$$\tilde{h}_t = \tanh(x_t W_{xh} + (r_t \odot h_{t-1}) W_{hh} + b_h) \tag{6.7}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{6.8}$$

where W denote weight matrices, b denote bias vectors and $\tilde{h}_t$ is the candidate hidden state.

6.3.3 Temporal Convolutional Network

While RNN architectures (either LSTMs or GRUs) have shown promising results in time series applications, including stability prediction in power systems [15], they are usually very computationally intensive with a large number of parameters. Convolutional networks have many applications on datasets containing images [111], but have also been applied on sequences through the last decades. The Temporal Convolutional network (TCN) is an architecture designed for sequence modelling [112]. This architecture uses a 1-dimensional Convolutional network, where the size of each hidden layer

is the same as the input size. The kernel is a matrix that slides over the input data through 1 dimension, in this case the time, and it performs the dot product with this region of input data. The output of this convolution process is calculated as the matrix of dot products. More specifically, the 1 dimensional CNN takes as input a matrix with dimensions $N_F \times N_T$ where N_F is the number of features and N_T is the length of the time window considered.

Similar to an RNN, the TCN model can be used to model a many-to-one type problem, meaning it can get a sequence of data as an input and generate a label for classification. The TCN uses a 1D fully-convolutional network (FCN), where the size of each hidden layer is the same as the input size. FCN is a network that does not contain any Dense layers, as in traditional CNNs. Dense layer is a regular neural network layer with a deep connection, meaning that every neuron of this layer is connected to all the neurons of the previous layer. Instead FCN contains 1×1 convolutions that perform the task of the dense layers, which results in reduced complexity [113].

To ensure that there is no leakage from future time steps to the past, TCN uses causal convolutions, which are convolutions that for the output at time t only data points from time t and earlier in the previous layer are considered. The architecture of the TCN can be expressed as: $TCN = 1D\ FCN + causal\ convolutions$.

The receptive field of CNNs is the region in the input space that a particular CNN's feature is affected by. Normally CNNs have many layers in order to extend the receptive field, but at the expense of increasing the number of parameters. The proposed architecture in this study employs dilated convolutions to extend the receptive field while using fewer layers and consequently reducing the computational complexity. A dilated convolution is a convolution where the filter is applied over an area larger than its length. This is achieved by skipping input values with a certain step. For a 1-dimensional sequence input $x \in R^n$ and a filter $f : \{0, ..., k-1\}$, the dilated convolution F on an element s of the sequence is given by:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{6.9}$$

where d is the dilation factor, k is the filter size, and $s - d \cdot i$ is the direction of the

past.

More than one layers of dilated causal convolutions with different dilation factors can be stacked. Using a larger dilation the output at the top level can represent a wider range of input points. A schematic of a dilated causal convolution with dilation factors d = 1, 2, 4, 8 is presented in Fig. 6.4. A dilated convolution with dilation of 1 equals to a regular convolution.

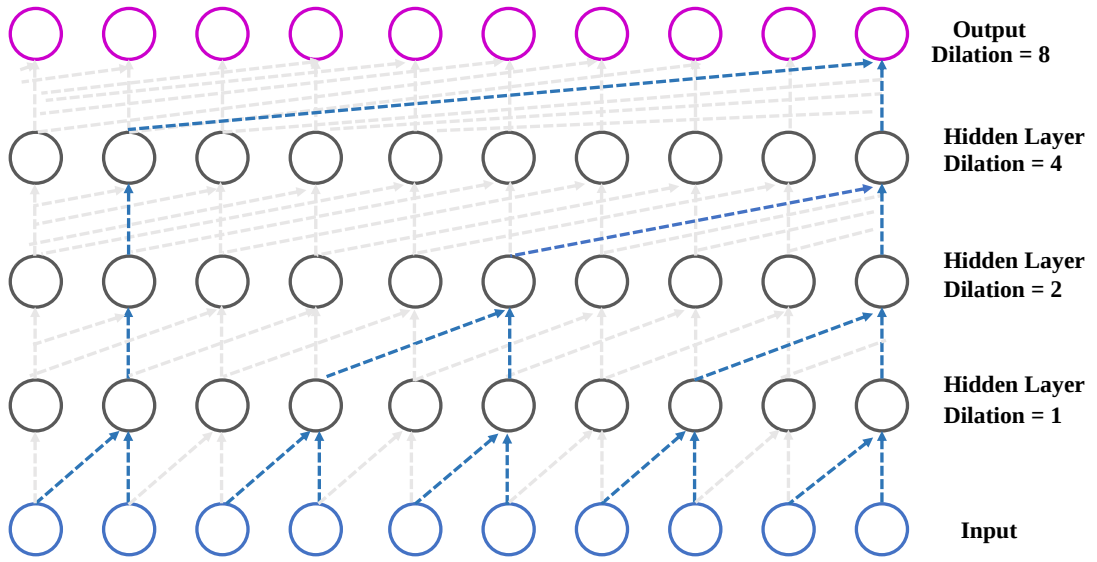

Figure 6.4: A stack of dilated causal convolutional layers.

The main advantage of TCNs compared to RNNs, is that because convolutions use the same filter in each layer, the computations can be done in parallel, where in RNNs the predictions for later timesteps must wait for the previous timesteps to complete. As a result, an input sequence of time steps during the training or the evaluation phase can be processed as a whole in TCN, where this is processed sequentially in RNN. Also, due to the propagation path of TCNs, they do not suffer from the vanishing or exploding gradients problem that can appear in vanilla RNNs.

In addition, some architectures of RNNs, like LSTMs, have multiple cell gates. In order to store the partial results for these gates, they can utilize a lot of memory space. In a TCN the filters are shared across a layer, and the backpropagation path depends only on the depth of the network. In practice, as it is concluded in [112], gated

RNNs can use up more memory than TCNs. The simplicity and lower computational requirements of TCNs, along with the ability to capture dependencies in time efficiently, are very useful characteristics for the particular problem of moving window prediction of an upcoming cascading event.

6.3.4 Evaluation Metrics

The evaluation of the proposed models is performed on the same test dataset, consisting of unseen data. Accuracy, Precision, Recall and F1 score are common metrics used in machine learning classification problems that capture the performance of a model [86].

The overall *Accuracy* of the model, describes the percentage of correct predictions of the reason of cascading events.

$$Accuracy\ (\%) = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{6.10}$$

In order not to rely only on an aggregated metric, the following metrics are calculated individually for each class, to identify possible challenges with the prediction of certain classes [86]. $Precision_i$ describes the percentage of correct predictions among all the predictions for a particular class $i \in \{1, .., K\}$, with K representing the number of classes. $Recall_i$ is the percentage of cases of a particular class that are predicted as belonging to that class. $F1\ Score_i$ is defined as the harmonic mean of $Precision_i$ and $Recall_i$.

$$Precision_i\ (\%) = \frac{n_{TP_i}}{n_{TP_i} + n_{FP_i}} \tag{6.11}$$

$$Recall_i\ (\%) = \frac{n_{TP_i}}{n_{TP_i} + n_{FN_i}} \tag{6.12}$$

$$F1\ Score_i\ (\%) = 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i} \tag{6.13}$$

where n_{TP_i}, n_{FP_i}, n_{TN_i} and n_{FN_i} is the number of true positive, false positive, true negative and false negative predictions for each individual class respectively.

The confusion matrix is also used to investigate the performance of the classification algorithms. Each row of the confusion matrix represents the actual labels of each class, while each column represents the predicted labels of each class. The n_{TP_i} are the values where the actual value and predicted value are the same. The n_{FN_i} for a class is the sum of values of the corresponding row for this class except for the n_{TP_i} value and n_{FP_i} for a class is the sum of values of the corresponding column for this class except for the n_{TP_i} value. The n_{TN_i} value for a class is calculated as the sum of values of all columns and rows except the values of the corresponding rows and columns of the particular class that is investigated.

In the context of cascading events reasons, true positives are the correct predictions of the reason of cascading events and false positives are the predictions of any other incorrect reason that are indicated as this reason. True negatives of a particular class are the when the model correctly predicts a different reason, and false negatives of a particular class are when the model incorrectly predicts a different reason from the actual reason.

6.3.5 Focal loss function

Due to the nature of associated phenomena, some cascading events can appear more frequently than others, according to the network vulnerabilities. This leads to a different number of samples for each reason of cascading events in the training dataset. Due to the imbalance of the classes in the dataset, categorical focal loss is used as the loss function during the training of all the models. Focal loss [114] is a loss function designed to focus on hard to classify examples. The focal loss is defined as:

$$FL = -(1 - p_y)^{\gamma} log(p_y) \tag{6.14}$$

where $y \in \{0, 1, .., K-1\}$ is an integer class label with K representing the number of classes and p_y is the predicted probability vector. The focusing parameterγ is a modulating factor that reduces the loss contribution from easy-to-classify examples. By adjusting γ, more focus is given to hard to classify examples. When $\gamma = 0$, Focal Loss is equivalent to the Categorical Crossentropy loss function. In this study $\gamma = 2$

has shown to produce the best results.

