University of the Witwatersrand

Master of Science in Engineering

Machine Learning Using PMU Data to Perform Eigenvalue Prediction

Teboho Machabe - 0311109T

School of Electrical Engineering and Information Engineering

Supervised by

Prof. Ken Nixon and Ms. Ellen De Mello Koch

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg in fulfilment of the requirements for the degree of Master of Science in Engineering.

30 August 2021

Abstract

A modern power system faces daily challenges due to changes in load and generation patterns. These changes contribute significantly to the secure operation of the electrical network and power system stability and small signal stability. The work presented extends and contributes to research in power system stability and focuses on the predicting inter-area low-frequency oscillations on the integrated power system. Previous work in this area has made use of load flow models and traditional machine learning techniques such as linear regression and support vector machines to predict disturbances. The introduction of synchrophasor technology through the use of Phasor Measurement Units (PMUs) has, amongst other benefits, contributed to the monitoring of low-frequency oscillations. However, high resolution time series data provided by the PMUs require advanced methods of analysis to achieve the prediction target. In the research presented, data from five different PMU devices on the Southern African power system is used to predict eigenvalue locations on the eigenvalue plane using a Recurrent Neural Network technique known as Long Short Term Memory (LSTM). It is shown that the LSTM algorithm can accurately predict a small signal stability disturbance using a preceding window length of 20 seconds before the actual event occurs. Eigenvalues are estimated from the measured PMU data using the Matrix Pencil Method of eigenvalue estimation. LSTM showed an 80% accuracy in predicting the eigenvalues on the eigenvalue plane. Recommended future work would be to investigate the use of PMU devices for power system inertia prediction and to further determine the location of the disturbance on the network.

Declaration

I declare that this research proposal is my own unaided work. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

....

(Signature of Candidate)

30th day of **August** year **2021**

Acknowledgments

I would like to thank my supervisor, Prof. Ken Nixon, and co-supervisor, Ms. Ellen De Mello Koch for their expertise, guidance, time and encouragement throughout this journey.

A special heartfelt thanks to Itani Phafula, my colleague and fellow student turned mentor. Thank you for your wise counsel and always being available when I needed you.

I would like to thank my mother, Puseletso Machabe, and father, Toko Machabe, for their continued support, love and prayers not only through this process of my career but for my entire existence. Words cannot express my gratitude. God bless you abundantly.

I would like to thank my friends for all their support, believing in me and always urging me to never give up.

I would like to express my deepest thanks and sincere appreciation to Thabo Modisane and Andrew Hank who have played a pivotal role in completing this dissertation.

To my wonderful kids, Orefile-Bontle and Lehakoe, this is dedicated to you with love. To my beautiful wife, number one cheerleader and best friend, Kamogelo Machabe, thank you for your undying love and allowing me the time to complete this. I couldn't have completed this mammoth task without you. I love and appreciate you with all of my heart. I thank God for you. **"All of the family"**.

Finally, to God be the Glory. It is not my will but Yours. I am eternally grateful for Your unconditional love and mercy. Amen!!

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence. 26

ANN Artificial Neural Network. 31

AR Autoregressive. 43

ARIMA Autoregressive Integrated Moving Average. 33

ARMA Autoregressive Moving Average. 31

AVR Automatic Voltage Regulator. 1

BPA Bonneville Power Administration. 20

CSV Comma Separated Values. 54

DFT Discrete Fourier Transform. 14

FFT Fast Fourier Transform. 31

GPS Global Positioning System. 13

LPM Linear Predication Model. 44

LSTM Long Short Term Memory. 33

MAR Multivariate Autoregressive. 32

MP Matrix-Pencil. 38, 49

NN Neural Networks. 32

OCSVM One Class Support Vector Machine. 32

PDX Power Dynamics Extractor. 23

PMU Phasor Measurement Unit. 2

POD Power Oscillation Damper. 2

PSS Power System Stabiliser. 1

RNN Recurrent Neural Networks. 33

ROCOF Rate Of Change Of Frequency. 13

SAPP Southern African Power Pool. 20

SCADA Supervisory Control And Data Acquisition. 17

SPDC Substation Phasor Data Concentrator. 17

SVC Static Var Compensator. 2

SVR Support Vector Regression. 33

TSO Transmission System Operators. 20

WAMS Wide Area Monitoring System. 13

ZESA Zimbabwe Electricity Supply Authority. 2

1 Introduction

This Chapter introduces the topic of the dissertation and provides a high-level overview of the rest of the Chapters.

1.1 Background

The primary purpose of a power system is to generate electrical energy, transmit electrical energy over vast distances, and finally distribute to the end customer through the grids. The primary part of the power system is made up of different types of generation where the most traditional is the rotating machine or the synchronous generator, which utilises fossil fuel to generate electricity.

An important condition for the power system's satisfactory operation is that all synchronous machines should operate in synchronism. This is related to the dynamics of generator rotor angles, power angles as well as the balance between generation and load which have an impact on stability. In conducting the analysis of stability, the response of the power system following a disturbance is a matter of concern [1].

A power system experiences a number of disturbances at all times which could be small or large. Large disturbances on the network can be in the form of large load increases, loss of tie lines or loss of generating units [1]. In the interconnected system, some machines will speed up while other machines slow down in order to maintain synchronism following a disturbance. However, in the absence of adequate control mechanisms, some generators can lose synchronism or go "out-of-step" [2].

The use of Automatic Voltage Regulator (AVR) is introduced to mitigate against this problem. The AVR makes adjustments to the field voltage which in turn controls the electrical speeds of synchronous generators within allowable limits thereby maintaining synchronism. The AVR, however, cannot achieve "fine adjustments" in its attempt to control the speed. The Power System Stabiliser (PSS) is utilised to address the generator's fine speed changes, thereby damping out the power oscillations, also known as low-frequency oscillations [2].

In the case of small disturbances, these are related to load changes where the system adjusts itself to the changing conditions and finds a new stable operating point. Ideally, the power system, with modern control strategies, should clear these disturbances quickly and return to the original or new stable operating point in a short duration of time. Although these small disturbances, of low-frequency oscillations in nature, are an inherent property of the power system, if not managed correctly, they could lead to a partial or a total system blackout [1, 2].

When two or more net groups exchange energy in an inter-connected power system, inter-area oscillations will likely be observed. This is associated with the swinging of machines in one part against machines in another part of the power system. These slow damped power oscillations with low frequency in nature, usually between 0.1 -2.0 Hz, are termed small-signal disturbances [2]. Small-signal disturbance analysis identifies the causes of poorly damped or unstable power system oscillations where modal analysis based on the computation of eigenvectors and eigenvalues is utilised for the analysis [3].

1.2 Problem Statement

The 0.3 Hz small-signal inter-area oscillatory mode is well-known oscillation mode between the two area system; namely, ESKOM in South Africa and Zimbabwe Electricity Supply Authority (ZESA) in Zimbabwe. This mode has been a familiar phenomenon over the past number of years. If negatively damped or of growing magnitude, the oscillations pose a significant risk to the stable operation of the power system when they are not sufficiently damped.

They are particularly pronounced during cases when the Static Var Compensator (SVC), Power Oscillation Damper (POD), PSS or other active controllers are out of service. Although a vast amount of research work has been done on predicting system disturbances using neural networks, the combination of the prediction using machine learning and the Phasor Measurement Unit (PMU) data, mainly the 0.3 Hz inter-area mode is not known. Machine learning algorithms offer valuable tools for harnessing and interpreting large amounts of PMU data to assist in the power system operation decision-making process. The benefits of predicting the small-signal instability on the network would be to offer increased situational awareness of the system, therefore improving the stable system operation.

1.3 Research Question

The research question addressed by this dissertation is:

> **Can machine learning techniques with PMU data for prediction of the 0.3 Hz oscillatory mode improve the situational awareness of the interconnected power system?**

Over and above the question, the following points are important considerations:

- What aspects of the PMU data are relevant and what feature extraction techniques and features are essential for stability prediction?
- What machine learning techniques can be used for prediction of the eigenvalue region?
- What accuracy measures or indicators can be implemented to quantify eigenvalue area prediction in determining the system stability?

1.4 Structure of the Dissertation

This dissertation is composed of eight Chapters. Chapter 1 introduces the current problems and reviews the problem of power system oscillations. Chapter 2 highlights the fundamentals of power system stability and how it affects the power network's integrity. It further addresses the small-signal stability and the eigenvalue determination, the damping ratio and the mode oscillation. Chapter 3 describes the synchrophasor technology which comprises of PMUs. It further presents a case study where the benefits of PMUs are demonstrated. Chapter 4 shows the different machine learning algorithms highlighting the differences in the application for this research. Chapter 5 compares the different eigenvalue estimation methods and their respective challenges. From this Chapter, one estimation method will then be selected to generate eigenvalues for further studies. Chapter 6 looks at the machine learning combined with the PMU data to predict the small-signal stability. Chapter 7 shows the prediction simulation results and these are discussed in detail. Finally, Chapter 8 concludes the dissertation and recommends future work.

1.5 Conclusion

This Chapter provided the background of this research. The problem highlighted, is the inter-area small-signal disturbances experienced by the power system. These disturbances, if not managed correctly, could lead to a partial or a total system blackout. The research question seeks to investigate the use of machine learning and PMU data to address the problem. This Chapter concludes by outlining the structure of the dissertation.

The following Chapters address the fundamental considerations in power system stability important to this dissertation.

2 Power System Stability

This Chapter introduces the various aspects of power system stability and their significance in the secure operation of the network. The eigenvalue theory is also discussed.

"*Power system stability* is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most of the system variables bounded so that practically the entire system remains intact" [4, 5]. The disturbances could be due to faults, sudden load changes or a combination of both these occurring on the system. Power system stability can be broadly classified into rotor angle, voltage and frequency stability as indicated in Figure 1 [5].

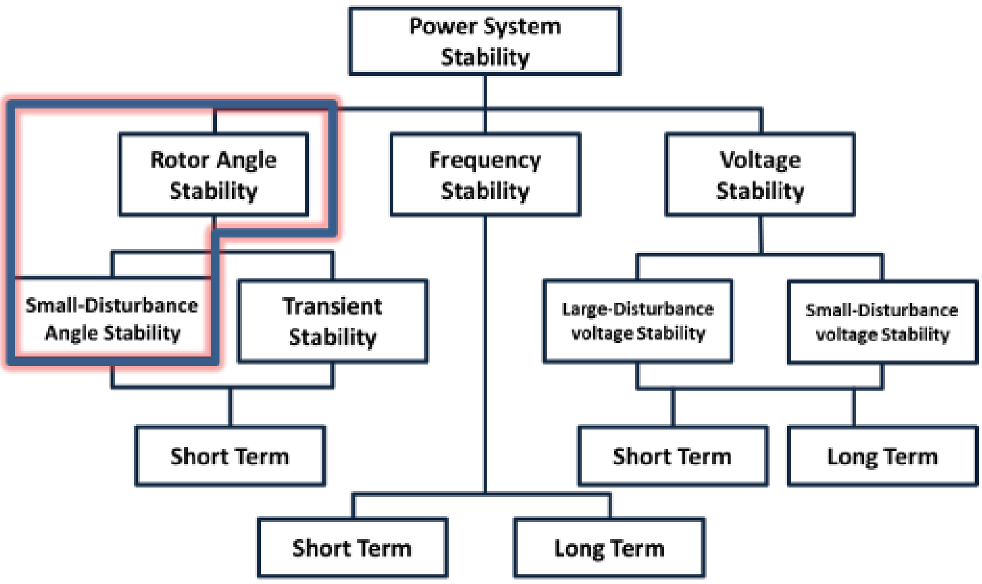

Figure 1: Classification of power system stability highlighting the rotor angle and small disturbance angle stability, derived from [1]

Though, stability is classified into rotor angle, voltage and frequency stability they need not be independent isolated events. A voltage collapse at a bus can lead to large excursions in rotor angle and frequency. Similarly, large frequency deviations can lead to large changes in voltage magnitude [1].

2.1 Voltage Stability

Voltage stability "is the ability of the system to maintain steady state voltages at all the system buses when subjected to a disturbance. If the disturbance is large then it is called a large-disturbance voltage stability and if the disturbance is small it is called a small-disturbance voltage stability" [6].

Unlike angle stability, voltage stability can also be a long term phenomenon. The main difference between voltage stability and angle stability is that voltage stability depends on the balance of reactive power demand and generation in the system whereas the angle stability mainly depends on the balance between real power generation and demand [1].

2.2 Frequency Stability

Frequency stability "refers to the ability of a power system to maintain steady frequency following a severe disturbance between generation and load". It depends on the ability to restore equilibrium between system generation and load, with minimum loss. Frequency instability may lead to sustained frequency swings leading to tripping of generating units or loads. During frequency excursions, the characteristic times of the processes and devices that are activated will range from fractions of seconds, like under frequency control; to several minutes, corresponding to the response of devices such as prime movers. Frequency stability can therefore be represented be a short-term phenomenon or a long-term phenomenon [6].

2.3 Rotor angle stability

Rotor angle stability "is the ability of the system to remain in synchronism when subjected to a disturbance". The rotor angle of a generator depends on the balance between the electromagnetic torque due to the generator electrical power output and mechanical torque due to the input mechanical power through a prime mover. Remaining in synchronism means that all the generators' electromagnetic torques are exactly equal to the mechanical torque in the opposite direction. If the balance between electromagnetic and mechanical torque is disturbed in a generator, due to disturbances in the system, then this will lead to oscillations in the rotor angle. Rotor angle stability is further classified into small disturbance angle stability and large disturbance angle stability [6].

2.3.1 Large-disturbance or transient angle stability

Large-disturbance or transient angle stability "is the ability of the system to remain in synchronism when subjected to large disturbances". Large disturbances can be faults, switching on or off of large loads, large generators tripping etc. When a power system is subjected to a large disturbance, it will lead to large excursions of generator rotor angles. Since there are large rotor angle changes the power system cannot be approximated by a linear representation like in the case of small-disturbance stability [6].

2.3.2 Small-disturbance or small-signal angle stability

Small-disturbance or small-signal angle stability "is the ability of the system to remain in synchronism when subjected to small disturbances" [6]. Small disturbances can be small load changes like switching on or off of small loads, line tripping, small generators tripping etc. Due to small disturbances, there can be two types of instability: non-oscillatory instability and oscillatory instability. In non-oscillatory instability, the rotor angle of a generator keeps on increasing due to a small disturbance. In the case of oscillatory instability, the rotor angle oscillates with increasing magnitude. In the absence of insufficient damping torque, the rotor angle oscillations of increasing magnitude will be observed. This is referred to as negative damping. High gain automatic voltage regulators may cause grid connected generators to experience insufficient damping to system oscillations [2].

2.4 Characteristics of small signal stability problems

Small disturbance rotor angle stability problems may be local or global in nature [1, 5].

1. Local Problems are associated with rotor angle oscillations of a single generator or a single plant against the rest of the power system. Such oscillations are called local plant mode oscillations. These may also be associated with oscillations between the rotors of a few generators close to each other. Such oscillations are called inter-machine or inter-plant mode oscillations. The local plant mode and inter-plant mode oscillations have frequencies in the range of 0.7 to 2.0Hz.

2. Global Problems are caused by interactions among large groups of generators and have widespread effects. They involve oscillations of a group of generators in one area swinging against a group of generators in another area. Such oscillations are called inter-area mode oscillations. Large interconnected systems usually have two distinct forms of inter-area oscillations:

    (a) A very low frequency mode involving all the generators in the system. The frequency of this mode of oscillation is of the order of 0.1 to 0.3Hz.

    (b) Higher frequency modes involving subgroups of generators swinging against each other. The frequency of these oscillations is typically in the range of 0.4 to 0.7Hz.

2.5 Eigenvalue Analysis

As described above, small-signal stability is a power system's ability to maintain synchronism when subjected to small disturbances [5]. A disturbance is considered small if the equations that describe the system's resulting response can be linearised for analysis. The dynamic behaviour of a complex non-linear power system may be represented by Equations (1) and (2).

$$\dot{x} = f(x, u, t) \tag{1}$$

$$y = g(x, u, t) \tag{2}$$

Where x and u are the state and input column vectors respectively and t is time in seconds.

When Equations (1) and (2) are linearised using the Taylor series they yield the following [1]:

$$\Delta\dot{x} = A\Delta x + B\Delta u \tag{3}$$

$$\Delta y = C\Delta x + D\Delta u \tag{4}$$

Where the following matrices A, B , C and D are the plant, input, output and the feedforward respectively. Δx, Δy and Δu represent the state vector, output vector and the input vector respectively [1].

Taking the Laplace transformation of Equations (3) and (4) results in the following equations in the frequency domain:

$$s\Delta x(s) - \Delta x(0) = A\Delta x(s) + B\Delta u(s) \tag{5}$$

$$\Delta y(s) = C\Delta x(s) + D\Delta u(s) \tag{6}$$

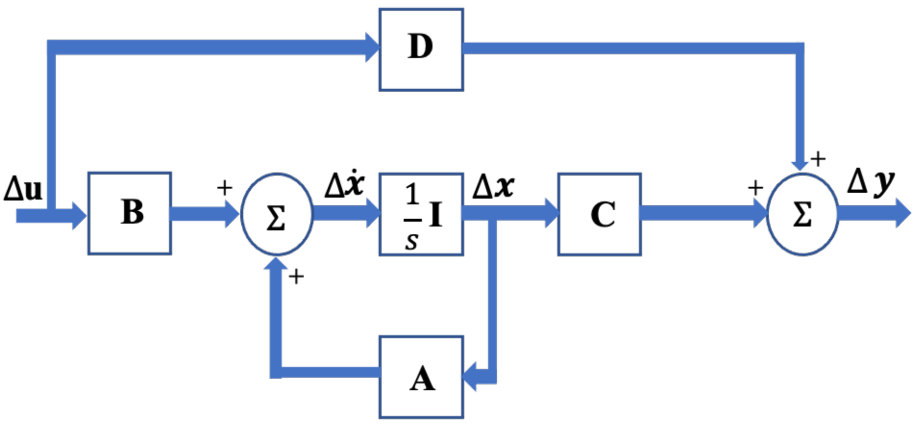

Figure 2: Block diagram of a State-space representation derived from [1]

A representation of the state space in a block diagram form is shown in Figure 2. For the state matrix **A**, the eigenvalues are given by the values of the scalar parameter λ for which a solution has at least one non-zero component to the equation [1]:

$$A\Phi = \lambda\Phi \tag{7}$$

where **A** is an nxn matrix and Φ is an nx1 vector.

To find the eigenvalues of **A**, Equation (7) may be written as follows:

$$(A - \lambda I)\Phi = 0 \tag{8}$$

$$det(A - \lambda I)\Phi = 0 \tag{9}$$

The n solutions of $\lambda = \lambda_1, \lambda_2, \lambda_3, ..., \lambda_n$ are the eigenvalues of A. The matrix **A** is referred to as the *characteristic equation*. The roots of the characteristic equation determine the stability in a non-linear system by the eigenvalues of **A**:

i. Eigenvalues with negative real parts indicate that the system is asymptotically stable.

ii. Eigenvalues with at least one positive real part indicates that the system is unstable.

iii. Eigenvalues with real parts equal to zero is inconclusive on the basis of the first approximation.

The damping is given by the real component of the eigenvalue while the imaginary gives the frequency of oscillation. A positive real part represents oscillations that are increasing in amplitude. A negative real part represents damped oscillations [1]. For a **complex pair of eigenvalues**:

$$\lambda = \sigma \pm j\omega \tag{10}$$

The real part σ represents the damping of the mode and the imaginary represents the frequency f of the oscillation in Hz given by:

$$f = \frac{\omega}{2\pi} \tag{11}$$

This represents the actual or damped frequency. The **damping ratio** is given by:

$$\zeta = \frac{-\sigma}{\sqrt{\sigma^2 + \omega^2}} \tag{12}$$

The damping ratio ζ is a dimensionless measure describing the rate of decay of the amplitude of the oscillation.

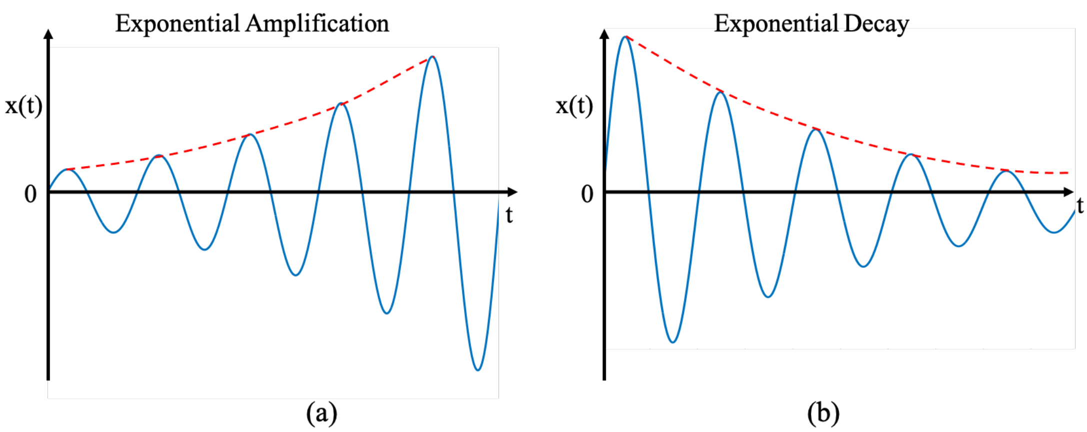

Figure 3: (a) Dynamically unstable or negatively damped sinusoid (b) Dynamically stable or positively damped sinusoid [1]

The damping ratio indicates how stable the system is, the higher the positive value of a damping ratio, the more stable the system is for a given oscillation. Figure 3 depicts the different stability scenarios.

2.6 Conclusion

This Chapter presented the theory behind power system stability which can be subdivided into frequency, voltage, and rotor angle stability. It further focuses on the characteristics of small-signal stability and the eigenvalue analysis. The dynamic behaviour of the power system can be characterised by a set of linear equations wherein the eigenvalues are determined. Eigenvalues are particularly important in this study as they determine the nature of a disturbance and are a central part of what this study aims to predict. Fundamental equations to eigenvalue analysis of signals are equation (10) and equation (12), which represent real imaginary eigenvalues and the damping ratio. Eigenvalues provide information about the damping and frequency parameters of the oscillation which will be used in the following Chapters.

The next Chapter will focus on synchrophasor technology and its application in the monitoring of the power system dynamic behavior.

3 Synchrophasor Technology

This Chapter brings to the discussion the industry developments in wide-area monitoring and synchrophasor technology. It further highlights the benefits of Phasor Measurement Units on the modern power system.

3.1 Overview

A PMU is a device or a functionality embedded in other devices such as protective relays which measures electrical signals in a power system, particularly voltage and current signals. PMUs are highly dependent on the availability of a time source provided by a Global Positioning System (GPS) clock to time-stamp the measured voltage and current signals and enable synchronisation of measurement data from vast locations on the network. Furthermore, PMUs can provide frequency and Rate Of Change Of Frequency (ROCOF) measurements.

The architecture supporting measurement system is referred to as Wide Area Monitoring System (WAMS). WAMS implies a set of modern digital metering devices, such as PMUs, along with communication infrastructures. The system is designed to acquire, transmit and process data over wide geographic locations maintaining the same time frame. The use of synchrophasors has made more advanced real-time monitoring, protection and control applications possible, therefore improving the electric power system stability and resilience.

3.2 History

Around the year 1970, transmission line microprocessor-based relaying experienced challenges in the computational capabilities to perform relay function calculations. Further research for reducing the computational requirements suggested a solution based on symmetrical component analysis of line voltages and currents. This gave rise to the development of the PMU as it is known. The modern phasor measurement system received substantial contributions, in 1977, when calculations of the positive-sequence voltages and currents using Discrete Fourier Transform (DFT) algorithm were presented [7]. The subsequent year (1978), the GPS project began and it provided the possibility of synchronising measurements. The very first PMU prototype using GPS was designed at Virginia Tech in the early 1980's. PMUs were then commercially manufactured in 1991 and further led to the development of "IEEE Standards for Synchrophasor Measurements for Power Systems" which introduced governing of all issues relating to phasor measurements [7].

3.3 Fundamentals of PMUs

A PMU is an electronic device with the technology to provide phasor information (both magnitude and phase angle) in real time. The phase angle is referred to a global reference time and this allows for the capturing of the wide-area snap shot of the integrated power system. This technology has provided the power system operator with knowledge of the power system's real time behavior, which is essential in mitigating system disturbances and power system blackouts [8].

A sinusoidal waveform can classify the mathematical definition of a phasor with constant frequency and magnitude, $x(t)$:

$$x(t) = X_m \cos(\omega t + \phi) \tag{13}$$

where X_m represents the signal peak value, $\omega = 2\pi f$ is the angular frequency, and ϕ is the signal's phase angle. This sinusoid can be represented in phasor form by:

$$x(t) = \frac{X_m}{\sqrt{2}} e^{j\phi} = \frac{X_m}{\sqrt{2}} (\cos\phi + j\sin\phi) \tag{14}$$

where the magnitude of the phasor is the RMS value $\frac{X_m}{\sqrt{2}}$ of the sinusoid. The phase has to be correctly referenced to the initial time instant which it is dependent. Furthermore, a positive phase angle is measured in a counterclockwise direction from the real axis. A representation of the phasor and the sinusoid is shown in Figure 4 [8].

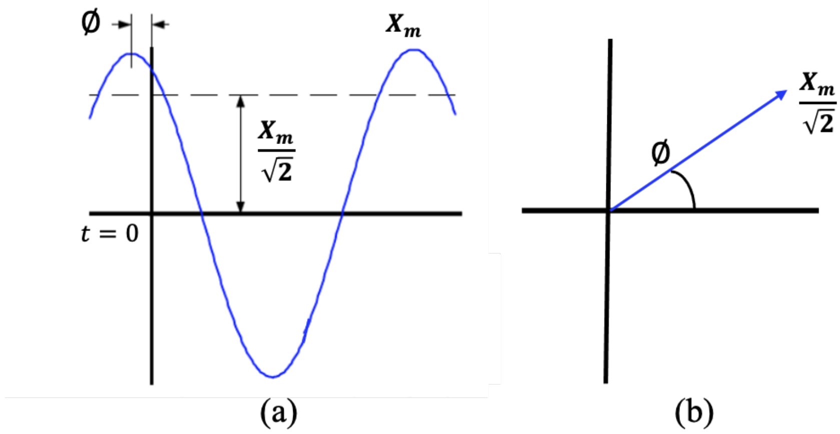

Figure 4: Phasor representation of a sinusoidal signal (a) Sinusoidal signal (b) Phasor representation, derived from [8]

The main aim of the PMU is to provide time referenced phasor measurements of voltage and current at a substation. These measurements are obtained from the respective voltage and current transformers. Over and above these measurements, the PMU also provides the system frequency and the ROCOF measurements. Figure 5 shows the representation of a typical hardware structure of a PMU device.

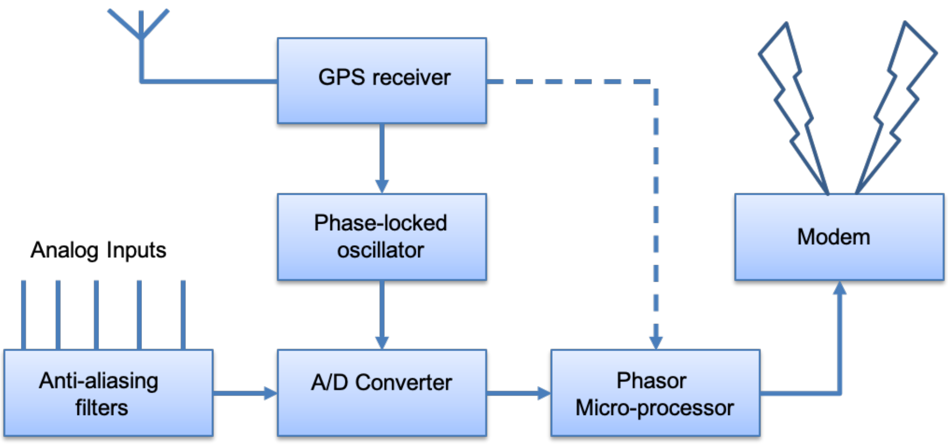

Figure 5: Illustration of the PMU Hardware layout [9]

Each PMU comprises many components, and these include the anti-aliasing filter, the A/D converter, phase-locked oscillator, GPS receiver, phasor micro-processor and a modem. The function of each of these components is described as follows:

- **Anti-aliasing filter:** In sampling of the data through measurement, there will be an overlap of the signals if the sampling frequency is not more than two times the analysed signal frequency causing errors in the samples. This phenomenon is referred to as aliasing and the purpose of the anti-aliasing filter is to avoid this condition. This principle of anti-aliasing of the sample data is per the Nyquist criterion [9].
- **Analog to Digital (A/D) Converter:** The measured or sampled data is analog. The A/D converter aims to covert this data to digital format for further processing by the phasor micro-processor.
- **GPS Receiver:** The GPS receiver, in some cases, can be a stand-alone device which has the unique function of transmitting a continuous and systematic time signal as well as a one pulse per second signal to the PMU. The GPS allows for the measured data to be time stamped [9].
- **Phase-Locked Oscillator:** The phase-locked oscillator which is locked to the GPS clock pulse controls the sampling intervals. It can check time automatically to ensure the lock is maintained [9].
- **Phasor Micro-Processor:** The micro processor uses a mathematical algorithm that has been developed to estimate the measured phasors and calculate the state [9].
- **Modem:** The final stage of the PMU hardware configuration is the modem or other suitable communication streams. This makes data streaming to a centralised location through the telecommunication network possible [9].

In order to calculate the positive sequence measurements for the PMU, three-phase voltage and three phase currents are required. At a substation where the PMU is located, these measurements can be supplied by feeder bays or even transformers. Different manufacturers of the PMU will have a different number of input channels as analog inputs. For example, a PMU may have a total of four feeder bays on the same voltage magnitude as inputs. A complete layout of the PMU structure across the power system is shown in Figure 6.

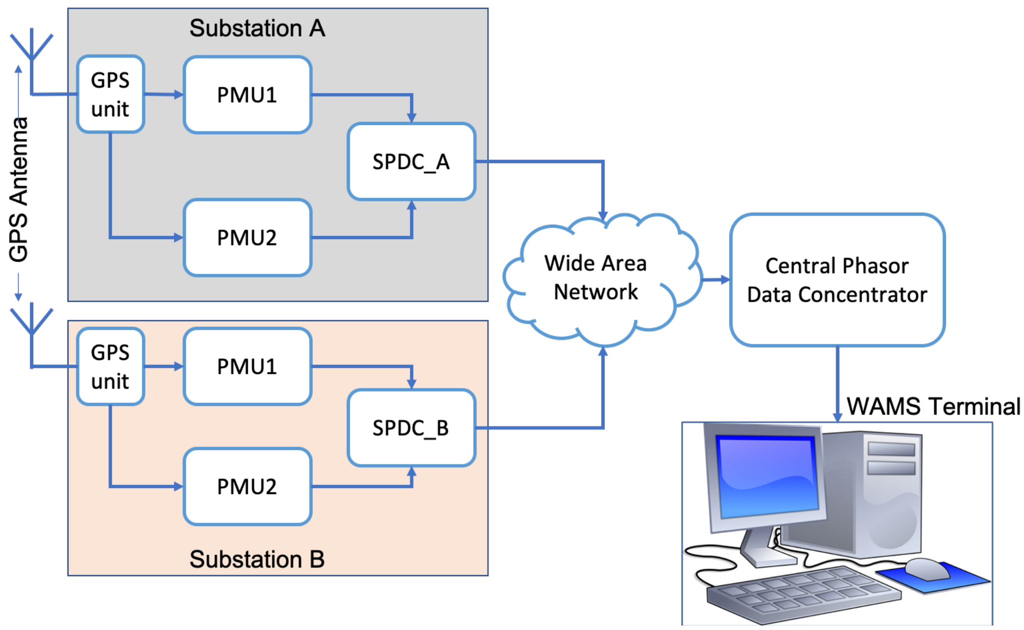

Figure 6: WAM System layout with multiple substations connected to the Control Centre

A substation will have one or several PMU devices depending on the number of feeder bays to be monitored. This time-stamped phasor data is then packaged in a Substation Phasor Data Concentrator (SPDC) which streams this data packet through the telecommunication wide area network to a control centre. The measured phasor analytics can be visualised and analysed at the WAMS terminal [10].

3.4 WAMS Applications

Synchronised phasor measurements play an essential role in real-time system monitoring due to the high-resolution measurements. Phasor measurements technology is not a replacement of the Supervisory Control And Data Acquisition (SCADA) system but rather a perfect compliment of that system. The advantage that synchronised phasor measurements offer over the SCADA measurements is that tracking of dynamic and transient magnitude and phase shifts in the monitored megawatt, megavar, frequency and voltage values is now possible. It has also offered an enhanced view of the power system controllers and assisted in their decision-making process. Some of the benefits to the transmission system operators include [10]:

- Improved State Estimation
- Post Event Analysis
- Monitoring of the Power System Stability
- Model validation
- Re-synchronise islanded networks

3.4.1 Improved State Estimation

SCADA system provides the state estimation with measurements to obtain the bus voltages' estimates on the power system. A valid state is defined by the minimum error magnitude for a given set of measurements. In order to accurately estimate and match a real network, the state estimator uses the following data [10]:

- Line megawatts and megavars
- Bus voltages
- Loads in megawatts and megavars
- Transformer tap position

The state estimator's accuracy and robustness are critical in the efficient operation of the online network assessment tools. A PMU measures bus voltages and angles. Measurements from the PMU can be fed into the state estimator together with the SCADA measurements to offer even more accurate network estimation.

3.4.2 Post Event Analysis

Following a major disturbance, a thorough disturbance investigation and analysis is conducted to determine the events' root cause and further put measures in place to prevent a repeat of such disturbances. This is also referred to as *post-moterm analysis*. PMUs are very effective for disturbance analysis due to the data being time-stamped and synchronised across the entire network making the tracking of dynamic transients possible [10].

3.4.3 Monitoring of the Power System Stability

The power system monitoring is critical for the safe operation of the integrated power system. Voltage stability remains one of the significant concerns in planning and operating the network, particularly for networks operated with a small margin. The development of synchronised phasor measurements has realised real-time voltage stability monitoring, thereby improving the systems security and power transfer issues.

During an event or a disturbance on the network, some lines may be heavily loaded. When there is insufficient reactive power to support the system voltages, the system may reach a voltage collapse situation. The high-resolution PMU data can be utilised in special applications to assist controllers' decision-making process during such scenarios [11]. Other evolving PMU applications such as small-signal stability monitoring are also being used by control centres worldwide to observe parameters such as system oscillations and damping thereof [10].

3.4.4 Model validation

Power system engineers use various offline analysis tools to model the network for network planning, contingency analysis, protection studies etc. The high-resolution data provided by PMUs is beneficial in evaluating the accuracy of these models and can improve the confidence level of the personnel in the analysis and planning phase of the power system [10].

3.4.5 Re-synchronise islanded networks

The use of PMU recorded data for re-synchronisation checks is well documented in black-start scenarios. In [12], the high accuracy PMU measurements contributed significantly to synchronising an island where the rate of change of frequency in the islanded network was used to resolve the time delay concern in data transmission that caused breaker closing at unacceptable voltage phase angles [10].

3.5 WAMS across the World

There has been a large deployment and commissioning of phasor measurement units throughout many Transmission System Operators (TSO) worldwide. This deployment is mainly credited to the PMUs ability to provide greater system-wide visibility of the network.

The Danish Transmission system has four PMUs installed on their 400kV and on the 132kV voltage levels. These PMUs were important in analysing the event on the 8th January 2005 where a severe storm passed Denmark, and the wind speeds were so high that the installed wind turbines disconnected from the network as a protection mechanism [13].

The Bonneville Power Administration (BPA), in the USA, implemented a PMU-based architecture in early 2000. In 2003, America experienced the largest power system blackout which affected 50 million people. In response to this, the New York Independent System Operator installed several PMUs to avoid similar catastrophic events occurring [14].

China has also made significant progress in the deployment of PMUs on its electrical network. By 2013, more than 2000 PMU sets had been deployed in China across all 500kV substations and on a number of other power plants and substations of interest. Important dynamic information about the condition of the power system is provided by more than 30 WAMS centre stations [15]. Other countries that have seen a deployment of PMUs on their power networks include Mexico and Brazil [16, 17].

3.6 PMU Case Studies - South Africa

The Southern African Power Pool (SAPP) consists mainly of the hydro power system in the North and a mostly thermal system in the South. A simulation-based small-signal study conducted in 2014 established that there exists a 0.3 Hz inter-area oscillatory mode in the SAPP network.

These oscillations are generally caused by hydro units ramping up in the evening to balance the evening demand and keep the frequency stable. However, a unique protection scheme is activated if the power swings' magnitude exceeds the pre-defined limits on the Eskom-ZESA AC-tie-line. This is so that the two systems' (North and South) integrity is maintained and to guarantee system security [18].

A power system event was observed on the power system by the Eskom WAMS on June 2018, which subsequently led to poor damping of the 0.3 Hz oscillatory mode on the network. The monitoring from the WAMS assisted in developing the sequence of events of that incident shown in Figure 7. A multi-unit trip occurred on the Eskom network causing a frequency decline. The frequency was within the required limits; however, an oscillation of a negative damping in nature was experienced. The frequency remained above 49.5 Hz.