It should be noted here that, other common techniques for imbalanced datasets, like under-sampling, over-sampling, and generation of synthetic time data using Generative Adversarial Networks (GANs) for the minority class have been tested but in this particular case have not improved the models performance. Under-sampling has led to losing important information during training. Over-sampling and synthetic data have not affected the performance. Also, using balanced class weight according to the distribution of each class has deteriorated the model performance on the classes with more samples, which in this case is important.

6.4 Test System and Application

6.4.1 Power Systems Dynamic Model

The proposed framework is showcased on a modified version of the IEEE-39 bus model as described in Section 3.4.2. In this Chapter the network model has been augmented with a distance protection scheme, which has been implemented by adding two distance protection relays at the ends of each line. The distance protection has two zones of protection. The first zone is set at 80% of the line's reach with no delay time. The second zone of protection acts as a backup zone protection and it is set at 120% of the line's reach with a delay of 400ms. An inter-tripping Permissive Under-reaching Transfer Trip (PUTT) scheme between the relays of the same line has also been modelled. This inter-tripping scheme enables communication between the relays of the same line. This feature is particularly useful when a fault happens near either end of the line, and one relay identifies the fault as a first zone fault, and the other relay identifies it as second zone fault. Through this scheme both relays send a signal to the circuit breakers to disconnect the line with no delay. The implementation of the distance protection with the PUTT scheme on a line is illustrated in Fig. 6.5.

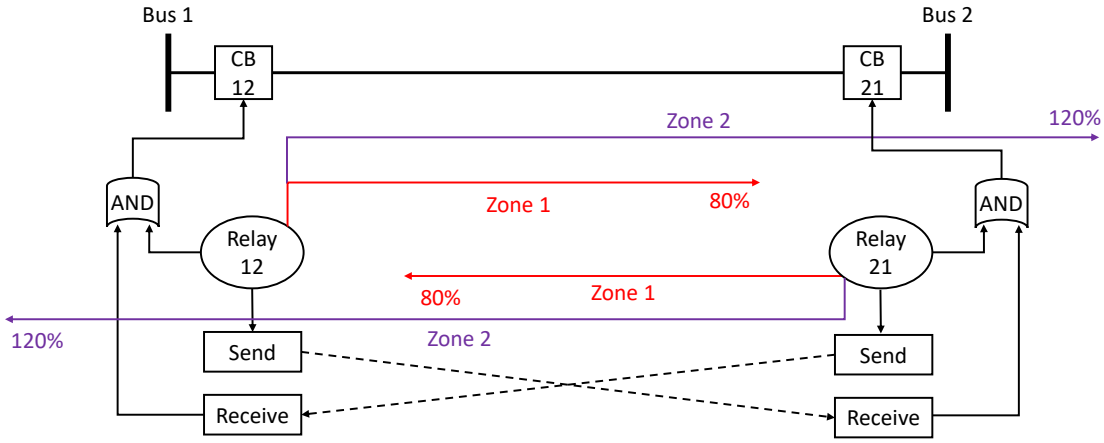

Figure 6.5: Distance protection on line with PUTT scheme.

6.4.2 Case studies

In this study 1470 LHS points have been selected from the 4-dimensional search space that is defined by the system loading and the output of the three windfarms. The loading of the system ranges from 80% to 120% of the total network demand, as it is calculated in the base case. The output of each of the three wind farms ranges from 0 to 100% of the nominal capacity. This sampling approach ensures that representative samples from equally divided areas of the search space are considered.

The initiating events in this case are three phase faults in the middle of each line (34 total lines). That gives 34 different cases for each LHS point. In total, 49980 cases have been simulated in this study. In 306 cases (0.6%) the OPF problem has not converged. with cascading events appearing in 19650 cases (39.3% of simulated cases). It should be noted that all the lines of the network are assumed to be single circuit lines, the disconnection of some of which following the fault causes the system to become islanded and consequently causing appearance of cascading failures. In reality, double circuit lines might exist in such cases so the applied initial contingency can be assumed to be N-2. More elaborate techniques to sample for N-k contingencies can also be applied when creating the training database [12], [9]. In addition, a similar efficient sampling [62] or importance sampling [107] method can be deployed to define the simulation cases in a larger network.

The average time that elapses between cascading events is 0.89s and the standard

deviation of this time is 5.24s. The moving time window length has been set to 10 time steps, which equals to 0.1s. A fast inference time is critical for system operators as that could provide more reaction time for preventive measures. For this application, the moving windows start after the clearing of the initial applied fault by the distance protection relays, up to 10s, which is the time duration when most time windows with cascading events (93.2%) appear in the dataset.

For the models application, the pre-processed time windows as described in Section 6.2.4 are used with a total of 180 time domain features. To create a dataset with equal percentage of safe cases and cases with cascading events, a balanced dataset has been created, consisting of 19650 safe cases and 19650 cases with cascading events, which includes all the cases with cascading events and randomly selected safe cases to match this number. The dataset is split in 31440 (80%) simulations for training, 3930 (10%) simulations for validation and 3930 (10%) simulations for testing of the model. The same percentage of safe and non-safe cases exists in the training, validation and test datasets.

6.4.3 Model Parameters

In this study, the models that are used are a Stacked LSTM, with two layers of LSTM cells as this configuration has shown to have the better performance compared to a single layer LSTM, a Bidirectional LSTM, a GRU and a TCN. For the RNNs, the batch size is set to 128 and the number of hidden neurons is set to 150. The batch size and the size of the hidden units is chosen based on model performance after performing a grid search for the following values for the batch size: {64, 128}, and for the hidden neurons: {100, 150, 200}. According to the best performing parameters for the TCN, it is comprised of 5 layers with dilation factors of: [1, 2, 4, 8, 16]. In addition, the number of kernels has been set to 128, the kernel size to 3, and a dropout rate of 20% to reduce over-fitting. To find the best performing parameters a grid search has been performed for number of layers: {4, 5, 6}, number of kernels: {128, 256}, kernel size: {2, 3} and dropout: {0, 20%}. Also, layer normalization is enabled.

The deep learning models are trained for 30 epochs with a learning rate of 0.001 and

the Adam optimizer. To avoid over-fitting, early stopping based on validation loss is enabled. The model with the best performing accuracy on the validation dataset across the epochs is saved. To ensure reproducibility, we set the model seed to 17 during all the models training.

6.5 Results

6.5.1 Performance for online identification of the reason of upcoming cascading events

The performance of the deep learning models is evaluated on the same test dataset, and the results are presented in Table 6.1. The total accuracy of each model, and the individual metrics are expressed in (%). The number of samples of each class are also included. It should be noted that 'O-o-S', 'OF', 'OV', 'UF' and 'UV' correspond to Out-of-Step, Over-Frequency, Over-Voltage, Under-Frequency and Under-Voltage respectively.

All the models showcase high total accuracy (>96.5%). The TCN model performs slightly better, with a total accuracy of 97.4%. The two LSTM based models, the Stacked LSTM and the Bidirectional LSTM, perform with a similar accuracy, 97% and 96.9% respectively. The GRU has the lowest total accuracy, at 96.6%. This can be attributed to the fact that this is a simpler RNN model, compared to the LSTM.

Looking at the individual metrics for each class can provide more information about how accurately the models identify each cascading event reason. When no cascading event follows, which is the majority class, all the models showcase high performing metrics. All models perform with a high recall of 99% for the 'No event' class. This is very important for a system operator, as all the models can identify whether an upcoming cascading event will occur with very high accuracy. The class of Out-of-Step events is the minority class, with only a few samples (71) in the test dataset. Due to this, the performance of the model on this class is lower. The GRU and TCN models perform with higher F1 scores, with the TCN having a higher Precision (67%).

When predicting cascading events due to frequency violations, the TCN model per-

forms best. More specifically, it performs with F1 scores of 96% and 95% for cascading events due to Over-frequency and Under-frequency respectively. The Stacked LSTM model showcases a high performance as well, with 96% F1 score for cases of Over-frequency and 94% for cases of Under-frequency. The performance metrics for the cases with voltage instability are lower. Again, the TCN model exhibits the highest performance, with a F1 score of 92% for cases of Over-voltage and 83% for cases of Under-voltage. In these cases of Under-voltage the Stacked LSTM model performs with the lowest F1 score of 75% and a low recall of 67%.

Regarding the prediction of distance relays operation, the TCN and the Bidirectional LSTM model perform with the highest F1 score of 91%. The TCN has a lower Precision, but higher Recall score. It can be concluded that the TCN model showcases generally better overall performance and slightly outperforms the Stacked LSTM, Bidirectional LSTM and GRU model on most of the individual class metrics.

The learning curves of the models during the training and validation are shown in Fig. 6.6. The categorical focal loss function is calculated during training and validation at each epoch. All the models converge well during training, and after 30 epochs signs of over-fitting appear, as the training loss decreases while the validation loss increases. The model that converges faster and has the lowest loss during training and validation is the TCN, which is also the model with the highest accuracy. The losses of the GRU model are the highest, which contributes to the fact that it showcases the lowest accuracy of these models on the test dataset.