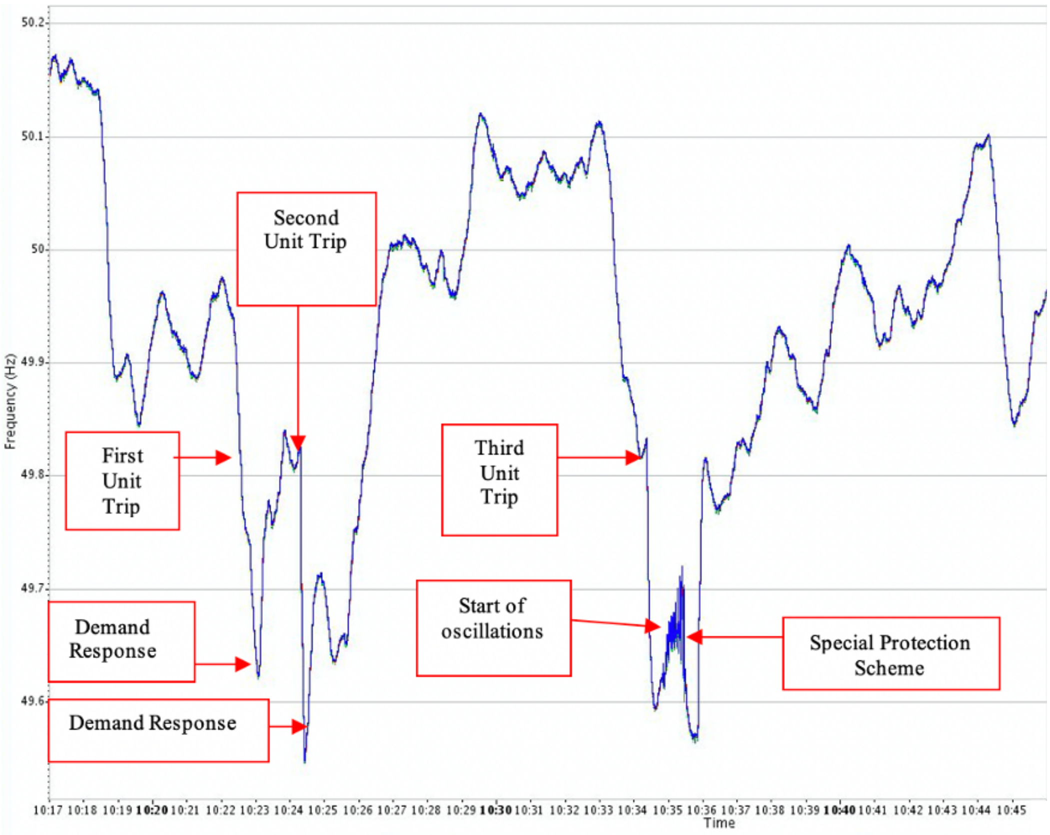

Figure 7: Frequency signal showing the multi-unit trip, system response, small signal oscillations the response of the special protection scheme (annotated screenshot from software)

The event begins with a unit trip at around 10:22 am. This causes an imbalance between the power generated and the load demand, and the frequency starts to decline. Due to the system frequency decline, demand response is activated (at 10:23 am) to recover the frequency and prevent low frequencies on the network, which will subsequently lead to a frequency collapse of the network. A second unit trips less than a minute later, and again demand response is activated (at 10:24 am) to recover the frequency to allowable limits. The system stabilises and maintains this condition for about 10 minutes when the third unit trips. The frequency begins to decline and as the demand response is activated for the third time to recover the frequency, small signal oscillations are observed. A closer look at the power oscillations is shown in Figure 8.

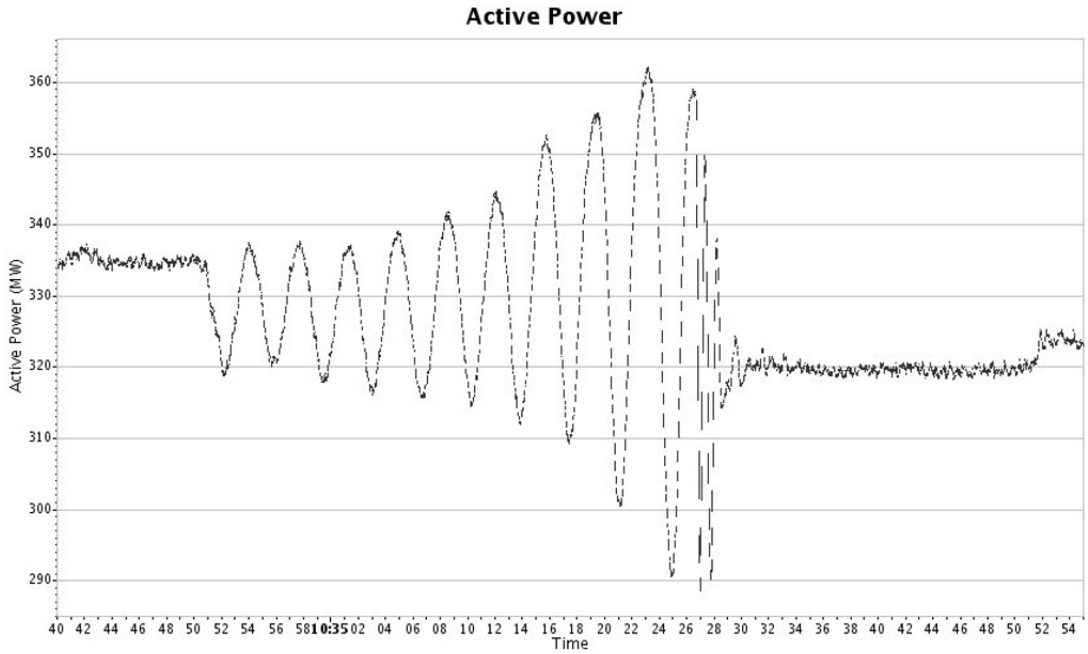

Figure 8: Power Oscillations as observed from the PMU located at the inter-connection point of the network (screenshot from software)

The oscillation continues for 36 seconds until a special protection scheme is activated to safe guard the integrity of the power system by tripping the affected line where these oscillations are observed. As can be seen, the oscillations are growing in magnitude and hence they are classified as negatively damped oscillations. The risk that these oscillation pose to the network is that they can lead to power system blackouts. This was the case in 1996 on the North American Western Interconnected System where the whole power system collapsed due to inter-area oscillations with a frequency of 0.23 Hz occuring [19]. The WAMS is well positioned to provide information on the oscillation damping properties shown in Figure 9.

Power Dynamics Extractor (PDX) is an analysis technique which characterises observable modes of oscillation. It determines the system stability through random perturbations that appear in the measured signals in the power system. The specialised software application tool that integrates with the WAMS is equipped with the PDX functionality. The dynamic characteristic is derived from a window of the most recent data using PDX processing. PDX1-3 represents a window length of 3 minutes and updates every 5 seconds while PDX2-n represents a window length of 20 minutes with an update rate of 20 seconds. For accurate and stable results, a longer window of analysis is used. It does, however, not respond rapidly to change in conditions [10].

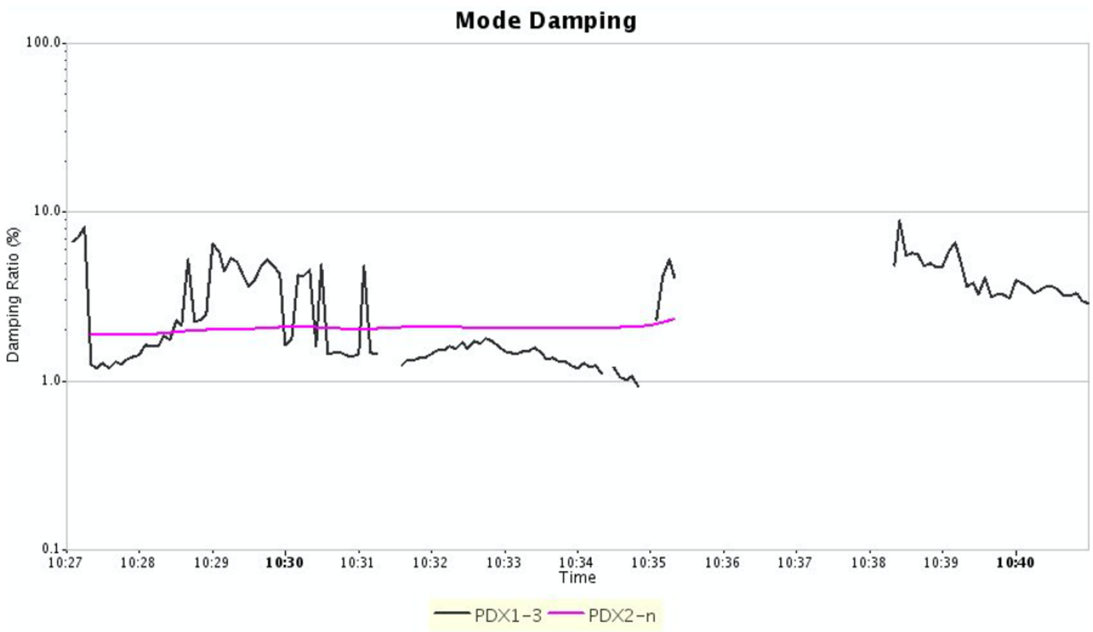

Figure 9: Damping ratio showing the 3-minute window (PDX1-3) and the 20-minute window (PDX2-n) at the time of the disturbance (screenshot from software)

The damping ratio is around 2% for this oscillation if the PDX2-n is considered. The allowable damping ratio is not well defined; however, anything between 2%-4% is regarded as adequate damping. Anything below 2% is considered poorly damped and is viewed as an alarming point of the power system [10].

3.7 PMUs and Big Data

The introduction of smart technology on the traditional power system in smart meters and PMUs has brought about large volumes of data requiring extensive data storage services. Due to the high-resolution measurement data or the fast sampling rate of these smart devices. The traditional SCADA system measures only once every two to four seconds [20]. The PMU, on the other hand, can sample between 10 and 60 times in one second depending on the system frequency [21]. For example, the number of PMUs in the US were about 1100 units by the end of 2014 and these collectively result in a data volume of 310GB everyday [22]. This, however, is not a deterrent to the use of smart technology but rather provides a new opportunity in the field of artificial intelligence and machine learning [22].

3.8 Conclusion

PMUs, their fundamentals as well as the architecture of their hardware were introduced in this Chapter. The benefits, which include post-event analysis and model validation, were highlighted and discussed in detail. The high-resolution data provided by the PMUs makes this system an adequate complement to the traditional SCADA system. The case study demonstrated the disturbances observed by these high-resolution measurements, which in the absence of PMUs, would have normally not been observed. The current SCADA system has a reporting rate of about 2 to 4 seconds. A 20-second long disturbance would therefore only be represented by five data points as compared to the 20ms reporting rate of the PMUs which would represent the same disturbance in 1000 data points. Important information critical to the determination of the power system stability can hence be acquired through the use of PMUs. The requirement for an improved observability of the modern power system has seen an increase in the number of PMU devices across the world and South Africa.

The next Chapter discusses the machine learning theory, which plays an essential part in predictive data science.

4 Machine Learning Algorithms

This Chapter discusses the machine learning theory, the relationship to artificial intelligence and the various Artificial Intelligence (AI) algorithms. The machine learning application in the power system stability prediction space and the different algorithms will also be discussed.

4.1 Background

In recent times, machine learning has become a critical topic that has enabled various organisations to find innovative methods to better understand enormous data sets. Through the use of appropriate machine learning models, changes into the various sectors can be accurately predicted. The ability to predict the future provides valuable insight into planning and applying corrective measures to be taken before the actual event occurs. In simple terms, the machine learns, in the form of AI, from previously supplied data rather than explicit programming [23].

There are some algorithms available for use in the machine learning space for predictive purposes. However, the same process will be executed in using all of these algorithms. They will all ingest training data, compute accurate models based on the input data, and finally use the computed model and statistics to generate a predicted output. These predictive algorithms have been experienced in various fields such as stock prediction based on the currencies, international news and many other contributing factors. Machine learning has also been extensively used on various social media platforms for facial recognition applications to tag and share photos of friends [23, 24]. A series of steps that are followed during a machine learning process are as follows [23, 25]:

- **Data acquisition and preprocessing**: Datasets are sensitised from outliers, missing data and necessary computations.
- **Feature selection and extraction**: Important features that are significant to the data are extracted and identified.

- **Model selection**: Depending on the task at hand, the correct machine learning model is carefully selected.
- **Validation**: The accuracy or performance of the machine is evaluated.

Machine learning is encapsulated within the AI context, which in general terms describes systems that can "think" [23]. Figure 10 below depicts the four subsets of AI, however, the focus of this research will be on machine learning.

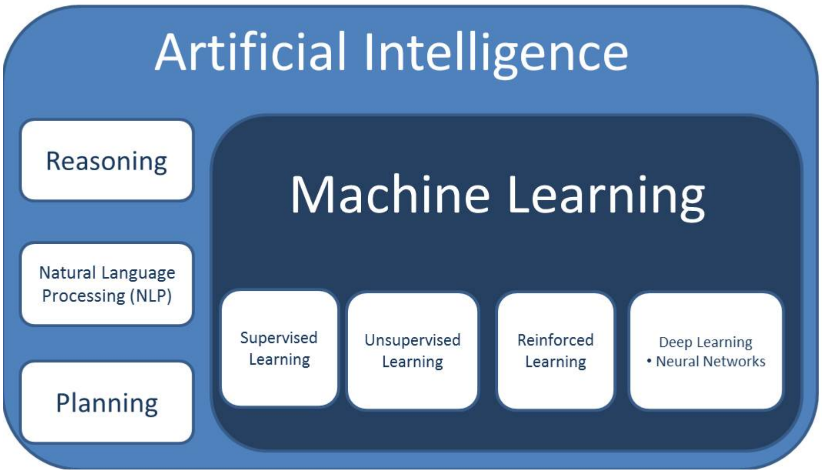

Figure 10: Artificial Intelligence categories [23]

The four categories that make up machine learning, i.e. supervised learning, unsupervised learning, reinforced learning and deep learning, will be discussed in order to provide reasonable insight in narrowing down the viable options for the machine learning algorithms that could be applicable in this research.

4.2 Supervised Learning

Supervised learning models, which form part of the predictive modelling algorithms, use a chosen set of data to predict a target variable.

The target variable represents the solution whereas the data used for prediction is the problem that any organisation is trying to solve [26]. The training data includes both the input and the desired result. In some cases the correct results(targets) are known and used as inputs during the training process. The models' primary purpose is to provide the correct result using given information and without prior knowledge of the result. Supervised learning can further be distinguished between models that predict a numeric variable (regression) or a categorical variable (classifiers) [25] as shown in Figure 11.

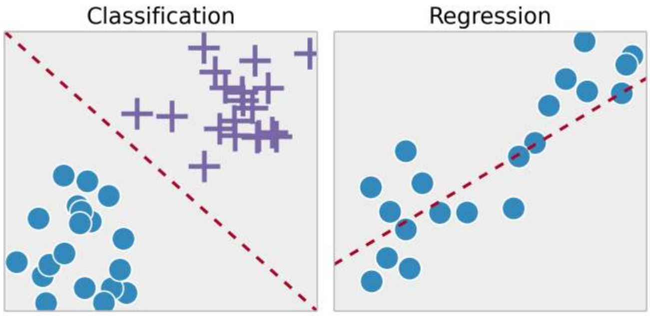

Figure 11: Example of Classification in (a) and Regression in (b) [27]

An example of supervised learning would be an algorithm which is provided with images of a lion labeled as cats and images of a python labeled as a snake. Following the training, the supervised learning algorithm should correctly identify the unlabeled image of a lion as a cat and an unlabeled image of a python as a snake [24]. In other words, Yousefian et al (2017) describes supervised learning as a system that learns to perform its assigned task with the help or supervision of a teacher. Some supervised machine learning algorithms include [28]:

- Support Vector Machine
- Decision Tree
- K Nearest Neighbour
- Linear Regression
- Naïve Bayes

For the purpose of this research, the detailed working of each of those algorithms will not be discussed, however, only the relevant ones that stand out from the power systems applications literature review will form part of the focus.

4.3 Unsupervised Learning

Unsupervised learning deals with unlabeled data and gives the learning algorithm responsibility in its own accord to establish commonalities or common relationships within the input dataset. The main purpose of unsupervised learning is to discover hidden patterns within the given dataset. It also has the ability to feature learn where the required classification of raw data representation can automatically be found by the computational machine [24, 29].

A common practical example of unsupervised learning is in the customer transactional data. A large amount of customer purchasing data is made available, but the challenge is making sense of that data in terms of customer attributes. However, through the use of unsupervised learning algorithm, it may be determined that men of a certain age range who purchase a certain energy or health product most likely go to gym often and therefore the marketing campaigns can be targeted to this audience in order to increase sales. The machine can accept seemingly unrelated and complex data and arrange it in a meaningful way without being trained on the "correct" answer [24].

- K-Means
- Hidden Markov Models
- Hierarchical Clustering

4.4 Reinforced Learning

Reinforcement learning is a challenge of an agent that must produce a certain output in a form of an action or a sequence of actions from learned behavior through an iterative process with a dynamic environment. It employs an occasional scalar reward system as the only supervisory signal. This learning consists of:

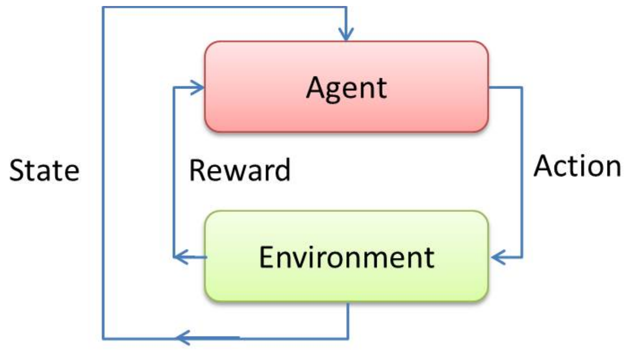

Figure 12: Reinforced Learning action and reward system

- an agent which interacts with a set of environments.
- a set of actions and several rules that govern the interaction between the states.
- the reward and what is finally observed by the agent as shown in Figure 12 [30, 31]

In the basic reinforcement learning algorithm, an agent interacts with its environment through the action, as depicted in Figure 12. At each iteration executed, the agent receives an input. The input is an indication of the current state of the environment. The agent then selects an action to generate as an output. The action alters the state of the environment, and this new state transition is communicated to the agent through a scalar reward signal. The aim is that the agent's behavior should select the actions that best increases the rewards signal. This is learned through a systematic trial and error which is controlled by a wide variety of algorithms [31].

4.5 Deep Learning

Deep learning methodology uses computational models that comprise multiple processing layers to learn how the data is represented in several levels of abstraction [32]. Layers of these models reveal discriminative/descriptive information from the raw input data. It functions on the backpropagation algorithm to determine how a machine should alter its parameters to compute each layer using representation from the previous layer [32]. Examples of deep learning algorithms include multi-layer perceptrons, deep neural networks, convolutional neural networks, deep belief nets and auto encoders. Figure 13 [29] shows plateauing performance of the old machine learning as data volumes increase. Deep learning, however, shows increasing performance with large amounts of data.

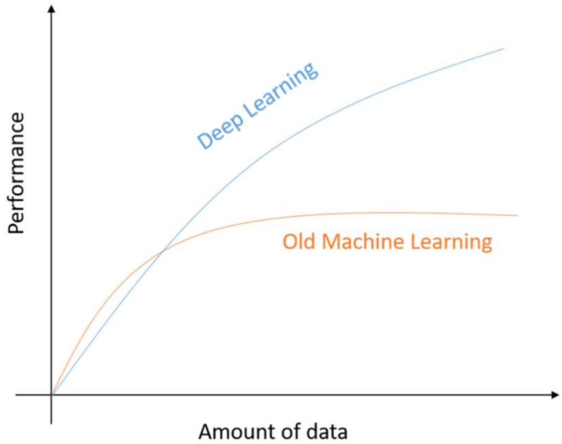

Figure 13: Performance of deep learning relative to the volume of data, derived from [29]

As the data increases in velocity, volume, veracity as well as variety, computing capacity also needs to be scaled as such which is where the difficulty lies with this kind of algorithm [29]. Speech recognition as well as computer vision have seen great developments from deep learning approaches [24].

4.6 Machine Learning Application in Power Systems

Prediction of the power system behaviour has gained some attention in the past few years, especially with PMU data availability. The techniques used in the prediction of anomalies on the network vary from the traditional signal processing methods which include spectral Fast Fourier Transform (FFT) [33], to more advanced techniques such as Autoregressive Moving Average (ARMA) [34], and Artificial Neural Network (ANN) [35].

De Callafon and Wells statistically determine a threshold that defines a stable or an unstable event by automatically adjusting the variance from the measured PMU data. Using ARMA to obtain ambient oscillations, the signal is then appropriately labeled as stable or unstable if it falls outside of this threshold. In this way, an event can be dynamically characterised and further removes possible outliers [34]. Dynamic predictions using PMU data from 17 voltage buses were used by Bhatt et al in a Multivariate Autoregressive (MAR) [36].

Comparison between the measured data and simulated data in this prediction method revealed that the prediction with measured data was only accurate for the first few seconds before a significant deviation is observed. This method also has limitations in data from a few PMUs and precise prediction of the continually changing network perturbations [36].

One Class Support Vector Machine (OCSVM) is used in the telecommunication field to detect anomalies in the signals. OCSVM creates detectors that can recognise a data profile. The first step involves data preprocessing of the training data and the test data where a vector set is extracted. This vector set is then used to train the detector and an anomaly alarm is activated when a negative value is returned. A positive value is labeled as normal. Unseen events are effectively detected using this method [37].

Prediction of eigenvalue for small signal stability assessment was investigated by Teeuwsen et al [35]. The method presented makes use of an observation area on the real-imaginary axis where low-frequency inter-area eigenvalues are positioned. In order to generate training data for the Neural Networks (NN), the observation area is sampled, and the distance between the sample points and the actual eigenvalues is computed.

The sample point is activated if this distance is small, which becomes the NN training data. More NN training data is generated through the 2966 load-flow patterns of the network. After the training, the activations are transformed into regions on the real-imaginary axis where the eigenvalues are located. The machine aims to predict these regions. The prediction method success criteria are based on selecting the scaling parameter relating to the acceptable region and it affects both the training and the testing process [23].

In order to perform eigenvalue analysis efficiently for the Taiwan power system, Chen et al. developed an ANN using a multi-layer feedforward method [38]. The multi-layer feedforward was trained using 120 training patterns and 80 test patterns which achieved a result of 2.22% error. This means that the worst electromechanical modes can be identified and the stability of the system can be accurately established. It is essential to note the use of NN for this eigenvalue area prediction in the studies above. Furthermore, time series PMU data will be used instead of loadflow data for this research. The next sections will look at NN and time series prediction methods.

Time series analysis and prediction have been studied for 40 years [39]. Basically, the aim of a time series prediction $y_1, y_2, ...$ is to estimate the value at time i based on its previous data y_{i-1}, y_{i-2}, ... Assuming $\mathbf{x} = \{y_{i-k}, y_{i-k+1}, ..., y_{i-1}\}$, $i = \{k, ..., n\}$, the goal is to find a function $f(x)$ so that $\hat{y_i} = f(x)$ is as close to y_i as possible.

The Autoregressive Integrated Moving Average (ARIMA) model has been used to study time-varying signals. The one limitation presented by ARIMA, is its natural tendency to concentrate on the mean values of the past series data. Therefore, it is difficult to capture rapidly changing processes [40].

Support Vector Regression (SVR) has been applied successfully for time series prediction, however, it experienced disadvantages, like the lack of structured means to determine some important parameters of the model [40].

Recurrent Neural Networks (RNN), which is one of the deep learning models, manages to cope with with time series by recurrent neural connections. For a standard RNN architecture however, the impact of a given input on the hidden layers and eventually on the neural network output would either decay or blow up exponentially when cycling around recurrent connections [41]. To address this problem, Long Short Term Memory (LSTM) has been comprehensively designed by changing the structure of the hidden neurons in traditional RNN [42].

4.7 Neural Networks

The neural network architecture is comprised of a network of neurons or nodes. Edges are lines which connect the nodes. Information is transferred as input to the nodes in the network via the edges. Input information is further multiplied by the weight determined by the constructor of the network. The weight is used as a scaling factor to adjust the importance of different computational results from the given node. Activation function, typically in the form of $tanh$, is added to the result and its purpose is to regulate the weight contribution from the neurons to the output result. Typically, nodes are aggregated into layers. The final output of the nodes of the neural network accomplish the task, such as recognising text or an image [43].

4.7.1 Recurrent Neural Networks

In a similar way that human beings can predict the next word in a sentence due to the recognition of context, so are RNN in nature. RNN forms part of a class of neural networks that have memory [44]. Due to to this memory facility, RNNs are explicitly designed for processing sequential data and hence can be applied to time-series data [45].

4.7.2 LSTM

A LSTM network is a type of RNN that can consider long-term dependencies. It differs from RNN in that it can remember information for a more extended period of time and can further avoid long term dependencies. The internal workings of an LSTM algorithm include gates and layers of neural networks similar to that of RNN. A cell state propagates through the entire LSTM. Additionally, the value is changed by gates that allow the data to be added or removed from the cell state. Other gates permit information from previous LSTM outputs to be stored, and this is the memory function of LSTM's [46].

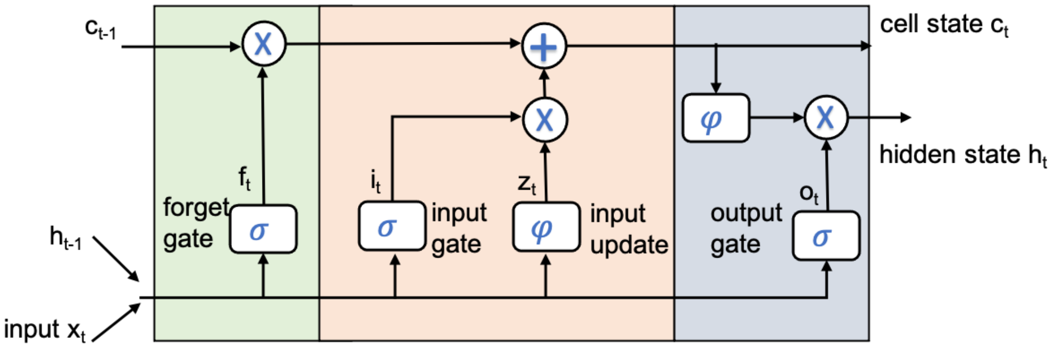

Figure 14: Block diagram showing LSTM Memory block derived from [41]

The sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

and the *tanh* function:

$$\varphi(x) = 2\sigma(2x) - 1 \tag{16}$$

are generally used as activation functions in ANNs. The sigmoid function returns a value between 0 and 1 while the *tanh* function returns values between -1 and 1. The detail behind the LSTM memory block can be interpreted in three steps as shown in Figure 14.

In the first step, the sigmoid function implements a decision on what information is to be removed from the memory cell (i.e. forget gate). Information from the previous hidden state and the current input state is put through the sigmoid function. The returned value is between 0 and 1. If the returned value is closer to 0, it will be forgotten and if the returned value is closer to 1, it will be retained in memory [41, 46].

The second step (input gate) aims to decide what new information will be stored in the memory cell state. The second step involves passing the previous hidden state and the current input state to the sigmoid function where 0 is not important and 1 is important. The hidden state and current input state are passed through the *tanh*, which aims to regulate the network between -1 and 1.

The sigmoid output (i_t) and the *tanh* output (z_t) are then multiplied and the sigmoid output will decide which information from the *tanh* to keep. The cell state can now be calculated. The cell state is initially multiplied by the forget vector, potentially discarding some values in the cell state if multiplied by values closer to zero. The cell state is then updated by adding the forget gate's output to that of the input gate. The cell state, c_t, will be the new cell state [41, 46].

The final step is the output gate which decides what the new hidden state should be. The hidden state is used for prediction and also has information about the previous inputs. The last hidden state and the current input state are passed through the sigmoid function and the output of this is o_i. The newly modified cell state is also passed through the *tanh* function. These two are then multiplied to decide what information the hidden state should carry. The output is the hidden state. The new hidden state (h_t) and the new cell state (c_t) are then carried through to the next time stamp. Through the cooperation between the memory cell and the gates, LSTM is empowered to predict time series with long-term dependencies [41, 46].

4.8 Conclusion

This Chapter introduced the machine learning theory which includes supervised and unsupervised learning. It further investigated the various machine learning algorithms from the literature that have been previously applied in power systems. A more-in-depth view of neural networks with a particular interest in RNN, ARMA, ARIMA, SVR and LSTM was also provided. The literature survey revealed LSTM to be most favourable in time-series prediction and will henceforth be applied to this study with PMU data. The long term dependencies of the LSTM model make it ideal for time-series prediction.

The following Chapter analyses small-signal power system stability using real measurements from PMU devices.

5 Small-Signal Power System Stability Analysis using synchrophasor measurements

Analysis of power system stability, notably, small-signal stability, is performed in this section. This analysis is carried out through the use of PMU data and eigenvalue estimation methods.

Measurement-based analysis of the power system utilises real-time measurement data to analyse the power system's condition in terms of stability. It uses synchronised PMU data from the strategically located PMU devices across the network. This method of analysis varies from the model-based analysis because it provides a more accurate observation as well as prediction of power system stability [47].

The main reason for using wide-area measurement is to accurately determine the system's oscillation and damping characteristics during an event or a disturbance. As mentioned before, small-signal disturbance contains a low-frequency element and damping. This phenomenon occurs when one or a group of generators swing against another group of generators, respectively. The undamped oscillation can cause a power system blackout [47].

There are a few methods that can be employed in determining the oscillation property. These include oscillation detection and mode metering. Mode metering can estimate damping of the oscillation and provide early detection of the mode oscillation [47].

This Chapter investigates the various mode metering methods used to perform eigenvalue extraction from the measured PMU data.

5.1 Background

Phasor Measurement-based methods of analysis do not rely on complex mathematical models which are generally developed from the knowledge of the system in advance. If the system outputs and/or inputs are measurable, models that best estimate the system dynamic characteristics can be approximated.

A 'black box' approach where the detail of the box (the system) is unknown, and the outputs and inputs are known is presented in Figure 15. The inputs and outputs are extracted from the PMU measurements.

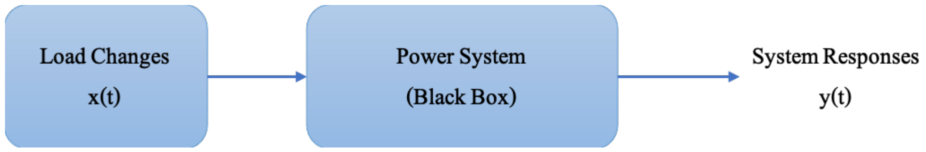

Figure 15: Measurement-Based Analysis - the black box approach

Assuming that there is no knowledge of the power system details, it will be regarded as a 'black box'. It is known, however, that the power system is continually subjected to small disturbances in the form of load changes on the network. It would mean that the only information that will be able to determine the dynamic characteristic of the system will come from the inputs and output measurements. These disturbances are a random phenomenon and can neither be predicted nor measured.

5.2 Analysis Methods

In order to analyse the power system stability, especially small-signal stability, a method commonly known as eigenvalue analysis is used to determine the eigenvalues for power systems based on the model characteristics. This method involves the linearisation of the model formed by the differential and algebraic equations around only one equilibrium point. A model based analysis will not provide a real-time stability analysis because the operating point of the power system is not static. Synchrophasor measurement from PMUs is therefore critical in capturing the instantaneous condition of the power system. Due to its low resolution, the traditional SCADA system cannot detect low-frequency oscillation. PMUs also provide an added advantage with their time-stamped data to view the entire interconnected power system. The following eigenvalue analysis methods will be explored in detail: Mode Estimation, Matrix-Pencil (MP) method, and Prony Analysis. All these methods can extract the low-frequency eigenvalues from the measured PMU data [47]. For the purpose of the analysis of the different techniques, the following voltage signal will be used.

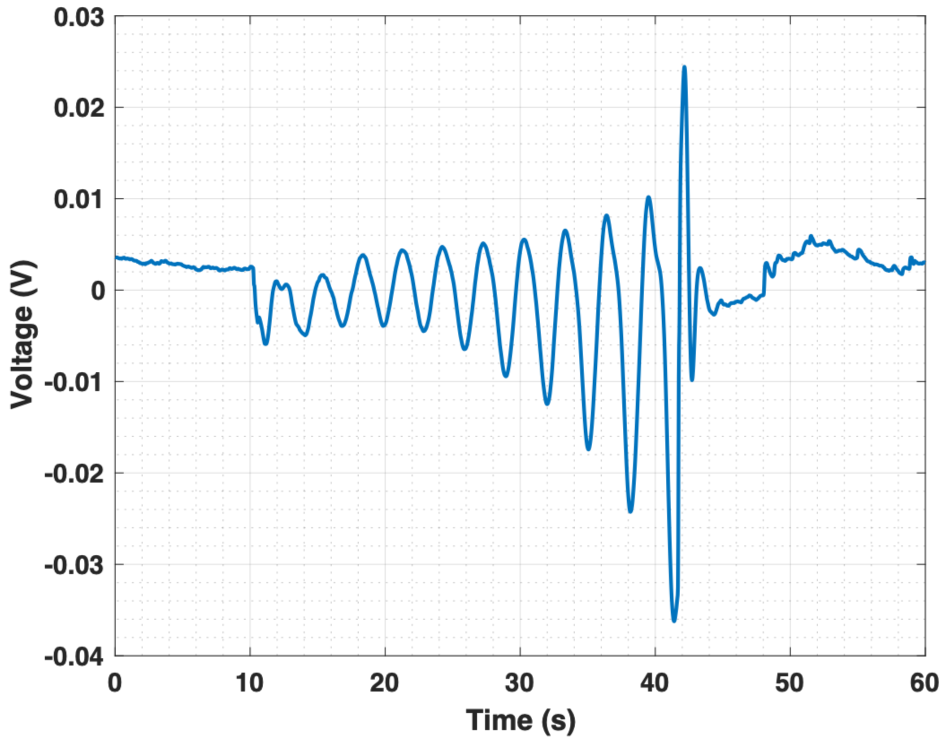

Figure 16: A detrended Voltage Signal, measured from a PMU, that demonstrates negative damping or growing oscillations

Figure 16 represents a negatively damped oscillation measured from one of the PMUs on the network. The PMU selected for this study is located at the Northernmost part of the network and forms part of the interconnection link between two utilities. Therefore, this particular PMU is perfectly placed to be able to monitor the inter-area oscillatory behaviour of the network. The measured voltage data was detrended to pre-process the data for analysis. The eigenvalue estimation methods will be discussed in the following sections.

5.2.1 Mode Estimation Method

The mode estimation method uses the FFT computation which returns the discrete Fourier transform. The discrete Fourier transform is used to find the most substantial frequency component within a given data set or measurement window. The Fourier spectra contains information on the frequency (ω) and damping (d) of the observed mode. When these parameters are established from estimation, the eigenvalue of the oscillation is given by:

$$\lambda = -d \pm j\omega\sqrt{1 - \left(\frac{d}{\omega}\right)^2} \approx -d \pm j\omega \tag{17}$$

where d and ω are the damping constant and undamped natural angular frequency of the oscillation. These parameters (d and ω) are estimates from the Fourier spectra. A curve-fitting method is used to determine the amplitude as well as the above-mentioned parameters from:

$$g(x) = \frac{A}{\sqrt{(x^2 - \omega^2)^2 - 4d^2x^2}} \tag{18}$$

such that the error difference between the estimated signal and the original signal is minimised. The estimation method that was implemented considers the voltage signal (Power or current could have also been used) from the PMU data during an event. The signal is first detrended to remove the direct current (dc) offset. Because the frequencies of interest (inter-area mode) are around $0.3Hz$, a second-order butterworth bandpass filter is then applied on the detrended signal to filter out the frequencies between 0.01 and $0.4Hz$. The FFT spectrum signal is represented in the Figure 17. It can also be observed that the frequencies larger than 0.3Hz are present in the FFT signal, which needs to be considered when performing the curve fitting.