6.5.2 Confusion matrix

In order to have a better understanding of how the model classifies the individual classes, the confusion matrix of the TCN model, the model with the highest accuracy, is presented in Table 6.2. Each row of the confusion matrix represents the actual labels of each class, while each column represents the predicted labels of each class.

As it can be observed, 99.32% (270315 samples) of the cases with no cascading events are predicted correctly. Of all the cases of trippings due to Out-of-Step, 39.44% (28 samples) are misclassified as cases of Under-voltage and 29.58% (21 samples) are

Table 6.1: Model evaluation metrics.

StackedLSTM				
Accuracy 97.0%				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	272174	98	99	99
Out-of-Step	71	52	17	26
OF	1781	96	95	96
OV	9726	90	88	89
UF	45789	96	93	94
UV	3648	86	67	75
Distance	20475	92	87	89
BiLSTM				
Accuracy 96.9%				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	272174	98	99	98
Out-of-Step	71	37	27	31
OF	1781	96	95	95
OV	9726	92	89	91
UF	45789	95	90	93
UV	3648	85	79	82
Distance	20475	94	88	91
GRU				
Accuracy 96.6%				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	272174	97	99	98
Out-of-Step	71	50	37	42
OF	1781	93	95	94
OV	9726	91	85	88
UF	45789	96	89	93
UV	3648	84	76	80
Distance	20475	91	88	89
TCN				
Accuracy 97.4%				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	272174	98	99	99
Out-of-Step	71	67	28	40
OF	1781	97	96	96
OV	9726	93	90	92
UF	45789	97	92	95
UV	3648	82	83	83
Distance	20475	93	89	91

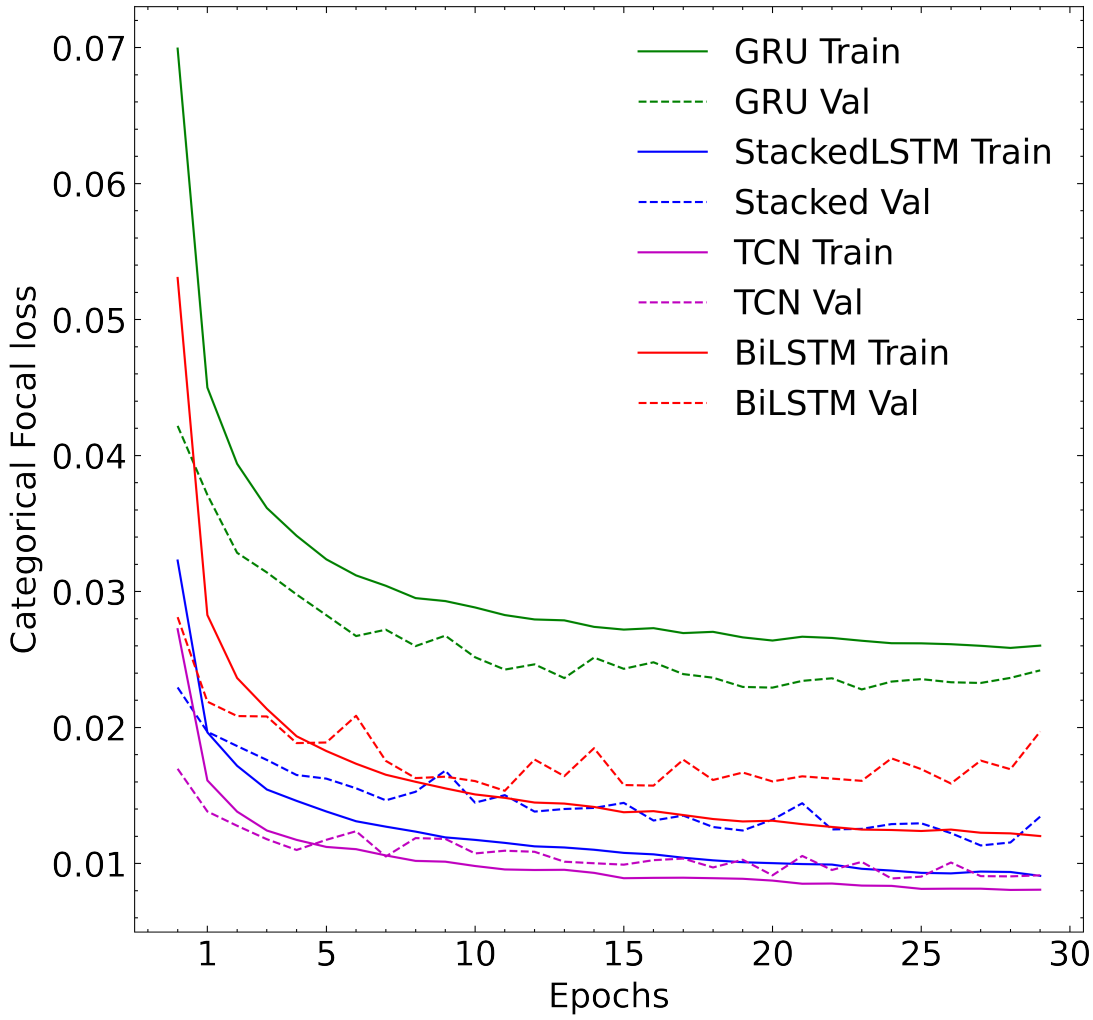

Figure 6.6: Learning curves of the models.

misclassified as cases of Distance protection activation. While the out of step class is generally the most under-performing class, it is worth noting that all instances are misclassified as a different relay protection operation but not as no event. After further investigation, in these cases the system is highly disturbed with several cascading events appearing and the events of Out-of-Step are preceeded by events of Distance or Under-voltage or followed by events of Under-voltage.

For the Over-voltage class, 282 (2.9%) of the cases are incorrectly predicted as safe cases. The disconnection of windfarm NSG-2 with no further events is common in this particular system and case studies. The high number of events due to Under-

frequency is caused by the UFLS scheme, which is activated in steps and disconnects a percentage of loading in order to restore the system frequency when cases of frequency instability are detected. For the Under-frequency class, 42280 (92.3%) of the cases are identified correctly. Finally, for the Under-voltage class the model predicts accurately 3038 (83.39%) of the cases, and for the Distance relays operation class predicts correctly 18284 (89.3%) of the cases. These results indicate that identifying the individual reason of cascading events is more challenging for the model than identifying the class of no cascading events.

Table 6.2: Confusion Matrix.

Actual label	Predicted label						
	No event	0-o-S	OF	OV	UF	UV	Dist.
No event	270315	0	1	347	838	46	627
O-o-S	0	20	1	0	1	28	21
OF	1	3	1710	1	25	39	2
OV	282	1	3	8778	229	8	425
UF	2826	4	45	71	42280	403	160
UV	255	1	9	8	182	3038	155
Dist.	1743	1	2	222	93	130	18284

6.5.3 Predicting cascading event sequences

An example of the proposed method application that shows a cascading event sequence, the moving time windows, and the output of the TCN model is presented in Fig. 6.7. The initial applied fault is a three-phase short-circuit on Line 16-19 at t=1s, and it gets cleared by the disconnection of the faulted line at t=1.07s, triggered by the distance protection relays. This line disconnection causes an area of the system to become islanded, and triggers the first cascading event which is the disconnection of the synchronous generator G4 at t=1.84s due to Over-speed ('G4-OF'). The disconnection of the synchronous generator causes the frequency of the system island to drop, and at t=5.05s the first stage of UFLS at Load 20 is activated, disconnecting 10% of Load 20 ('L20A-UF'), followed by the activation of the second stage of UFLS that disconnects an additional 10% of Load 20, at t=5.21s ('L20B-UF'). At t=1.07s the TCN model starts receiving time measurements in moving time windows of 0.1s, and gives predictions

about the reason of the cascading event that follows. The first prediction of the model after applying reverse encoding is 'OF', which indicates correctly that a cascading event due to Over-frequency follows. The model outputs for the following time windows have the same value ('OF'). At the time window that includes the time of the first event, the model predicts correctly the reason of the next cascading event, which is Under-frequency ('UF'). Similarly, the model predicts the reason of the next cascading event, which is again 'UF'. This example showcases the accurate prediction of the reason of cascading events involved in a particular sequence by the TCN model. For all the cascading event sequences, the following analysis is performed.