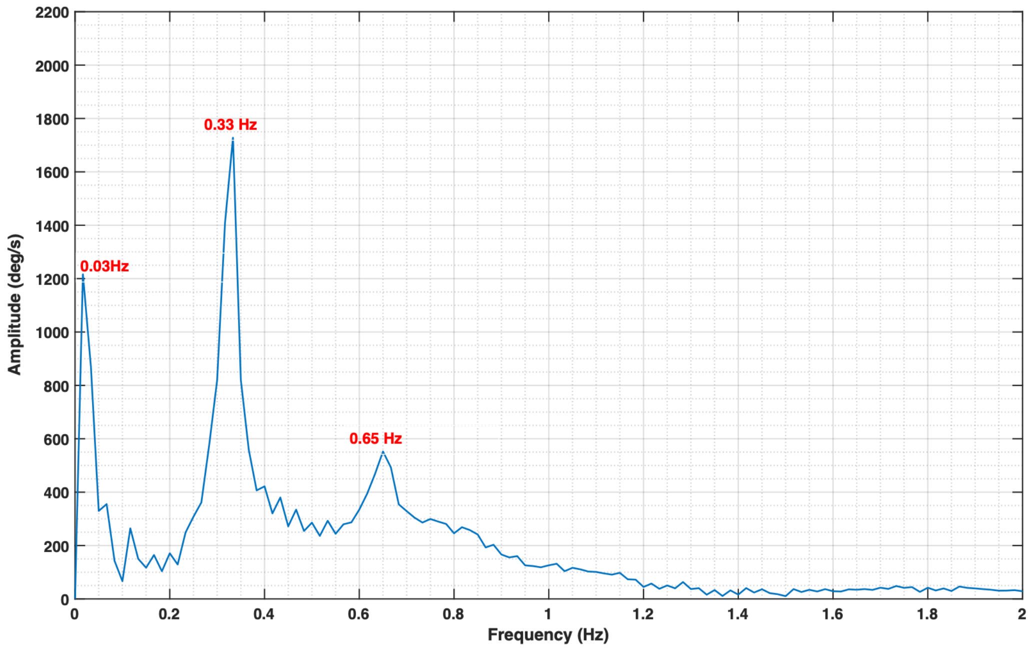

Figure 17: FFT Spectra of the voltage signal showing the dominant frequencies present in the signal

The signal contains many frequencies as shown in Figure 17. Using the MATLAB R2020a curve fitting method, the eigenvalues can be estimated from applying equation 18. Figure (18) shows the FFT curve with a model fit on the original signal. The estimated parameters were as follows:

Table 1: FFT fit results

General Model	fitresult (x) $= \frac{A}{\sqrt{(x^2-\omega^2)^2-4d^2x^2}}$
Coefficients (with 95% confidence bounds)	
Amplitude	5.254e-05 (4.959e-05, 5.549e-05)
Damping	0.01956 (0.01779, 0.02133)
Frequency	0.3301 (0.3283, 0.3318)

The estimated parameters represent the eigenvalues of the system. As stated in the previous Chapter, the eigenvalues contain information about the frequency and damping of the oscillation. The estimation of the fit can be visualised in Figure 18.

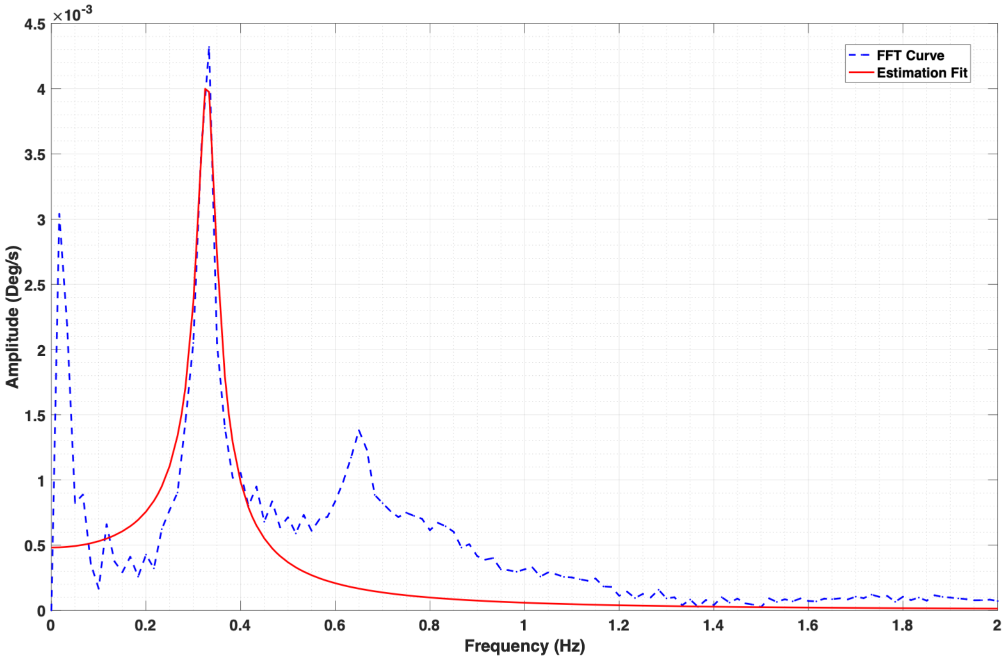

Figure 18: Modelled FFT curve and the estimated fit with frequency and amplitude parameters

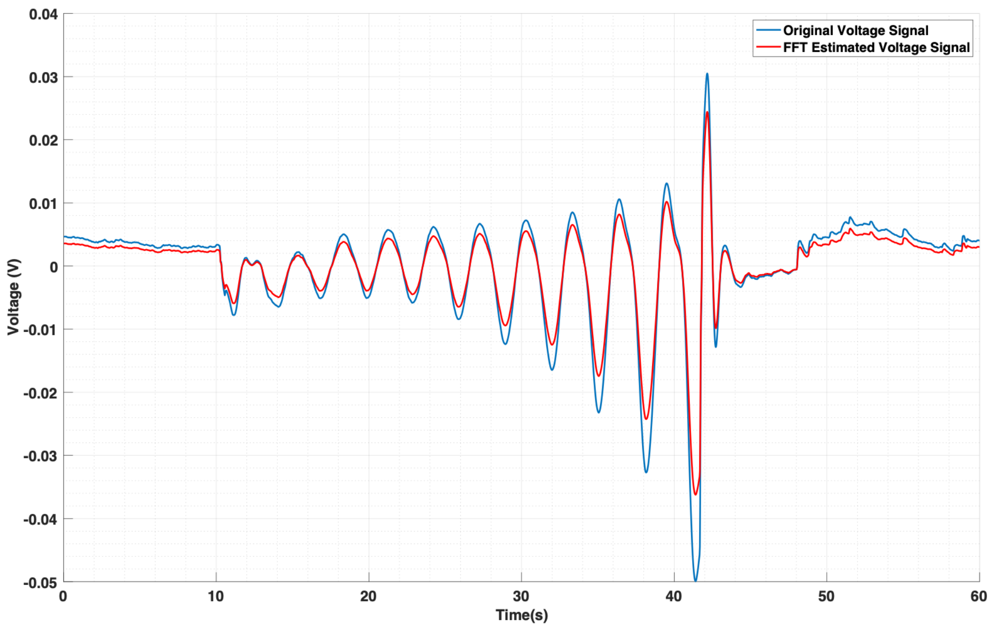

Figure 19: Reconstruction of the voltage signal using the FFT estimated model parameters

The FFT spectrum of the oscillation shows a clear peak corresponding to the inter-area mode. Its eigenvalue is estimated by a curve-fitting method. Figure 19 shows the reconstruction of the voltage signal from the FFT estimated model. It appears to closely estimate the original signal at low oscillation magnitudes but shows deviations as the oscillations increase in magnitude.

5.2.2 Prony Analysis

Gaspard de Prony developed the Prony method in 1795. It was used to interpret the expansion of gases as a linear sum of damped complex exponentials of uniformly sampled signals [48]. Prony Analysis is one of the signal processing method that extends Fourier analysis. The Prony method approximates equally-spaced samples of data to a linear combination of complex exponential functions with different characteristics. The classical Prony method solves a set of linear equations and converts a non-linear approximation of exponential sums [48].

The Prony method can estimate damping, phase, frequency and magnitude information contained within a given signal through this conversion. It bears close relationship to least-squares linear prediction algorithm used for Autoregressive (AR) and ARMA parameter estimation.

Prony Analysis is a method estimating the parameters for the exponential terms by defining a fitting function to a signal in a basic form of (19) [49]. The four elements in (19) are the magnitude A_n, the damping factor σ_n, the frequency f_n, and the phase angle θ_n. Each exponential component with a different frequency is observed as a unique mode of the original signal $y(t)$. The time interval between each of the sampled data is T [48, 49, 50, 51].

$$\hat{y}(t) = \sum_{n=1}^{N} A_n \, e^{\sigma_n t} \cos(2\pi f_n t + \theta_n) \tag{19}$$

Using Euler's theorem when $t = K * T$ and

$$\cos\left(2\pi f_n + \theta_n\right) = \frac{e^{j((2\pi f_n + \theta_n)}e^{j\theta_n}}{2} + \frac{e^{-j((2\pi f_n + \theta_n)}e^{-j\theta_n}}{2} \tag{20}$$

equation (19) is reduced to be:

$$y[k] = \sum_{n=1}^{N} B_n z_n^k, k = 0, ..., N-1 \tag{21}$$

$$B_n = \frac{A_n}{2}\ e^{j\theta_n} \tag{22}$$

$$z_n = \ e^{(\sigma_n + j2\pi f_n)T} \tag{23}$$

The two coefficients: B_n and z_n contain the frequency and damping ratio which are important components in analysing the system's behavior. Prony method aims to obtain these two components such that (24) is satisfied i.e. the error between the estimated signal and the measures signal is minimised.

$$\hat{y}(t) = y(t) \tag{24}$$

This estimation is obtained in the following three steps:

- **Solve the linear prediction model which is constructed from the measured data or the observed data set.**

Firstly, the Linear Predication Model (LPM) of order N, shown in (24), is built to fit the equally sampled data record $y(t)$ with length M.The typical length of M should be at least three times larger than the order N.

$$y[k] = a_1 y[k-1] + a_2 y[k-2] + ... + a_n y[0] \tag{25}$$

Equation (25) can be written in matrix form $Y = Da$ as:

$$\begin{bmatrix} y[n] \\ y[n+1] \\ .. \\ . \\ y[N-1] \end{bmatrix} = \begin{bmatrix} y[N-1] & y[N-2] & .. & .. & y[0] \\ y[N] & y[N-1] & .. & .. & y[1] \\ y[N+1] & y[N] & .. & .. & y[2] \\ .. & .. & .. & .. & .. \\ y[N-2] & y[N-3] & .. & .. & y[N-n-1] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ .. \\ a_n \end{bmatrix}$$

where

$$Y = \begin{bmatrix} y[n] \\ y[n+1] \\ .. \\ . \\ y[N-1] \end{bmatrix}, D = \begin{bmatrix} y[N-1] & y[N-2] & .. & .. & y[0] \\ y[N] & y[N-1] & .. & .. & y[1] \\ y[N+1] & y[N] & .. & .. & y[2] \\ .. & .. & .. & .. & .. \\ y[N-2] & y[N-3] & .. & .. & y[N-n-1] \end{bmatrix}, a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ .. \\ a_n \end{bmatrix}$$

The coefficients a_i from the above equation can be calculated using the least squares techniques [52].

- **Find roots of characteristic polynomial formed from the linear prediction coefficients.**

The characteristic (26) can then be determined from the calculated a_i coefficients which is of the form:

$$z^n - (a_1 z^{n-1} + a_2 z^{n-2} + ... + a_n z^0) = 0 \tag{26}$$

The eigenvalues of the system are obtained from calculating the roots of the characteristic equation above using (27) and (28):

$$\sigma_k = \frac{\ln |z_k|}{T_k} \tag{27}$$

$$f_k = \frac{tan^{-1} \left[\frac{Imag(Z_k)}{Real(Z_k)} \right]}{2\pi T_k} \tag{28}$$

- **Solve the original set of linear equations to yield the estimates of the exponential amplitude and sinusoidal phase.**

The residue, B_i, which is associated with each relevant eigenvalue can be calculated using the least squares method. This is achieved by constructing the equation

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ .. \\ y(N-1) \end{bmatrix} = \begin{bmatrix} z_1^0 & z_2^0 & .. & .. & z_n^0 \\ z_2^1 & z_2^1 & .. & .. & z_n^1 \\ z_1^2 & z_2^2 & .. & .. & z_n^2 \\ .. & .. & .. & .. & .. \\ z_1^{N-1} & z_2^{N-1} & .. & .. & z_n^{N-1} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ .. \\ B_n \end{bmatrix}$$

or $Y = ZB$

where

$$Y = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ .. \\ y(N-1) \end{bmatrix}, Z = \begin{bmatrix} z_1^0 & z_2^0 & .. & .. & z_n^0 \\ z_2^1 & z_2^1 & .. & .. & z_n^1 \\ z_1^2 & z_2^2 & .. & .. & z_n^2 \\ .. & .. & .. & .. & .. \\ z_1^{N-1} & z_2^{N-1} & .. & .. & z_n^{N-1} \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ .. \\ B_n \end{bmatrix}$$

Using MATLAB, the original linear prediction coefficients B_i can be computed by solving the over-determined set of equations above. As B and z are now known, the amplitude, frequency, phase and damping coefficients are computed using (22) and (23).

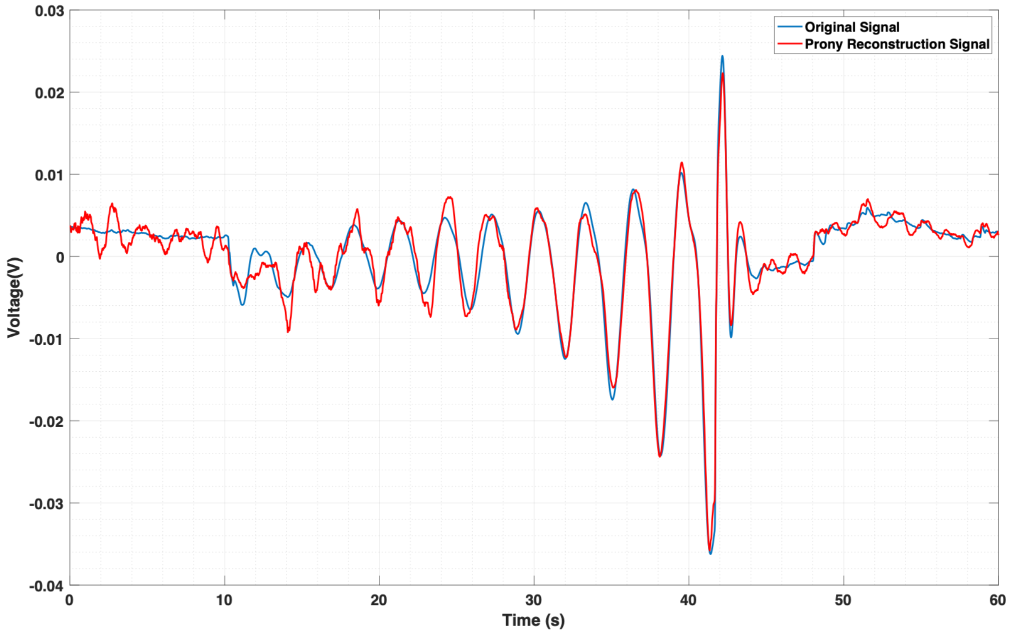

Figure 20: Prony Analysis Simulation with the original voltage signal and the reconstructed estimate at N = 3000 and p = 1400

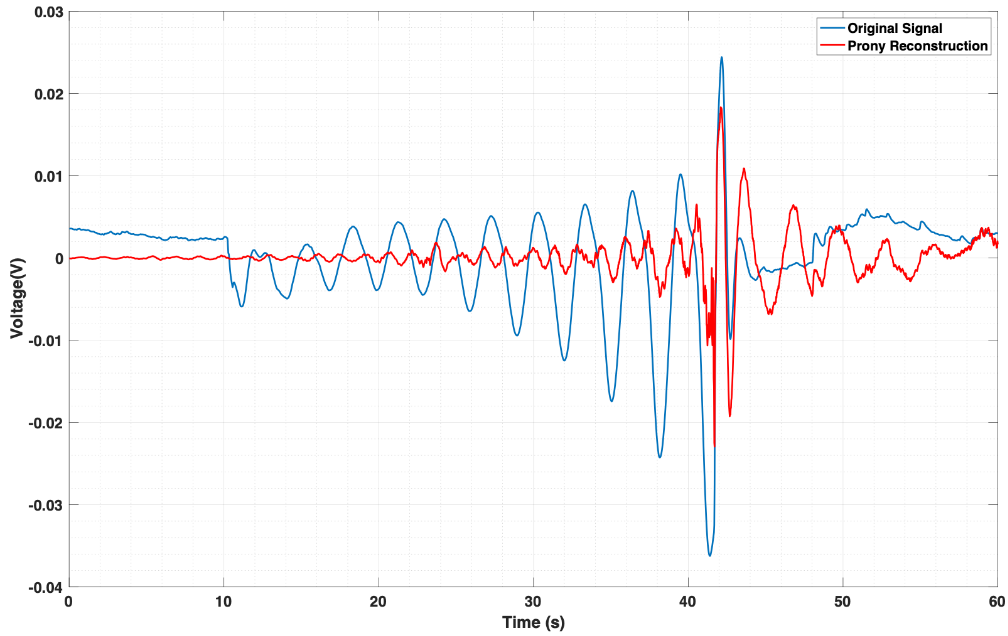

Figure 21: Prony Analysis Simulation with the original voltage signal and the reconstructed estimate at N = 3000 and p = 1450

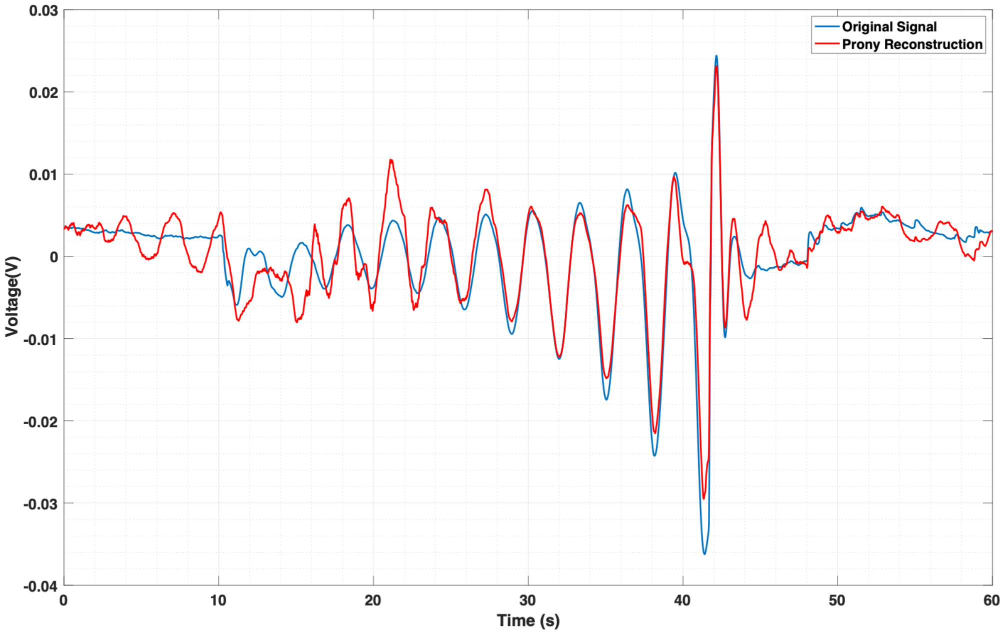

Figure 22: Prony Analysis Simulation with the original voltage signal and the reconstructed estimate at N = 3000 and p = 1300

Figure 20 shows the Prony Analysis against the measured PMU signal. The three steps mentioned above-provided information necessary to understand the power system stability. It is important to note how the accuracy of the estimation varies with the varying p parameter changes. Javad Khazaei et al. state that the D matrix dimension should be $N - n + 1$, n. It is further stated that if $N > 2p$, the result will be an over-determined linear equation and will best be solved by the Least Square Estimation (LSE) [49]. Linear equations will be under-determined in a case of $N < 2p$ and will result in more than one solution of a. A unique solution which is the best fit is derived from a square D matrix, and hence p is selected to be close to $N/2$ [49].

In this case, when $N > 2p$, the approximation is more accurate. The closest approximation was realised at $p = 1400$ for a given number of samples $N = 3000$. Figure 21 and Figure 22 show how the approximation varies with different p values.

The drawback with Prony Analysis is that it can be numerically unstable due to the steps that comprise the algorithm: solving an ill-conditioned matrix equation and finding the roots of a polynomial. With high exponentials, the sensitivity to the roots of the characteristic polynomial to disturbances of their coefficient will also be high [48]. Furthermore, the Prony method's accuracy is impacted by noise, distinct components and slowly-changing components added in real-time measurements [53].

5.2.3 Matrix-Pencil Method

The Matrix-Pencil (MP) method is derived from the pencil-of-functions approach of estimation. The fundamental variations between the pencil-of-functions and the MP is that although they both apply a similar philosophy, MP is more computationally efficient than the pencil-of-function method [54]. The MP method fits a sum of damped sinusoids to evenly sampled PMU data. The damped sinusoid parameters of interest, amplitude, phase angle, frequency, and damping are estimated to fit the measured PMU data. The estimated sum of damped sinusoids to be fitted to a set of sampled data can be represented by [55]:

$$y[k] = \sum_{n=i}^{n} R_i z_i^k \tag{29}$$

The discrete signal $y(k)$ above is represented in (29) where $y(k)$ is equal to the sum of the product of the residues or complex amplitudes R and the poles z. The parameter n is the number of sinusoids or modes to be estimated, $k = 0, 1, ...N-1$ (N is the number of sampled data and should be greater than $2n$). The damping and frequency estimates of the signal can be extracted from each z_i, as shown in (31) [55, 56].

$$\begin{aligned} z_i &= exp(\lambda_i \Delta t) \\ \lambda_i &= \sigma_i \pm j\omega_i \end{aligned} \tag{30}$$

The matrix $[Y]$ is generated from the noisy PMU data $y(k)$ of the form:

$$[Y] = \begin{bmatrix} y(0) & y(1) & .. & y(L) \\ y(1) & y(2) & .. & y(L+1) \\ .. & .. & .. & \\ y(N-L-1) & y(N-L) & .. & y(N-1) \end{bmatrix} \tag{31}$$

L is the pencil parameter that serves as a noise filter and is typically between $N/3$ and $N/2$ [55, 56]. A Singular Value Decomposition (SVD) is then applied to $[Y]$ as shown in (32):

$$[Y] = U\Sigma V^H \tag{32}$$

The matrices U and V are unitary matrices, and the operator H is the Hermitian transpose or the conjugate transpose. The singular values of $[Y]$ are located along the diagonal of matrix Σ in descending order. A significant parameter n is selected such that the singular values in Σ beyond M can be approximated to zero. Calculating the ratio between the maximum singular value and all the others is a way to estimate M [55, 56].

$$\frac{\sigma_n}{\sigma_{max}} = 10^{-p} \tag{33}$$

where p is the number of significant decimal digits in the data. It (p) in (33) determines the number of estimated complex exponentials (i.e., a lower value leads to a higher number of estimated complex exponentials) [55, 56]. A matrix $[V']$ of a reduced form is then constructed using the rows corresponding to the n more significant singular values as

$$\begin{aligned} [V'] &= [v_1, v_2, ..., v_n] \\ [Y_1] &= U\Sigma'[V_1']^H \\ [Y_2] &= U\Sigma'[V_2']^H \end{aligned} \tag{34}$$

The $[V']$ are the column vectors of V, and matrix Σ' is the first n columns of Σ, the matrices $[V_1']$ and $[V_2']$ represent $[V]$ with the final column and final row removed respectively. The matrices from (34) can create the following Matrix-Pencil definition for noisy data.

Using the MP method as defined, the matrices from (5) can create the following new matrix-pencil definition for noisy data.

$$[Y_1]^+[Y_1] = V_2'^H [V_1'^H]^T \tag{35}$$

To solve for R from (29) the eigenvalues for $V_2'^H [V_1'^H]^T$ are determined using

$$\begin{bmatrix} y(0) \\ y(1) \\ .. \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & .. & 1 \\ z_1 & z_2 & .. & z_n \\ .. & .. & .. & .. \\ z_1^{N-1} & z_2^{N-1} & .. & z_n^{N-1} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ .. \\ R_n \end{bmatrix}$$

The estimated values for the damping and frequency are derived from z, shown in (29), and the amplitude is derived from R [55, 56].

The MP method was applied on the voltage data of 60 seconds window length measured from a PMU located at the Northern-most part of the network. Figure 23 shows the original measured signal as well as the MP estimated signal.

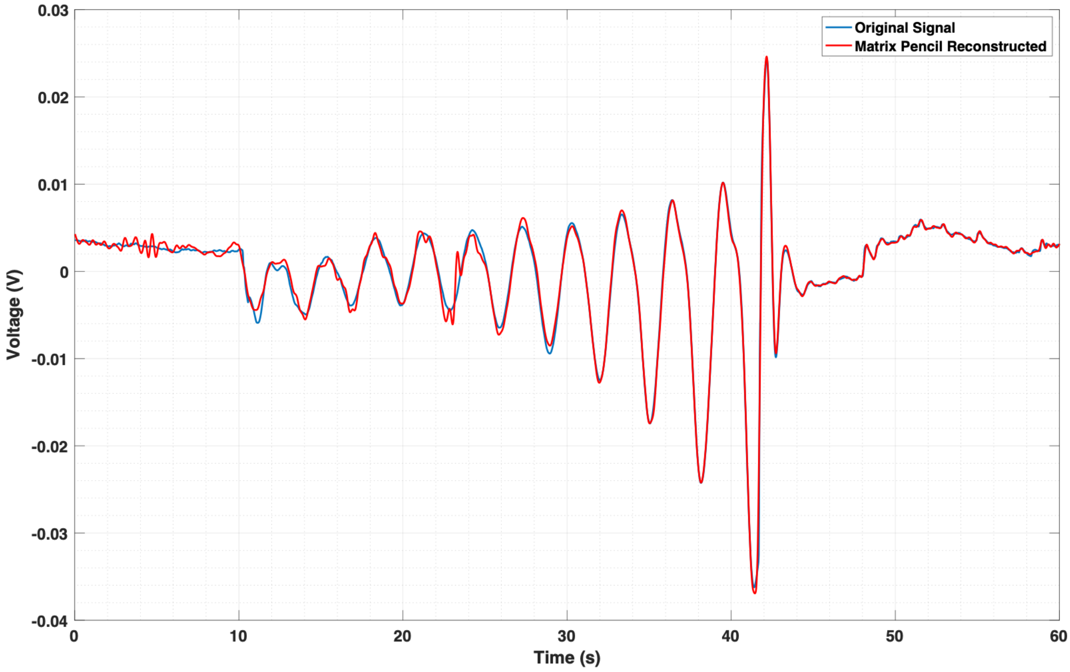

Figure 23: Matrix-Pencil Method estimation compared with the original signal during a disturbance period

The MP method bares similarities to the Prony method although it aims to solve the eigenvalue problem rather than follow the traditional three-step Prony method. MP has also been found to be less sensitive to noise as compared to the Prony method [48].

Table 2: Summary of Estimated Parameters from three different estimation methods

	Damping	Frequency (Hz)	Amplitude
FFT	0.01956	0.3301	5.5e-5
Prony Analysis	0.0549	0.3371	5.1e-4
Matrix Pencil Method	0.0475	0.3308	8.2e-4

The inter-area mode was estimated using all three methods and the results are shown in Table 2. The studied methods show a very close correlation in estimating the eigenvalue parameters. The accuracy of or limitations of each was discussed above. For the study, MP method will be used to generate the eigenvalues from the disturbances which will be discussed as it showed better estimation compared to the other two methods.

5.3 Conclusion

This Chapter discussed the three methods that can be used to perform eigenvalue extraction from a signal. The Estimation methods assume the power system to be a 'black box' with no knowledge of the network's intricacies but only relying on PMU data. The mode estimation method, the Prony Analysis method and the Matrix-Pencil method were explored. The mode estimation method uses FFT and a curve fitting method and estimates the eigenvalue parameters. The Matrix-Pencil method shares similarities with the Prony method, although it solves the eigenvalue problem rather than the three step Prony method. The advantage of the Matrix Pencil over Prony is that it is less sensitive to noise and not numerically unstable like the Prony method. Matrix-Pencil will therefore be utilised in the later Chapters to extract eigenvalues to perform machine learning from the analysis.

The next Chapter brings together the machine learning, power systems and PMU data in evaluating the prediction of disturbances on the network.

6 Evaluation of Machine Learning on PMU Data

The concept of power system stability and the potential risks related to small-signal stability were introduced in the previous Chapters. PMUs, due to their high resolution measurement ability, are ideal to monitor these small-signal disturbances. Measurements from the PMU devices are used to estimate the eigenvalues, which describe characteristics of the small-signal disturbance. In this Chapter, a machine learning algorithm which makes use of the high resolution measured PMU data to predict small-signal disturbances is proposed. The focus is on the two key challenges, which are determining, based on preceding PMU data, (i) if a disturbance is likely to occur, and (ii) can the disturbance be accurately classified in terms of its eigenvalue position on the eigenvalue plane. The method detailing the prediction algorithm, pre-processing data methods and data window selection are discussed in this chapter. The PMU data in this study is high-resolution time-series data from five different locations on the power system network. The pre-processing of that data, which is an essential step of the machine learning application to achieve prediction results, is also outlined in this Chapter.

6.1 Data Collection

This study and the analysis thereof is based on real PMU data obtained as a Comma Separated Values (CSV) file from the respective PMU devices at a sampling rate of 20ms. There are over ten PMU devices installed across the integrated power system; however, for this study's purpose, only five of these were considered due to the data quality issues of the other PMUs during the periods of interest when disturbances occurred. The data challenges can be due to line outages, GPS antenna failures and sometimes PMU hardware failure. The large volume size of the data also necessitated the use of only five PMUs for this study. The five PMUs are situated at the locations shown in Figure 24, and as can be seen, they are spread vastly across the entire power system network.

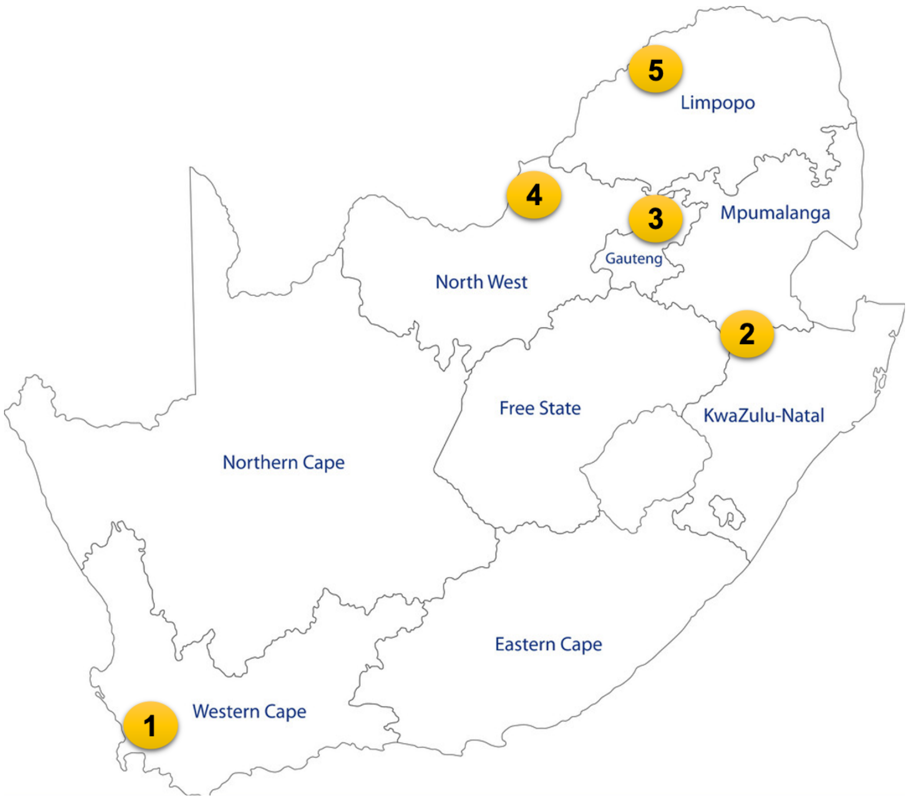

Figure 24: The five PMU Locations on the South African Power System network that are considered for this study

The PMU measures positive sequence voltage and current phasors (magnitude and angle) and frequency. As obtained from the PMU devices, the raw data is in the format, as shown in Table 3. Each PMU device can provide up to four measurements from feeder bays at a time as mentioned in Chapter 3.

Table 3: Data Format (CSV type) as reported by the PMU devices

Date (dd/mm/yy)	Time (s)	Freq (Hz)	df/dt (Hz/s)	V Mag (V)	V Angle (rad)	I Mag (A)	I Angle (rad)
01/12/19	0	49.998	-0.001	402092	136.72	849.47	147.86
01/12/19	0.02	49.998	-0.001	402082	136.73	849.06	147.87
01/12/19	0.04	49.912	-0.003	402067	136.71	849.69	147.85

Using the above-mentioned data, more parameters which will be used as variables can be computed using the following equations:

$$P = V \times I \cos(\theta_V - \theta_I) \tag{36}$$

$$Q = V \times I \sin(\theta_V - \theta_I) \tag{37}$$

where P represents the active power and the Q represents the reactive power. There had to be some careful consideration in the calculation of P and Q, particularly around the angles of voltage and current where the use of radians and the use of degrees had to be clearly understood and what the software tool uses as default. Appendix A contains the script used to perform the calculation for all the data collected from the five PMU devices.

6.2 Data Pre-processing

Before data can be analysised or subjected to some mathematical algorithms, it must first be normalised. Normalisation of the data is essential because, different features within the training and testing data will inherently exhibit different magnitudes and normalisation eliminates the prioritisation in the learning. For example, a variable that ranges between 0 and 1000 will outweigh a variable that ranges between 0 and 10. Transforming the data can prevent this problem. Normalisation involves the transformation of each of the features within the dataset. Data can be transformed between [0,1] or [-1,1], or normalised to have a zero mean and a unit variance. The latter is the most preferred method of normalising data according to literature [57]. Standardization, however is computed by subtracting the mean of each observation and dividing by the standard deviation as shown in (38).

$$z_j = \frac{x_j - \overline{x_j}}{\sigma_x} \tag{38}$$

where the standard deviation σ_x is given by (39):

$$\sigma_x = \sqrt{\frac{1}{p-1}\sum_{i=1}^{p}(x_i - \overline{x})^2} \tag{39}$$

and the mean is given by (40):

$$\overline{x} = \frac{1}{p}\sum_{i=1}^{p} x_i \tag{40}$$

Standardizing the features is an important step in the data preprocessing, more so when we compare measurements with different units: voltages, frequency, Active Power and Reactive Power. As shown in Table 3, the data entries correspond to samples that are obtained with a rate of 50 samples per second. That is to say that a 1000 data points characterise a 20-second event or data frame.

6.3 Prediction Method

The initial step in this study was first to determine if stability can be detected from historical data. That is to say, can an event or a disturbance be correctly classified through a machine learning method using data before the disturbance occurs. This is achieved by training the machine using a window of the PMU data before the event/disturbance occurs and labelling the subsequent disturbance as either being positively damped (1) or negatively damped (0) as shown in Figure 25 and Figure 26 respectively.

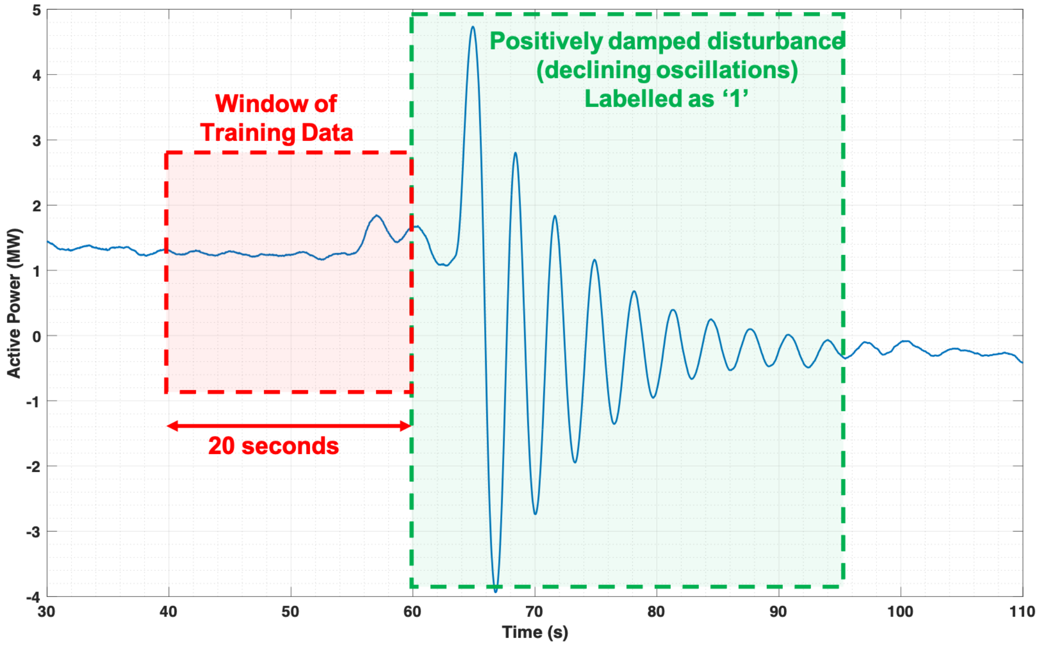

Figure 25: Positively damped active power signal where the 20 seconds training data window and disturbance label are demonstrated

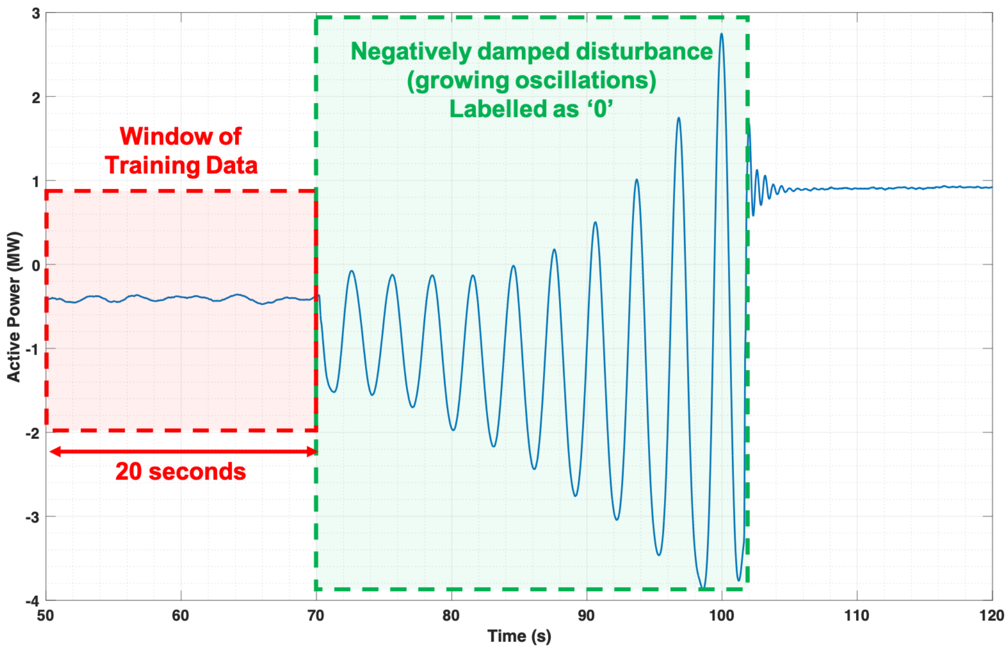

Figure 26: Negatively damped active power signal where the 20 seconds training data window and disturbance label are demonstrated

The window length is initially selected to have a duration of 20 seconds. The studied historical events' behaviour, positive or negatively damped, can generally be determined in 20 seconds as depicted in Appendix G. Furthermore, short window sizes are not ideal because the model will not acquire the longer dependencies hence neglecting important information. Too large a window size also has disadvantages in that it will then add a more considerable amount of redundant noise hence will overfit the training data [58]. The correct method in finding the most suitable window size is performing tests on different window sizes and finding the one with the least errors or mis-classification [59]. The 10-second window will be investigated to observe how the machine learning algorithm performs in half the initial window frame. The current SCADA system samples at 4 seconds time frame. The machine learning algorithm on the 4 seconds window frame, in alignment with the SCADA system, will be investigated. The last two windows which will be investigated will be the 1 second and the 500 ms windows to evaluate the shortest possible time frame that can be used for prediction of events.