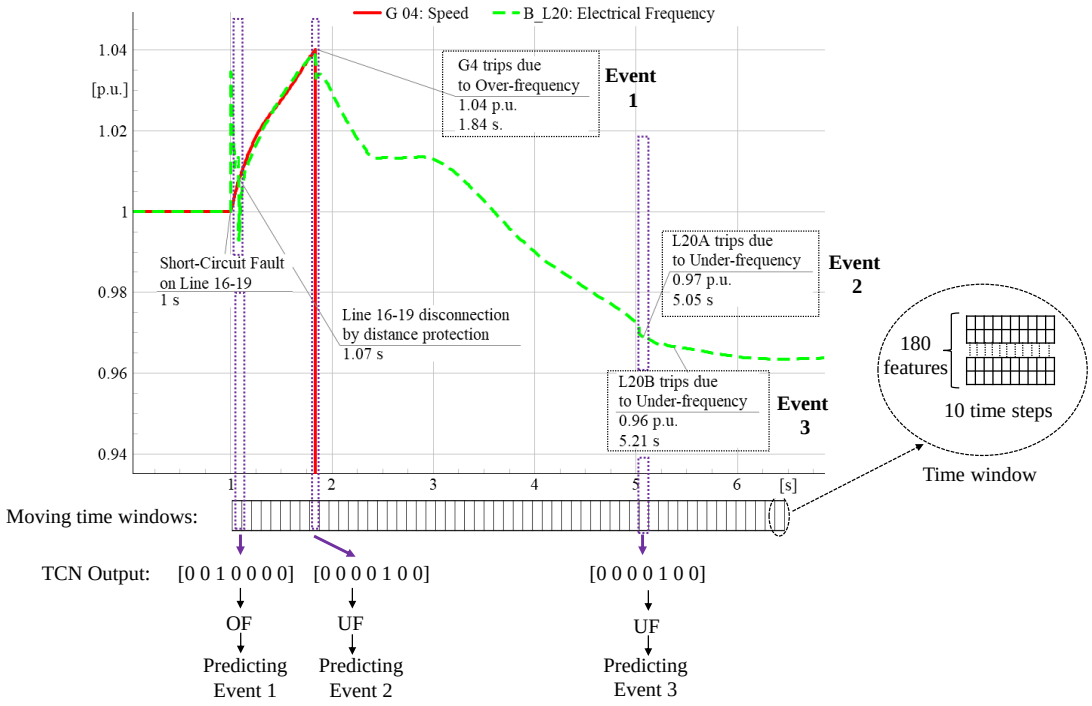

Figure 6.7: Predicting the reason of cascading events in a sequence.

The results discussed in Section 6.5.1 concern the whole test dataset and individual cascading events, i.e. the classification of each relay operation is investigated without taking into account what happens in the particular sequence of cascading events following a disturbance. In practical applications, it is also important to assess the performance of the models when predicting the reason of cascading events as they appear in a sequence. In this case only the cases that contain cascading events are considered,

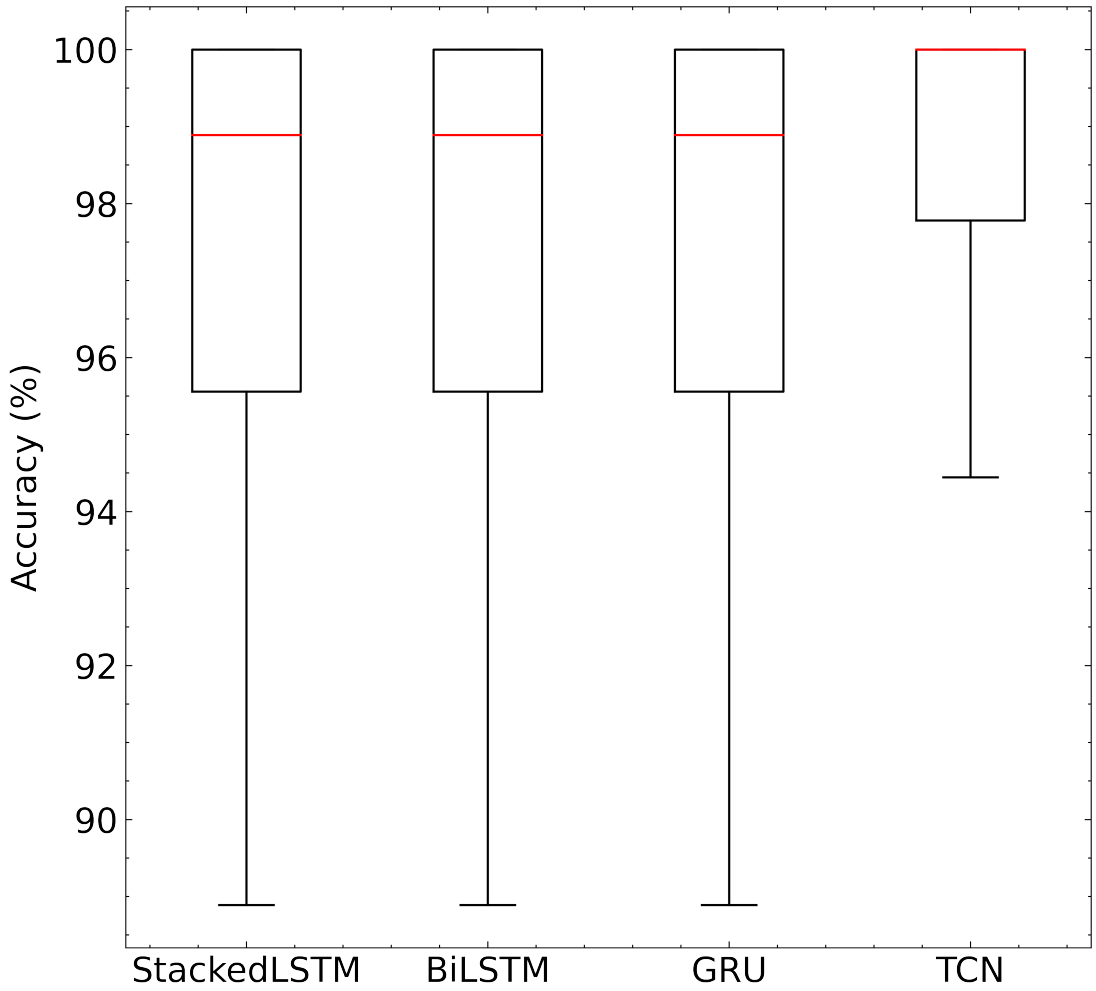

Figure 6.8: Model accuracy per cascading event sequence.

as this is what challenges the models most.

For each case with cascading events in the test dataset, the moving time windows are sequentially used as inputs to the models, and the overall accuracy for each cascading events sequence is calculated. The accuracy Acc_s for each cascading events sequence s is given by:

$$Acc_s(\%) = \frac{Ncp_{casc,s}}{N_{casc,s}} \tag{6.15}$$

where $Ncp_{casc,s}$ is the number of correct predictions for cascading sequence s, and $N_{casc,s}$ is the total number of cascading events in this sequence.

In Fig. 6.8 the results of the models performance are presented with boxplots. The performance of the BiLSTM, Stacked LSTM and GRU models is very similar, with the median value close to 99% and the first quartile value close to 95.8%. The TCN model has a better accuracy, with the median value close to 100%, the first quartile value close to 98% and smaller standard deviation. This can be explained by the fact that TCN showcases a better overall accuracy on the test dataset and higher performing metrics at individual classes.

6.5.4 Computational time and Implementation considerations

In this Section, aspects related to the computational time of the method are analysed. The computational time includes the time that is required for the dataset generation and the training of the deep learning models, both of which take place during the offline stage when more time is available, and the inference time which is critical, as it takes place at the online stage and it needs to be close to real-time. For the dataset generation, the adaptive time step option is used, in order to effectively reduce the computational time. The approximate averaged running time of one simulation without cascading events is 29s, and the running time of a simulation with cascading events is 94s. The interface between Python and DIgSILENT Powerfactory has been used to set up the dynamic simulations running 4 simulated cases in parallel in order to speed up the process of generating the described dataset. According to the available computational resources and the size of the test network, the number of parallel simulations can be adjusted proportionally. The deep learning models are implemented using TensorFlow [115], and are trained on a Nvidia Quadro RTX 6000 GPU. The training and inference times of the models are presented in Table 6.2. Due to the ability to perform computations in parallel, the training of the TCN model is faster than the training of the RNN based models.

Compared to CPUs, GPUs have a higher number of computational cores. For this reason, GPUs are better suited for parallel computations. In addition, GPUs are structurally optimized to have a higher memory bandwidth, meaning that data transfer between the processor and memory are performed more efficiently. On the other hand,

CPUs are optimized for latency, meaning that they can process small memory portions quickly, but slow down when they process larger amounts. This characteristic of GPUs is particularly important for deep learning models, which require a large amount of calculations to be performed in parallel.

During the online stage, the average time that a single prediction takes is critical, as this would provide system operators with valuable time for corrective measures. After performing 1000 predictions on the GPU the average time that is needed for a single prediction including pre-processing the data and the inverse transformation of the one-hot encoded label using the TCN model is 75ms, and it is faster compared to the other models.

To increase computational performance and lower the inference time, the pre-trained models are converted to the TensorFlow Lite format, which is smaller and more efficient. This model format is represented using the FlatBuffers protocol, which is an efficient serialization library designed for applications where performance is critical and allows data to be directly accessed without an extra parsing/unpacking step [116]. It should be noted that this conversion does not affect the prediction performance. In a similar manner, 1000 predictions are performed, and the average time that is needed for a single prediction using the TCN model is 1.09ms. As it is evident, deploying the model using the TensorFlow Lite format helps to achieve significantly faster inference times, compared to the standard TensorFlow format. It should be also noted that the prediction time is significantly faster than the duration of one time window (0.1s), considering also the maximum inference time (4.87ms), and that the system operator can have a fast prediction on a continuous basis.

6.5.5 Testing the pre-trained model on a different network topology

Transfer Learning is a ML method where a pre-trained model on a task is used for a different, but related task [117]. In other words, this technique focuses on transferring knowledge gained from solving a problem to applying it to another similar problem. This can be particularly useful when the amount of data available for training a ML model are limited. Such cases might appear in power system applications, as the

Table 6.3: Deep learning models computational time.

	StackedLSTM	**BiLSTM**	**GRU**	**TCN**
	Training time (hrs)			
	12.2	12.1	11.4	9.6
	TensorFlow format			
	Inference time (ms) (over 1000 predictions)			
Avg.	78	78	77	75
Min./Max.	70/84	69/84	66/81	63/81
	TensorFlow Lite format			
	Inference time (ms) (over 1000 predictions)			
Avg.	1.15	1.15	1.12	1.09
Min./Max.	1.05/8.89	1.05/8.89	1.05/7.11	1.04/4.87

connection of new RES units to the grid can change the network topology.

In this case, a new system topology is considered, by adding a new wind farm (NSG4) on Bus 26 of the modified IEEE-39 bus model (Fig. 6.9). The aim is to assess the performance of the pre-trained TCN model on this version of the network. Using again the LHS technique for the definition of the initial operating conditions, in this case the power output of the 4 wind farms and system loading, 1000 dynamic RMS simulations are performed. The performance of the TCN model for the prediction of the reason of the next cascading event in this test dataset is presented in Table 6.4.

For this network, the TCN model performs with a 96.0% overall accuracy. Compared to the performance of the TCN model for the original network (Table 6.1), the accuracy of the model decreases by 1.44%. By investigating the performance on individual classes, the F1-Score of classes 'No event', 'OF', 'UF' and 'UV' remains the same or decreases slightly (up to 1%), where the performance for classes 'Out-of-Step', 'OV' and 'Distance' is significantly lower. When more data from this network topology become available, the model weights could be updated by fine-tuning the pre-trained model. This can be achieved by further training the model on the new dataset, with the aim of adjusting its parameters to make it better suited for the new task.

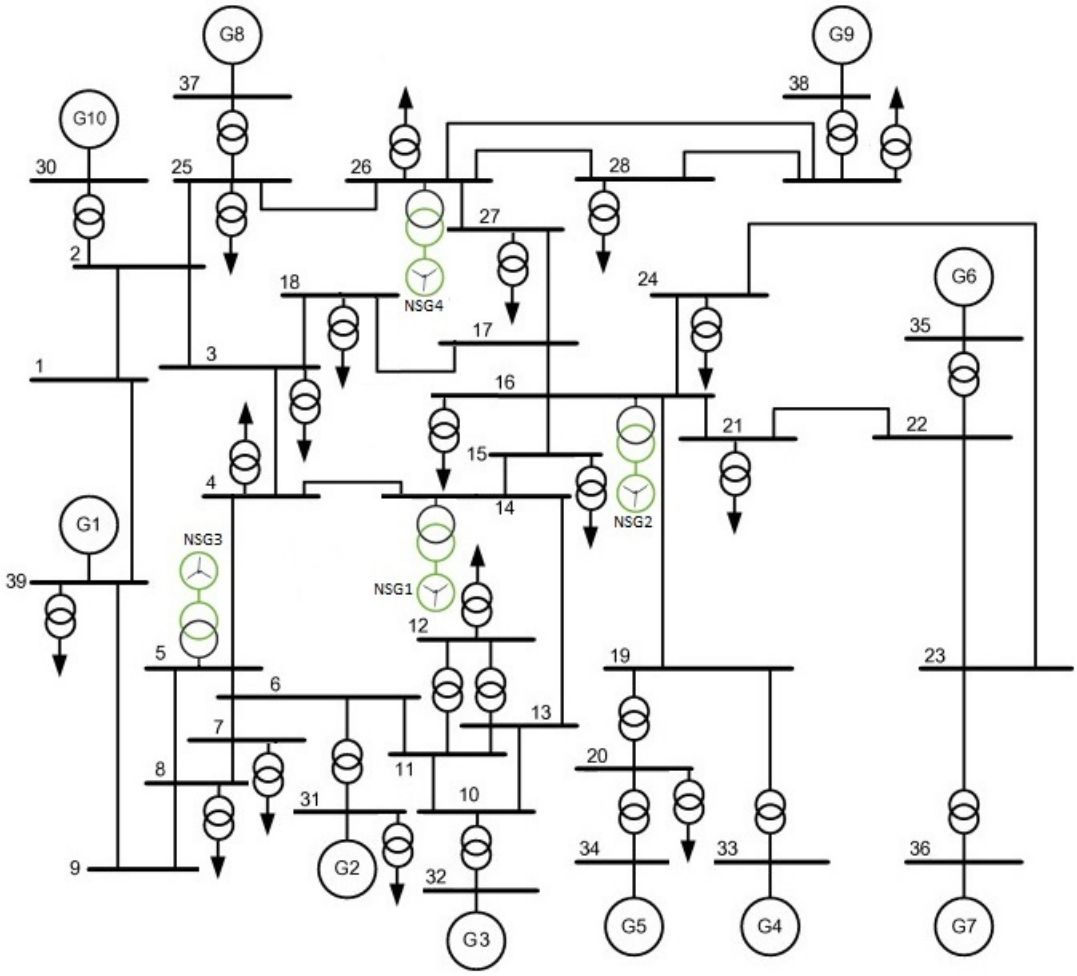

Figure 6.9: Modified version of the IEEE-39 bus model with 4 wind farms.

Table 6.4: Model performance on a different network topology.

TCN				
Accuracy 96.0%				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	74108	98	98	98
Out-of-Step	79	87	16	28
OF	336	96	96	96
OV	1874	70	80	75
UF	7624	92	97	94
UV	1046	85	80	83
Distance	5831	86	75	80

6.5.6 Application of the proposed method on a larger network model

In order to investigate the scalability of the proposed method, it is also applied on a modified version of the IEEE 118-bus system. In this case, the network model has been augmented with three wind farms equipped with under-/over-voltage protection relays and under-/over-frequency protection relays. In addition, the model has been augmented with under-/over-speed protection, under-voltage protection and out-of-step protection relays for the SGs. An identical sampling methodology as described in Section 6.2.2 is employed and 17511 dynamic RMS simulations using this model are performed.

The best performing model, TCN, is trained and evaluated on the resulting dataset. In this case, there are 4 output classes for the multi-class classification: No event, Over-Frequency, Over-Voltage, and Under-Voltage. No Out-of-Step or Under-Frequency events have been observed in this dataset. The model evaluation metrics on the test dataset are presented in Table 6.5. Overall, the model performs with 99.4% accuracy. Looking at the individual classes, the model achieves 100% Precision and 100% Recall scores for the No event class. The model performance is also high for class Over-Frequency, performing with 100% Precision and 96% Recall. However, the model performance for classes Over-Voltage and Under-Voltage is reduced, achieving a 71% and 82% F1-score respectively. The overall performance of the TCN is better in this case, highlighting the fact that the complexity of observed system dynamics is not necessarily scaling up with the size of the network but can also depend on other aspects. Consequently the learning problem posed might not necessarily be more difficult.

The online computational time that a single inference takes is calculated in a similar manner as in 6.5.4. After performing 1000 predictions, the average time that a single prediction of the TCN model takes is 1.51ms. Overall, this case study showcases the ability of the proposed method to be adjusted and applied on larger system models, while maintaining high predictive and computational performance.

Table 6.5: TCN Model evaluation on the IEEE 118-bus system.

TCN				
Acc (%)				
99.4				
Reason/Class	Samples	Precision (%)	Recall (%)	F1 (%)
No event	31246	100	100	100
OF	56	100	96	98
OV	6	62	83	71
UV	564	83	80	82

6.6 Chapter Conclusions

This chapter presents a method for the online identification of cascading event sequences in power systems. The proposed framework can identify in close to real-time the reason of cascading events about to happen by utilising measurement data and supervised deep learning techniques. The prediction is formulated as a multi-class classification problem and captures detailed dynamic responses of the system. The following classes, corresponding to different reasons of upcoming cascading events, are considered: No event, Out-of-Step, Over-Frequency, Over-Voltage, Under-Frequency and Under-Voltage.

The deep learning models are trained using a dataset that consists of dynamic RMS simulations performed on a model with renewable generation and protection devices, the action of which defines the appearance of cascading events after an initial contingency. The simulated data are formed into moving time windows and are pre-processed to represent typical PMU data to be used for the training of the proposed deep learning model. During the online phase the pre-trained model is used to predict the reason of cascading events about to happen. Different deep learning techniques that can handle time series data, appropriate for the formulated problem, have been examined and compared. The use of TCNs is proposed because of their ability to perform calculations in parallel using convolutions, as well as due to slightly improved performance and faster training and inference times. Focal loss function is also proposed to be used for dealing with the imbalanced multiclass dataset. The evaluation metrics of the model performance are analysed, taking also into consideration the performance per cascading event sequence. The method is applied on a modified version of the IEEE-39

bus system, augmented with wind generation and protection devices.