The power system is continually subjected to small disturbances due to load changes and generator activity. These are not necessarily classified as faults on the system but can lead to severe system constraints and ultimately, a system blackout [1]. These can be classified as being either positively damped or negatively damped. A total of 24 events recorded by five PMU devices were then used in a machine learning algorithm. From these five PMU devices, a total of 34 features were used for training the machine. A further two features, voltage and frequency, from the PMU at the interconnection point were also added bringing the total number of features to 36. In the previous Chapter, some machine learning algorithms and their respective applications were discussed with reference to literary publications. Therefore, it was concluded that LSTM is the most appropriate algorithm that will be used in this time-series classification. Figure 27 describes the LSTM model.

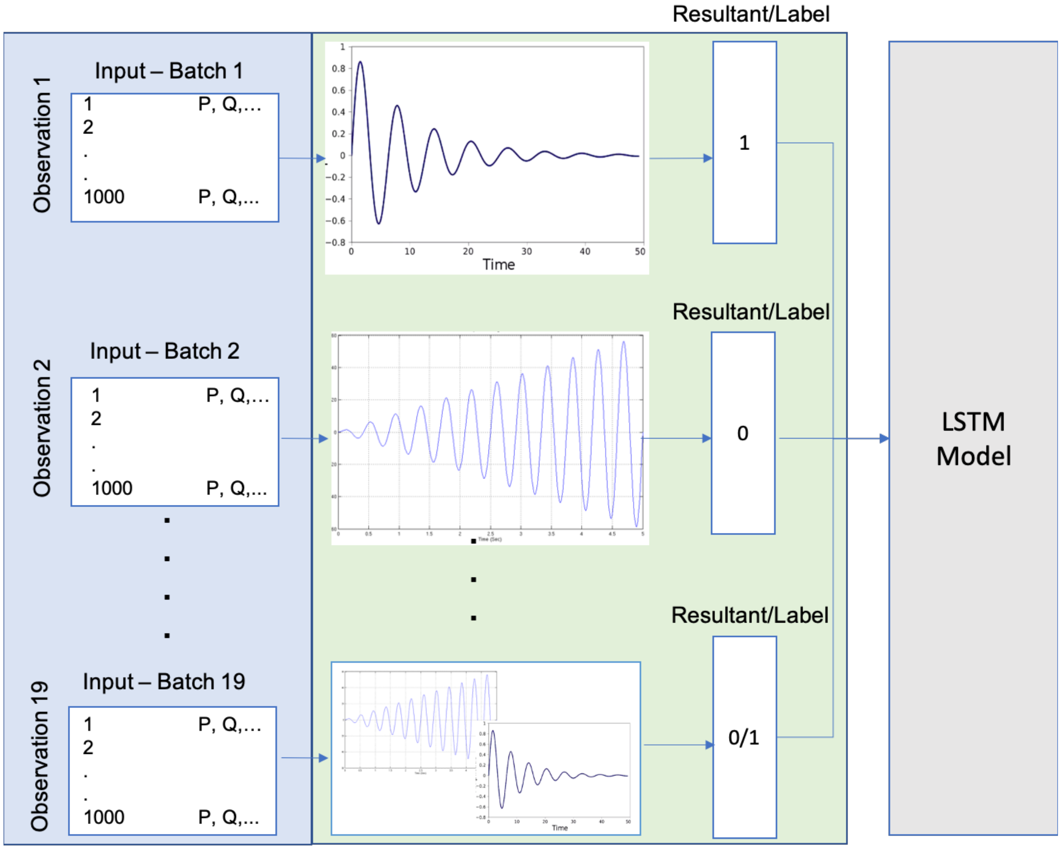

Figure 27: Observation and resultant preparation for the LSTM model for training

In the LSTM model setup, one batch is made up of 1000 data points, or in other words, a 20-second window. A disturbance with 36 features for training are represented in each batch. The batch also has a resultant, which is labeled as **1 for positive damping** and **0 for negative damping**. This labelling process will be carried through for all the 19 training disturbances.

In terms of LSTM, there are several important parameters that contribute to the LSTM's model performance.

- **Batch Size** - The batch size is selected to be the same size at the window length. In this case, since 20-second window length is being studied, this equates to 1000 data points. So batch size is 1000.

- **Epochs** - The number of epochs is a hyperparameter that determines the number of iterations the learning algorithm will propagate through the entire training dataset. It is an integer value and can vary in magnitude. The differences between a large epoch, a medium and a small epoch is that the curve can be over fitted, optimal or under fitted. In this study, the impact of different epoch sizes on the model accuracy was investigated. The epochs were set to be 5, 10, 20 and 50 for the analysis.
- **Hidden Layers** - The number of hidden layers and the number of LSTM cells are part of the considerations to be made when constructing an LSTM model. There is no correct or an incorrect solution to this and layers can range anywhere from 3 layers and two cells all the way to 4 layers and 1000 cells [60]. In this study, 100 hidden layers were selected, and these were kept constant for all the other studies.
- **Optimizer** - The LSTM model was built using Adam optimiser because it presents a high performance and fast convergence compared to other alternative optimisers [61], and it is generally used as a default recommendation. The optimizer Adam was set to the decay of 0.3.

6.4 Summary of the method

Figure 28 graphically shows the process flow of the training method. The details of this method are mentioned above. This method aims to train an LSTM model on pre-processed and correctly labeled PMU data to label unseen/new PMU data correctly.

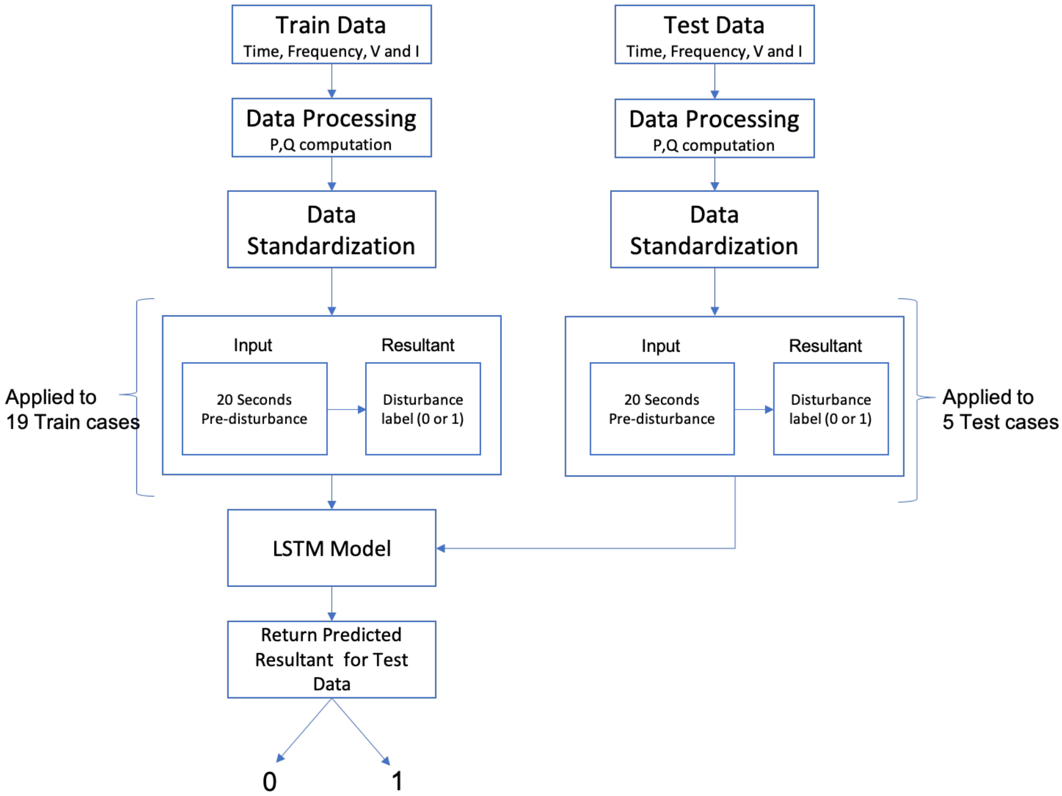

Figure 28: Summary of the data flow from data acquisition, pre-processing, training and testing the LSTM model

In this first instance of the training, the data only labelled either **0** for unstable (negatively damped) or **1** for stable (positively damped). The idea behind this initial step was to prove that indeed a disturbance can be predicted from historical data. More tests with shorter window frames before the disturbance occurred were undertaken. The tests which were initiated by the 20-second window, included the 10-second window, the 4-second window, the 1-second window and the 500 ms seconds window.

Machine Learning applications require a dataset to be split into the training set as well as the test set to verify the validity of the model. In many publications, the suggested approach uses a ratio of 80/20, where 80% is used for training and 20% used for testing. In this case, the data was split accordingly in the following manner:

- Training Set - 80%: This included labelled network disturbances. In the first instance, these disturbances are labelled **0** and **1**. In the second test, the disturbances are labelled according to their respective eigenvalue positions.
- Test Set - 20%: a set of new or unseen data used to assess the model's generalization and performance.

6.5 Eigenvalue Region Prediction

In the previous phase, the observations were labelled with either a 1 or a 0 for positive and negative disturbances respectively. The following phase of the study aims for the observations to be labelled with their eigenvalue position of the disturbance. The eigenvalues in this case were derived using the Matrix-Pencil method. This method is described in detail in Chapter 5 and was compared with the other two methods, namely FFT and Prony Analysis in terms of performance. Eigenvalues for all the 24 disturbances were derived from the Matrix-Pencil method using the MATLAB R2020a Simulation tool. Figure 29 and Figure 30 depict the negatively and positively damped signals respectively and the calculated eigenvalues in terms of their frequency and damping. The highlighted segments in Table 4 and Table 5 are the eigenvalues of interest: the inter-area modes of oscillation, i.e. modes having a 0.3 Hz frequency of oscillation.

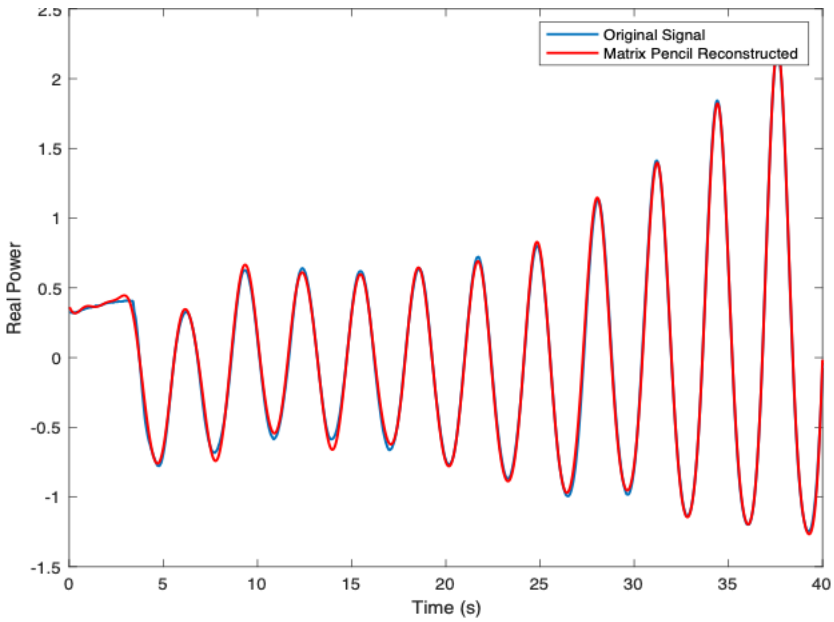

Figure 29: Negative Damping with Matrix Estimation

Table 4: Negatively Damped Estimated Parameters

sigma	Damping Ratio	Frequency	Amp	Theta
-0.0804	3.6066	0.3546	0.1911	1.2170
-0.0804	3.6066	-0.3546	0.1911	-1.2170
-0.1750	8.6146	0.3221	0.6736	1.3266
-0.1750	8.6146	-0.3221	0.6736	-1.3266

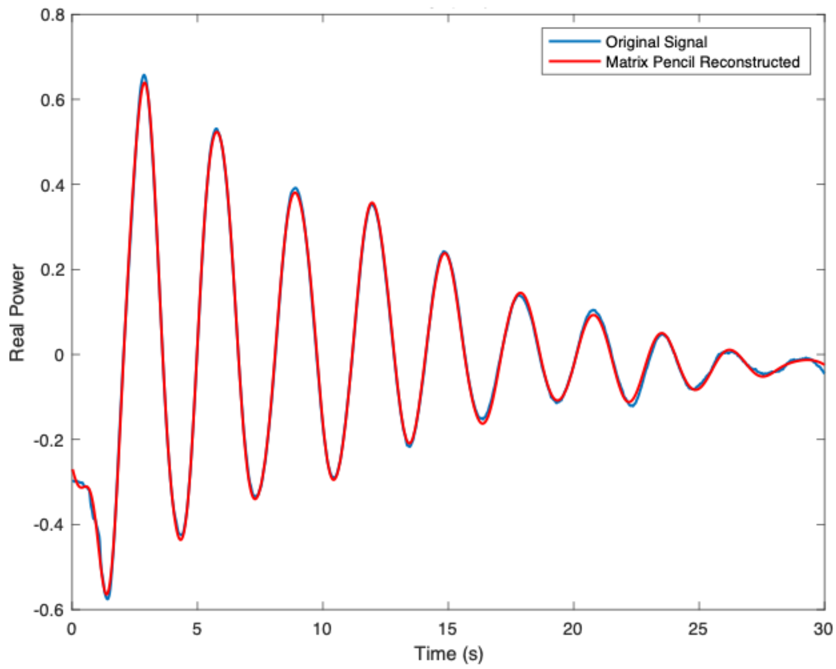

Figure 30: Positive Damping with Matrix Estimation

Table 5: Positively Damped Estimated Parameters

sigma	Damping Ratio	Frequency	Amp	Theta
0.0581	-2.9759	0.3108	0.0961	-1.1627
0.0581	-2.9759	-0.3108	0.0961	1.1627
-0.2211	11.255	0.3107	1.3665	-0.5297
-0.2211	11.255	-0.3107	1.3665	0.5297
-0.1032	5.1528	0.3186	1.2140	-0.5565
-0.1032	5.1528	-0.3186	1.2140	0.5565

The calculated eigenvalues for all the disturbances were then plotted on the real-imaginary plane, as shown in Figure 31. As can be seen, the events are clearly defined to be either on the left-hand side of the plane, positively damped, or on the right-hand side of the plane, negatively damped.

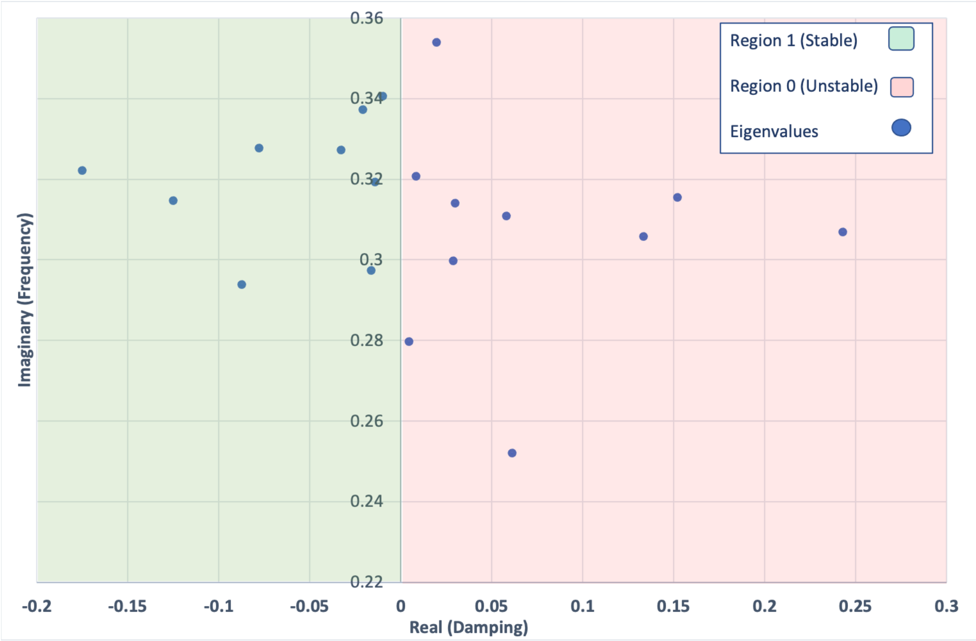

Figure 31: EigenvaluePlot

The eigenvalue plot is further divided into regions where the machine learning process will be trained on and it should be able to predict new or test data. These regions are allocated to be Region 1 and Region 2, as shown in Figure 31. An option to create four or six regions depending on the clustering of eigenvalues is available. However, this poses a risk of too many resultants or classes on a few observations or disturbances. Too many regions will require a lot more data to train the machine. So in this case, the left and right sections of the plane are split in half. The next Chapter will show the results of the simulations.