The results highlight that the proposed TCN model performs with a 97.4% accuracy on the multi-class classification of the reason of cascading events with a moving time window of 0.1s. The performance of the model is compared to other RNN configurations, including a StackedLSTM, a BiLSTM and a GRU model. Compared to the RNNs, the TCN model has the ability to predict the reason of cascading events sequences with a slightly higher performance and with a faster inference time, which is critical in online applications. However, after investigating the performance metrics for individual classes, it is revealed that the model showcases poorer performance for the minority class (Out-of-step). Results on the larger model showcase that the TCN model maintains high performance, highlighting therefore the scalability of the proposed method.

Chapter 7

Conclusions and Future Work

7.1 Summary and Key Outcomes

Throughout this thesis, particular focus has been given on the fact that the increasing penetration of RES and other converter interfaced devices in power systems along with the uncertainty that comes with them, makes the operation of modern power systems challenging, especially with respect to the dynamic behaviour. More specifically, because of the intermittent nature of RES, their power output can vary significantly, affecting the system SG generation. In certain occasions, a contingency can endanger the secure operation of the system, leading even to a system collapse with severe impacts to society. As power systems are highly complex and non-linear dynamical systems, it is challenging and computationally intensive to assess their response to such cascading events, which are low probability but high impact events. For this reason, recent research focuses on re-establishing stability definitions [1].

Nowadays, PMUs that exist in modern power systems can provide real-time measurement data about the grid operational status. For this reason, in recent literature, data-driven approaches that utilize these measurements are investigated in order to identify the system online security. In this thesis, ML techniques are employed to utilize simulated data, that have been pre-processed to represent typical data that can be acquired from PMUs, for the online identification of cascading events.

Chapter 2 presents a detailed literature review on relevant recent studies that focus

on the modelling of cascading events and on the online identification of security in power systems. The analysis of the studies that focus on the modelling of cascading events has shown that most of these use static methods to investigate the appearance of cascading events and how these propagate. However, studies [11], [10] have shown that dynamic simulations can provide more information about cascading events than static methods, at the expense of a larger computational effort. Furthermore, it is important to consider the RES penetration, as it can significantly impact the dynamic behaviour of the system following a contingency. The analysis of the studies focusing on online security has revealed that ML based approaches can accurately predict the network online state. Nevertheless, most of the methods predict only early cases of either transient, voltage, frequency instability or dynamic security, and do not consider the action of protection devices. In real power systems though, protection devices might activate before actual instability limits are reached, and the trip of a component (e.g. disconnection of a line), might cause subsequent cascading events. Thus, it is of great importance to accurately represent the action of protection devices to capture the cascading events that might appear in a system [2], [4].

Based on the previous remarks, in Chapter 3 a framework for the investigation of cascading events in power systems using dynamic simulations and considering RES penetration and the action of protection devices is presented. Each cascading event is characterized by the component that trips, the reason for tripping, and the time of the event, which provides the sequence in which cascading events appear. Using this information, the most common cascading event patterns and how many times each pattern appears are identified. For the initial method application, a modified version of the Anderson-Fouad 9 bus model, augmented with the implementation of protection devices and wind farms, is utilized. Furthermore, the SG disconnection because of RES penetration and the dispatch of SG units as defined by their operational cost through the AC OPF solution for each distinct operational scenario are considered. The results highlight that the appearance of cascading events can vary across different wind generation outputs and system loading conditions. In addition, the most vulnerable area of the system (Area 2) is identified, by looking into the most common cascading

event how it is related to locational aspects of the network.

Furthermore, this Chapter investigates the impact of AGC, a frequency related mechanism, and LTC, a voltage related mechanism, on the appearance of cascading events. To quantify the mechanisms impact, three distinct scenarios are defined. The base scenario, which includes only the action of LTCs, Scenario I, where the LTCs are deactivated, and Scenario II, where the action of both LTCs and AGC is considered. The effect of LTCs and AGC is highlighted by comparing the number and reason of cascading events, the average load loss, the time between consecutive events and the most common cascading event patterns across the three scenarios. The impact of LTCs and AGC on cascading events is showcased again with dynamic RMS simulations but on a larger network, a modified version of the IEEE-39 bus model that also includes RES penetration and protection devices. The results of this study highlight that in the scenario where the LTCs are deactivated, there is a higher number of cascading events, which has led to a greater total amount of load loss. The operation of AGC has shown to enhance the frequency stability of the network, resulting in a decreased number of UFLS activation events.

The investigation of the appearance of cascading events in the previous chapters has highlighted that the impact of wind penetration for different operating conditions can significantly affect the dynamic behaviour of the system in unpredictable ways, which can not be defined in a straightforward manner. In Chapter 4 the use of ML models and application of the SHAP XAI technique are explored, to develop a framework that can predict the appearance of cascading events in power systems using the measurements of the power systems during steady-state, providing also explanations about the decision-making process of the models.

The dataset used for this study consists of the dynamic RMS simulations that were carried out in the IEEE-39 bus model, as defined in the base case of the previous Chapter. First, various ML algorithms are implemented to predict the probability of cascading events, without considering the location of the initial contingency. After evaluation, the best performing model for this task is the Extra Trees model, achieving a 1.65% MAE score. However, it was revealed that the model can under-estimate

the probability of cascading events, with a value of up to 14.1%. This has led to the addition of information about the line on which the applied contingency appears to the input features of the model, framing now the prediction of appearance of cascading events in a specific simulation as a binary classification problem. After performing a similar investigation, the model that showcases the best performance is an XGBoost model, with 98.61% accuracy. The most important features for this model predictions are identified through a permutation feature importance analysis. On top of that, the SHAP XAI technique is applied, which can provide extra insights about how each feature affects the model prediction through feature attribution. Individual instances of both correct and incorrect predictions are provided and analyzed, as well as a global explanation plot. Based on the results, the most important power systems feature is the network line on which the initial fault is applied, followed by the feature that refers to the measurement value that causes the most common cascading event for this particular test system and study.

In Chapter 5 the time-series measurements that can be acquired from PMUs after a contingency are utilized to predict the appearance of cascading events. The time-series features in this study are used as inputs to a LSTM model, which is a type of RNN network, that can learn form sequential data. After training, the model is evaluated on the test dataset, including a detailed analysis into how the length of the time window and the wind generation output and system loading affect the model performance. Furthermore, a feature permutation importance analysis is carried out to identify the most important features and practical aspects related to PMUs, like noisy signals and limited availability, are taken into consideration.

The LSTM model achieves a 95.6% accuracy on the prediction of cascading events appearance, outperforming baseline approaches that comprise of a MLP and RNN model. The dataset consists of the results of the dynamic RMS simulations as performed on the modified IEEE-39 bus model, with protection devices and RES units. The state of the initial conditions, in this case wind generation output and system loading, appears to have an impact on model performance. More specifically, for this particular test system the performance of the model is higher when the system loading is at 100% of the

base value and when the wind generation output has higher values. The permutation feature importance analysis reveals that the time-domain features that have a higher impact on model performance are the active and reactive power measurements on lines, and also the measurement related to the most common appearing cascading event, the voltage magnitude on Bus NSG2. The added noise on PMU signals has shown to have no effect on model performance, while it has been identified that a smaller set of the most important features can be used as input with a slight decrease in accuracy.

Chapter 6 expands on the work of the previous chapter, and presents a method for identifying the reason of cascading event sequences in power systems in close-to-real time, by utilizing PMU data. In this case, the aim is to not only predict if a cascading event will appear or not, but also to predict the reason that causes the cascading event. For this reason, the problem is formulated as a multi-class classification problem and 7 classes are defined that correspond to different reasons of relay operations: No event, Out-of-Step, Over-Frequency, Over-Voltage, Under-Frequency and Under-Voltage. Furthermore, in order to identify the reason of cascading events as they appear in sequences, a dataset with moving time windows of 0.1s is formed and utilized for the training and online inference of the models.

In this Chapter particular focus is given on the use of TCNs, as calculations with convolutions can been performed in parallel, thus improving performance and achieving faster training and inference times. In addition, as the resulting multi-class dataset is highly imbalanced, the use of focal loss is proposed to penalize instances that are hard to classify. The multi-class classification of cascading events is showcased on the modified version of the IEEE-39 bus system, augmented with the action of distance protection relays. The use of an efficient sampling technique, LHS, is applied to define the study cases in order to achieve a more effective coverage of the search space, as defined by the wind farms output generation and system loading.

The TCN model predicts with a 97.4% accuracy the reason of cascading events, outperforming slightly a StackedLSTM, a BiLSTM and a GRU model. Compared to the RNNs, the TCN model has the ability to predict the reason of cascading events sequences with a slightly higher performance and with a faster inference time, which

is critical in online applications. Furthermore, after investigation the TCN model has also shown better performance than the RNN models when predicting the reason of cascading events as they appear in a sequence.

The fast evolving decarbonization effort and the adaptation of electricity markets functioning are expected to bring significant changes to the operation of power systems. For this reason, there is an immediate need for sophisticated and robust methods that can identify cascading events in an online manner, and can provide to system operators information about the mitigation and prevention of cascading events propagation. The ML-based methodological frameworks presented in this thesis can accurately predict the appearance of cascading events, with the data generation process consisting of detailed dynamic RMS network models with RES penetration that can capture the action of protection devices. The fast inference time of the developed models and the consideration of practical aspects related to PMU measurements verify the applicability and robustness of the proposed methods for real-life online deployment.

7.2 Future Work

The research work carried out as part of the PhD project that is presented in this thesis makes several contributions to the modelling and online identification of cascading events in modern power systems, investigating the application of contemporary ML algorithms. However, there have been identified some additional areas of research that the presented work could be extended.

- Performing simulations on a larger network to investigate the cascading event sequences that appear and consideration of $N-k$ contingencies. Based on this set of simulations, the application of the ML algorithms implemented in this thesis should follow in order to verify the scalability of the proposed methods. Potential challenges include the computational time of the simulations that would be greater in a larger network and for $N-k$ contingencies, and also the increased number of features for the ML application.

- A larger test network would also increase the search space, so a more sophisticated

method could be utilized for the definition of the initial operating conditions that lead to the more frequent appearance of cascading events. This could lead to the creation of a more balanced dataset for supervised ML models application. Potential techniques that could be applied are Reinforcement Learning and Monte Carlo Tree Search methods.

- The inclusion of information about past discrete events as input data to the ML model for the prediction of the next event cascading event that follows. This additional information could potentially increase the prediction performance and reveal connections between the cascading events of a sequence.

- The application of alternative ML algorithms could be investigated in order to try to increase the performance in the online identification of cascading events. Physics-informed Neural Networks that can include information about the equations of a power system could potentially showcase interesting results.

- The test network could be augmented with the modelling of additional contemporary technologies found in modern power systems. Such technologies could include Photo-Voltaic parks, Battery Energy Storage Systems and Electric Vehicles. The implementation of these elements would introduce additional uncertainty to the test network and it would be interesting to investigate how this would affect the appearance of cascading events. This process would increase the computational time as the network model would become more complicated.

- Following the prediction of the cascading event that follows, another ML based model could be trained to take corrective control actions in order to prevent or mitigate the effects of the next cascading event. This role could be assigned to a Reinforcement Learning agent that would interact with the environment, in this case the power system simulator, and would be trained to take actions that return the optimal reward.

Bibliography

[1] N. Hatziargyriou, J. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. Sanchez-Gasca, A. Stankovic, T. Van Cutsem, V. Vittal, and C. Vournas, "Definition and classification of power system stability – revisited & extended," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3271–3281, 2021.

[2] J. Bialek, E. Ciapessoni, D. Cirio, E. Cotilla-Sanchez, C. Dent, I. Dobson, P. Henneaux, P. Hines, J. Jardim, S. Miller, M. Panteli, M. Papic, A. Pitto, J. Quiros-Tortos, and D. Wu, "Benchmarking and Validation of Cascading Failure Analysis Tools," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4887–4900, 2016.

[3] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, 2004.

[4] M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller, and P. Zhang, "Risk assessment of cascading outages: Methodologies and challenges," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 631–641, 2012.

[5] M. Noebels, I. Dobson, and M. Panteli, "Observed acceleration of cascading outages," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3821–3824, 2021.

[6] "Ieee standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, 2011.

[7] S. Azizi, G. B. Gharehpetian, and A. S. Dobakhshari, "Optimal integration of phasor measurement units in power systems considering conventional measurements," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1113–1121, 2013.

[8] J. Hazra and A. Sinha, "Identification of catastrophic failures in power system using pattern recognition and fuzzy estimation," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 378—-387, 2009.

[9] M. Eppstein and P. Hines, "A "random chemistry" algorithm for identifying collections of multiple contingencies that initiate cascading failure," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1698—-1705, 2012.

[10] E. Ciapessoni, D. Cirio, and A. Pitto, "Cascadings in large power systems: Benchmarking static vs. time domain simulation," in *2014 IEEE PES GM*, pp. 1–5, 2014.

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

[12] T. Weckesser and T. Van Cutsem, "Identifying plausible harmful n-k contingencies: A practical approach based on dynamic simulations," in *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8, 2018.

[13] M. H. Vasconcelos, L. M. Carvalho, J. Meirinhos, N. Omont, P. Gambier-Morel, G. Jamgotchian, D. Cirio, E. Ciapessoni, A. Pitto, I. Konstantelos, G. Strbac, M. Ferraro, and C. Biasuzzi, "Online security assessment with load and renewable generation uncertainty: The itesla project approach," in *2016 International Conference on Probabilistic Methods Applied to Power Systems, PMAPS 2016 - Proceedings*, pp. 1–8, 2016.

[14] P. N. Papadopoulos, T. Guo, and J. V. Milanović, "Probabilistic framework for online identification of dynamic behavior of power systems with renewable generation," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 45–54, 2018.

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

[16] L. Zhu, D. J. Hill, and C. Lu, "Intelligent short-term voltage stability assessment via spatial attention rectified rnn learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7005–7016, 2021.

[17] G. A. Nakas, A. Dirik, P. N. Papadopoulos, A. R. R. Matavalam, O. Paul, and D. Tzelepis, "Online identification of cascading events in power systems with renewable generation using measurement data and machine learning," *IEEE Access*, vol. 11, pp. 72343–72356, 2023.

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

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

[20] A. S. C. Leavy, G. A. Nakas, and P. N. Papadopoulos, "A method for variance-based sensitivity analysis of cascading failures," *IEEE Transactions on Power Delivery*, vol. 38, no. 1, pp. 463–474, 2023.

[21] M. A. Ortega-Vazquez, "Assessment of n-k contingencies in a probabilistic security-constrained optimal power flow," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, 2016.

[22] P. Hines, I. Dobson, and P. Rezaei, "Cascading power outages propagate locally in an influence graph that is not the actual grid topology," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 958—-967, 2017.

[23] K. Zhou, I. Dobson, Z. Wang, A. Roitershtein, and A. P. Ghosh, "A markovian influence graph formed from utility line outage data to mitigate large cascades," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3224–3235, 2020.

[24] J. Qi, I. Dobson, and S. Mei, "Towards estimating the statistics of simulated cascades of outages with branching processes," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3410–3419, 2013.

[25] R. Fitzmaurice, E. Cotilla-Sanchez, and P. Hines, "Evaluating the impact of modeling assumptions for cascading failure simulation," in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8, 2012.

[26] Y. Dai, M. Noebels, M. Panteli, and R. Preece, "Evaluating the effect of dynamic and static modelling on cascading failure analysis in power systems," in *2021 IEEE Madrid PowerTech*, pp. 1–6, 2021.

[27] Y. Dai, M. Noebels, R. Preece, M. Panteli, and I. Dobson, "Risk assessment and mitigation of cascading failures using critical line sensitivities," *IEEE Transactions on Power Systems*, pp. 1–12, 2023.

[28] B. Schaefer, D. Witthaut, M. Timme, and V. Latora, "Dynamically induced cascading failures in power grids," *Nature Communications*, vol. 9, 2018.

[29] S. Datta and V. Vittal, "Operational risk metric for dynamic security assessment of renewable generation," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1389–1399, 2017.

[30] A. J. Flueck, I. Dobson, Z. Huang, N. E. Wu, R. Yao, and G. Zweigle, "Dynamics and protection in cascading outages," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, 2020.

[31] I. Konstantelos, G. Jamgotchian, S. H. Tindemans, P. Duchesne, S. Cole, C. Merckx, G. Strbac, and P. Panciatici, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1417–1426, 2017.

[32] Z. Shi, W. Yao, L. Zeng, J. Wen, J. Fang, X. Ai, and J. Wen, "Convolutional neural network-based power system transient stability assessment and instability mode prediction," *Applied Energy*, vol. 263, p. 114586, 2020.

[33] J.-M. H. Arteaga, F. Hancharou, F. Thams, and S. Chatzivasileiadis, "Deep learning for power system security assessment," in *2019 IEEE Milan PowerTech*, pp. 1–6, 2019.

[34] L. Zheng, W. Hu, Y. Zhou, Y. Min, X. Xu, C. Wang, and R. Yu, "Deep belief network based nonlinear representation learning for transient stability assessment," in *2017 IEEE Power & Energy Society General Meeting*, pp. 1–5, 2017.

[35] P. Sekhar and S. Mohanty, "An online power system static security assessment module using multi-layer perceptron and radial basis function network," *International Journal of Electrical Power & Energy Systems*, vol. 76, pp. 165–173, 2016.

[36] N. Amjady and S. F. Majedi, "Transient stability prediction by a hybrid intelligent system," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1275–1283, 2007.

[37] A. Iravani and F. de León, "Real-time transient stability assessment using dynamic equivalents and nonlinear observers," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2981–2992, 2020.

[38] T. L. Vu and K. Turitsyn, "Lyapunov Functions Family Approach to Transient Stability Assessment," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1269–1277, 2016.

[39] P. Ju, H. Li, C. Gan, Y. Liu, Y. Yu, and Y. Liu, "Analytical Assessment for Transient Stability under Stochastic Continuous Disturbances," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 2004–2014, 2018.

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

[41] J. Kruse, B. Schäfer, and D. Witthaut, "Revealing drivers and risks for power grid frequency stability with explainable ai," *Patterns*, vol. 2, no. 11, p. 100365, 2021.

[42] W. Yi and D. J. Hill, "Topological stability analysis of high renewable penetrated systems using graph metrics," in *2021 IEEE Madrid PowerTech*, pp. 1–6, 2021.

[43] T. Han, J. Chen, L. Wang, Y. Cai, and C. Wang, "Interpretation of stability assessment machine learning models based on shapley value," in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*, pp. 243–247, 2019.

[44] S. Wu, L. Zheng, W. Hu, R. Yu, and B. Liu, "Improved deep belief network and model interpretation method for power system transient stability assessment," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 1, pp. 27–37, 2020.

[45] C. Ren, Y. Xu, and R. Zhang, "An interpretable deep learning method for power system transient stability assessment via tree regularization," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3359–3369, 2022.

[46] B. Tan, J. Zhao, T. Su, Q. Huang, Y. Zhang, and H. Zhang, "Explainable bayesian neural network for probabilistic transient stability analysis considering wind energy," in *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, 2022.

[47] R. I. Hamilton, P. N. Papadopoulos, W. Bukhsh, and K. Bell, "Identification of important locational, physical and economic dimensions in power system transient stability margin estimation," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 1135–1146, 2022.

[48] R. I. Hamilton and P. N. Papadopoulos, "Using shap values and machine learning to understand trends in the transient stability limit," *IEEE Transactions on Power Systems*, pp. 1–12, 2023.

[49] T. Van Cutsem, M. Glavic, W. Rosehart, C. Canizares, M. Kanatas, L. Lima, F. Milano, L. Papangelis, R. A. Ramos, J. A. d. Santos, B. Tamimi, G. Taranto, and C. Vournas, "Test systems for voltage stability studies," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 4078–4087, 2020.

[50] N. Hasan, Ibrhaeem, and Singh, "An overview of agc strategies in power systems," pp. 72–79, 01 2009.

[51] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 10, pp. 1866–1876, 1968.

[52] R. Hamilton, P. Papadopoulos, and K. Bell, "An investigation into spatial and temporal aspects of transient stability in power systems with increasing renewable generation," *International Journal of Electrical Power & Energy Systems*, vol. 115, pp. 105486,, 2020.

[53] P. N. Papadopoulos, J. V. Milanović, P. Bhui, and N. Senroy, "Feasibility study of applicability of recurrence quantification analysis for clustering of power system dynamic responses," in *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6, 2016.

[54] P. Anderson, A. Fouad, and H. Happ, "Power system control and stability," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 2, 1979.

[55] "Digsilent-powerfactory user manual, digsilent gmbh," 2019.

[56] X. Xu, X. Su, D. Zhang, P. An, and J. Sun, "Parameter identification of six-order synchronous motor model based on grey box modeling," in *Green Energy and Networking* (X. Jiang and P. Li, eds.), (Cham), pp. 45–52, Springer International Publishing, 2020.

[57] P. Sørensen, B. Andresen, J. Fortmann, and P. Pourbeik, "Modular structure of wind turbine models in iec 61400-27-1," in *2013 IEEE Power & Energy Society General Meeting*, pp. 1–5, 2013.

[58] N. Grid, "National grid electricity system operator limited."

[59] D. T. Documentation, "Digsilent powerfactory application guide dpl tutorial digsilent technical documentation."

[60] R. Preece and J. Milanović, "Efficient estimation of the probability of small-disturbance instability of large uncertain power systems," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1063—1072, 2016-03.

[61] P. Papadopoulos and J. Milanovi, "Efficient identification of transient instability states of uncertain power systems," in *IREP 2017 Symposium on Bulk Power Systems Dynamics and Control*, pp. 1—-7, 2017.

[62] V. Krishnan, J. D. McCalley, S. Henry, and S. Issad, "Efficient database generation for decision tree based power system security assessment," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2319–2327, 2011.

[63] C. Machado Ferreira, F. P. Maciel Barbosa, and C. I. Faustino Agreira, "Transient stability preventive control of an electric power system using a hybrid method," in *2008 12th International Middle-East Power System Conference*, pp. 141–145, 2008.

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

[65] B. Isaıas Lima Lopes and A. Zambroni de Souza, "On multiple tap blocking to avoid voltage collapse," *Electric Power Systems Research*, vol. 67, no. 3, pp. 225–231, 2003.

[66] O. Kyrylenko, V. Pavlovsky, and A. Steliuk, "AGC software model validation for identification of renewables impact on frequency control in the IPS of Ukraine," *2014 IEEE International Conference on Intelligent Energy and Power Systems, IEPS 2014 - Conference Proceedings*, pp. 141–144, 2014.

[67] V. Pavlovsky and A. Steliuk, *Modeling of Automatic Generation Control in Power Systems*, pp. 157–173. Cham: Springer International Publishing, 2014.

[68] L. Veiber, K. Allix, Y. Arslan, T. F. Bissyandé, and J. Klein, "Challenges towards Production-Ready explainable machine learning," in *2020 USENIX Conference on Operational Machine Learning (OpML 20)*, USENIX Association, July 2020.

[69] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *CoRR*, vol. abs/1910.10045, 2019.

[70] R. Machlev, L. Heistrene, M. Perl, K. Levy, J. Belikov, S. Mannor, and Y. Levron, "Explainable artificial intelligence (xai) techniques for energy and power systems: Review, challenges and opportunities," *Energy and AI*, vol. 9, p. 100169, 2022.

[71] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[72] D. F. Andrews, "A robust method for multiple linear regression," *Technometrics*, vol. 16, no. 4, pp. 523–531, 1974.

[73] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[74] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[75] H. Zhang, "The optimality of naive bayes," in *The Florida AI Research Society*, 2004.

[76] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[77] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," *Biometrics*, vol. 40, p. 874, 1984.

[78] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, Sep 1995.

[79] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, p. 367–378, feb 2002.

[80] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016.

[81] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 2004.

[82] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[83] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, Apr 2006.

[84] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.

[85] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[86] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[87] S. Shankar, V. Rathore, K. Yadav, and A. Priyadarshi, "Observability of system using optimal pmus location," in *2021 7th International Conference on Electrical Energy Systems (ICEES)*, pp. 414–419, 2021.

[88] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[89] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *CoRR*, vol. abs/1705.07874, 2017.

[90] L. S. Shapley, *A Value for N-Person Games*. Santa Monica, CA: RAND Corporation, 1952.

[91] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

[92] S. M. Lundberg, G. G. Erion, and S. Lee, "Consistent individualized feature attribution for tree ensembles," *CoRR*, vol. abs/1802.03888, 2018.

[93] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 3145–3153, JMLR.org, 2017.

[94] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer New York, 2013.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

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

[97] P. N. Papadopoulos, T. Guo, X. Wang, and J. V. Milanović, "Impact of measurement signals on the accuracy of online identification of power system dynamic signature," in *2015 IEEE Eindhoven PowerTech*, pp. 1–6, 2015.

[98] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[99] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[100] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[101] A. Graves, "Long short-term memory," in *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, Springer, 2012.

[102] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*, 2014.

[103] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010.

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

[105] T. Ahmad and P. N. Papadopoulos, "Prediction of cascading failures and simultaneous learning of functional connectivity in power system," in *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–5, 2022.

[106] B. Tang, "Orthogonal array-based latin hypercubes," *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1392–1397, 1993.

[107] R. Preece and J. V. Milanović, "Efficient estimation of the probability of small-disturbance instability of large uncertain power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1063–1072, 2016.

[108] V. Krishnan, J. D. McCalley, S. Henry, and S. Issad, "Efficient database generation for decision tree based power system security assessment," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2319–2327, 2011.

[109] W. Duch, K. Grudzinski, and G. Stawski, "Symbolic features in neural networks," in *In Proceedings of the 5th Conference on Neural Networks and Their Applications*, pp. 180–185, 2000.

[110] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[111] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems* (D. Touretzky, ed.), vol. 2, Morgan-Kaufmann, 1989.

[112] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018.

[113] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[114] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[115] "TensorFlow: Large-scale machine learning on heterogeneous systems." Software available from tensorflow.org.

[116] "FlatBuffers: Flatbuffers white paper." Software available from https://google.github.io/flatbuffers/.

[117] L. Y. Pratt, "Discriminability-based transfer between neural networks," in *Advances in Neural Information Processing Systems* (S. Hanson, J. Cowan, and C. Giles, eds.), vol. 5, Morgan-Kaufmann, 1992.