6.6 Conclusion

The aim of this study is to investigate the use of machine learning to predict power system events. Events are in the form of small-signal disturbances and real measured PMU data was utilised in the study. LSTM proved to be the most suitable machine learning algorithm for time-series prediction, which is the type of data provided by PMU devices. This Chapter outlined the process and method employed to perform small-signal power system events predictions using LSTM machine learning algorithm on real PMU data. The prediction of the eigenvalue region, which describes the nature of the small-signal event, was also explored after generating eigenvalues from the Matrix Pencil estimation method.

The next Chapter presents and evaluates the results of the simulation

7 Results and Discussion

The study results and discussion based on the LSTM model and standardised PMU data from five different PMUs are presented in this chapter to demonstrate the model's performance in power system disturbance prediction.

7.1 Prediction Method Verification

The LSTM model was executed on the Matlab R2020a Simulation tool with the initialised parameters in Chapter 6. These parameters are, the number of Epochs, the number of hidden layers and optimizer = Adam at 0.3 decay. The initial window length or the batch size was selected to be 1000 or 20 seconds before the event happens. Shorter window lengths were also investigated to evaluate the performance further. The window lengths investigated are 10 seconds, 4 seconds, 1 second and 500ms window lengths as shown in Figures 32, 33, 34 and 35.

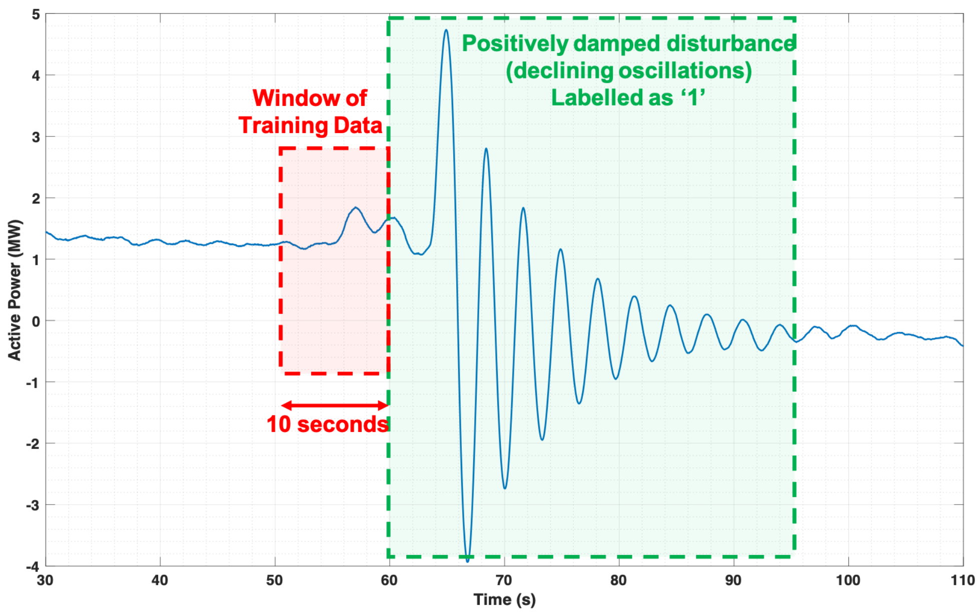

Figure 32: Positively damped active power signal showing the 10 second training data window and the disturbance label

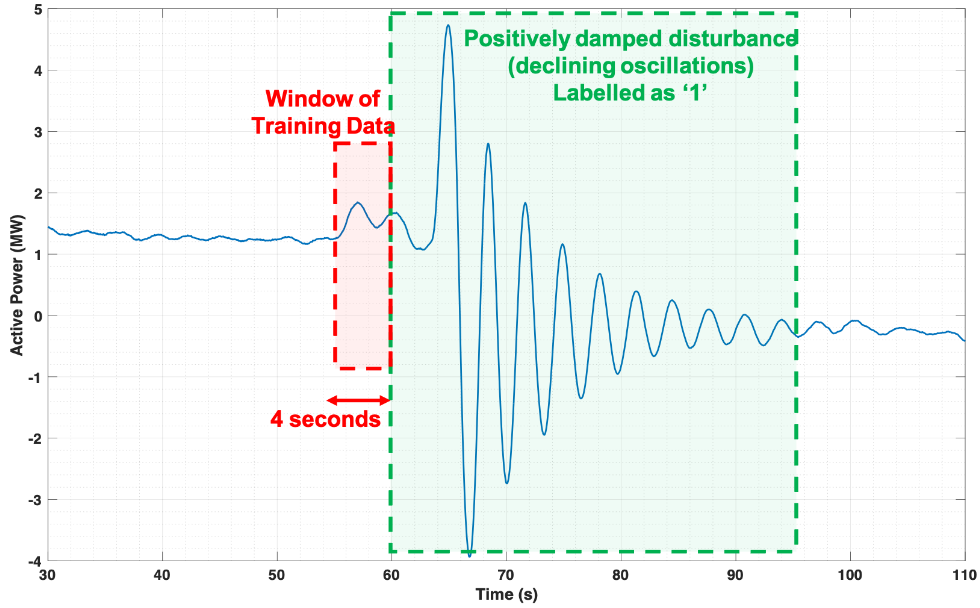

Figure 33: Positively damped active power signal showing the 4 second training data window and the disturbance label

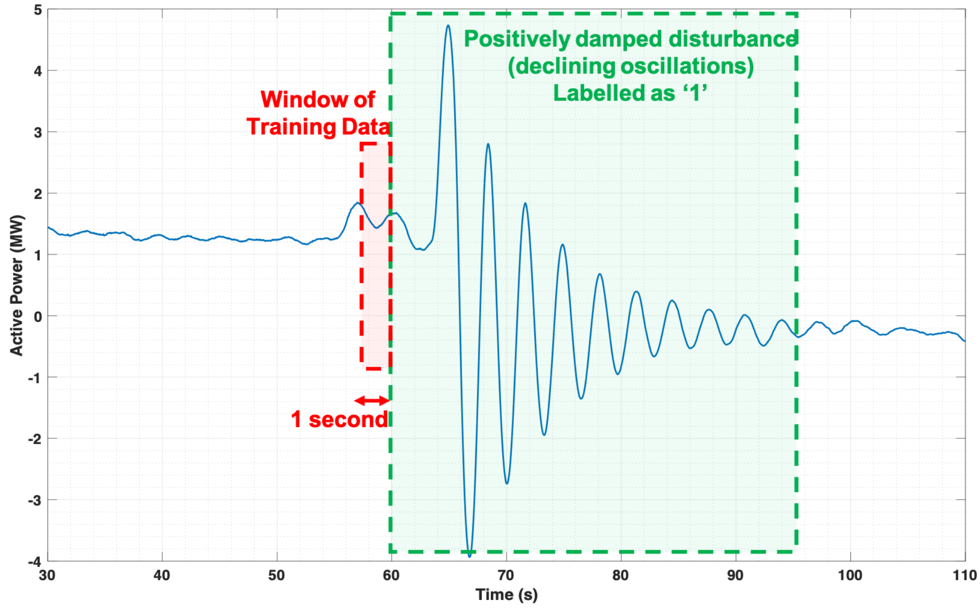

Figure 34: Positively damped active power signal showing the 1 second training data window and the disturbance label

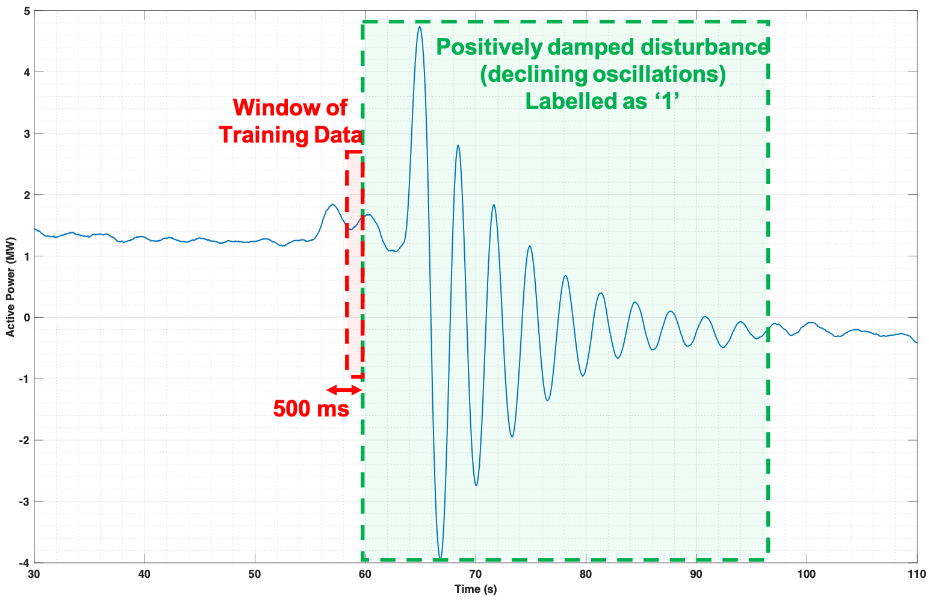

Figure 35: Positively damped active power signal showing the 500 milli-second training data window and the disturbance label

The 10 second window was made up of 500 data points. The 4 second window was made up of 200 data points. The 1 second and the 500 ms windows were made up of 50 and 25 data points respectively. The window lengths were selected based on the reasons stated in Chapter 6, which are:

- 20 Second window - Typical duration of the oscillation where the state of the oscillation can be determined (positive or negatively damped).
- 10 Second - Based on half the time of the 20 second window.
- 4 Second - Based on the SCADA sampling rate.

Jakob Nielson states that response times of 1 second or less are fast enough for web application users to feel they are interacting freely with the information [62]. The next window frames are based on this concept. The 1 second and the 500ms window frames were selected as part of the investigation on what could be the shortest possible window frame that can be used for prediction.

Finally, the feature size reduction to observe the accuracy of the prediction was also evaluated. Feature size reduction was carried out by excluding data from the furthest located PMUs on the network in a systematic order. All the other parameters were kept constant during this part of the study.

The five PMUs that were interrogated for their data are shown on the map in Figure 24. The order that was followed for the exclusion of the PMUs each time a study was executed was Western Cape (1), followed by Kwazulu Natal (2), then Gauteng (3), North West (4) and finally Limpopo (5), where the inter-area mode of oscillation is most observable located.

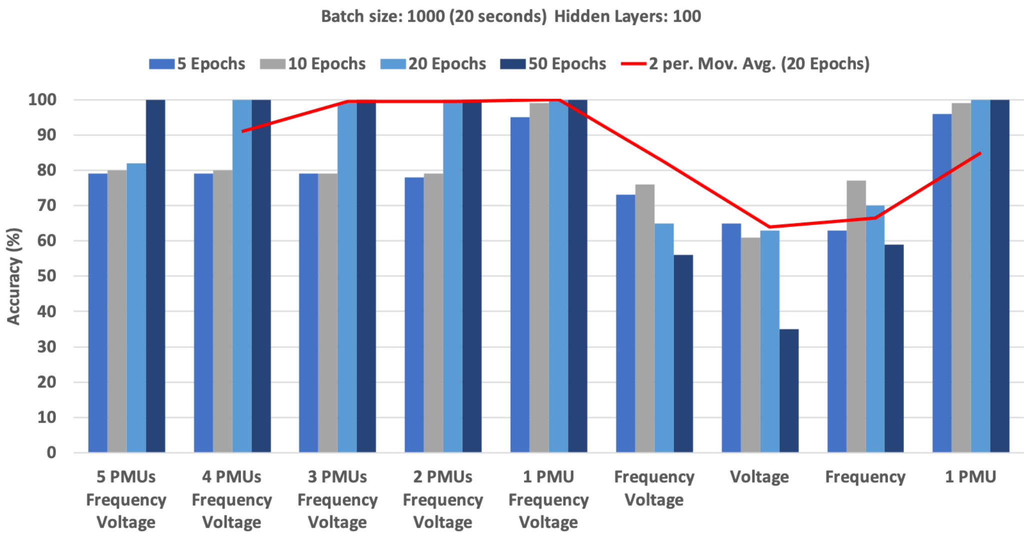

Figure 36: 20 Seconds window accuracy measure with comparison over the number of PMUs and numbers of iterations during training (Epochs)

The 20-second window accuracy results are depicted in Figure 36. This window has a batch size of a 1000 data points for each observation/event. The results show that with many epochs or iterations, higher accuracy of prediction is achieved. However, frequency and voltage features alone have little contribution to the high prediction accuracy, even at higher epochs. The 20 epoch moving average trend in Figure 36 depicts this poor accuracy from the voltage and frequency features.

That means that, high prediction accuracy cannot be achieved using only frequency and voltage as features. It must be highlighted that the poor accuracy has little to do with frequency and voltage as features but a lot to do with the size of the feature size. There is one feature in voltage and one feature in frequency. Current magnitude, current angle, voltage magnitude, voltage angle and frequency are important features to consider in this regard. These measured signals preceding the event, contain valuable information about the actual event response.

A high number of epochs increases the accuracy but the drawback with a high number of epochs is the amount of processing resources required from the computer hardware and the time it takes to produce a result. In this instance, the number of PMU devices used in this study is five, which amounts to a considerable amount of data. The execution takes about 2 seconds at 50 epochs compared to 0.5 seconds for 1 PMU data simulation. So a larger PMU set (many features) and a high epoch number will result in a long execution time and a more accurate prediction. It can also be observed that the PMU closest to the disturbance, 1 PMU, can provide more accurate prediction with an average of around 90% across all epochs.

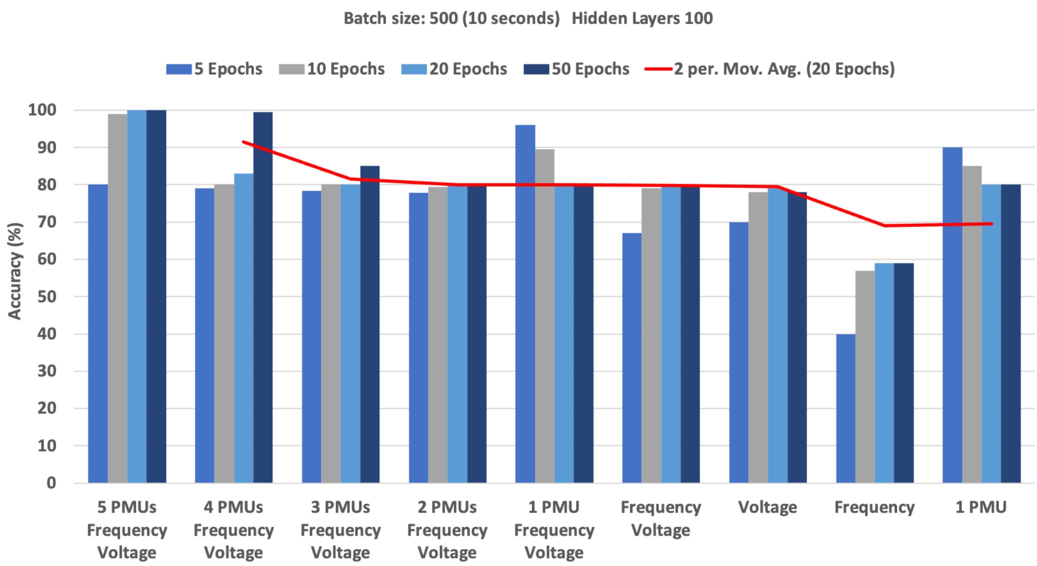

Figure 37: 10 Seconds window accuracy measure with a comparison over the number of PMUs and numbers of iterations during training (Epochs)

The 10-second window provided similar results to the 20-second window Figure 37. The 10-second window results prove that a shorter window or a reduced batch size can still be used successfully in the prediction of disturbances. This 10-second window has a batch size of 500 data points for each observation/event. Frequency showed poor performance with the lowest accuracy of around 60% at 20 epochs. Voltage, however, provided a better result than the 20-second window with an accuracy of around 80% at 20 epochs. This could be due to the voltage feature being a good predictor at 10-seconds window frame and not at larger or lower window frames, as will be seen later in the shorter windows results. The challenge observed with small epochs is that, the simulations needed to be executed several times to get an average result. That means that the machine requires more training sessions from shorter windows because results vary widely between simulations. Larger epochs were more stable but did not improve the accuracy, as was observed with the 20-second window.

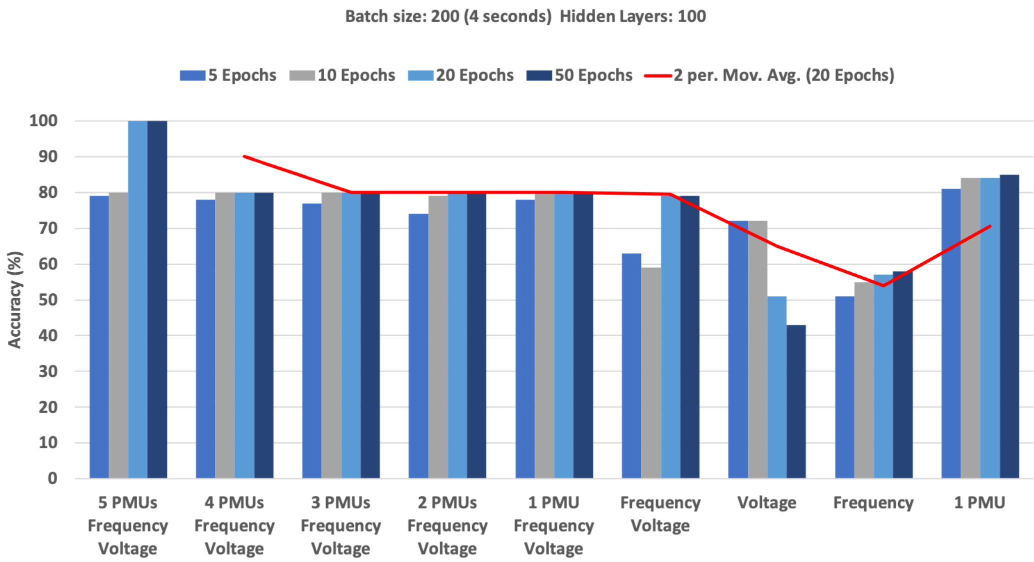

Figure 38: 4 Seconds window accuracy with a comparison over the number of PMUs and numbers of iterations during training (Epochs)

The 4-second window length (200 data points) results, depicted in Figure 38, showed the same behavior as in the 10-second window. The exclusion of the furthest PMUs did not impact the accuracy, frequency alone showed poor performance and the PMU closest to the disturbance gave around 80% accuracy across all the epochs.

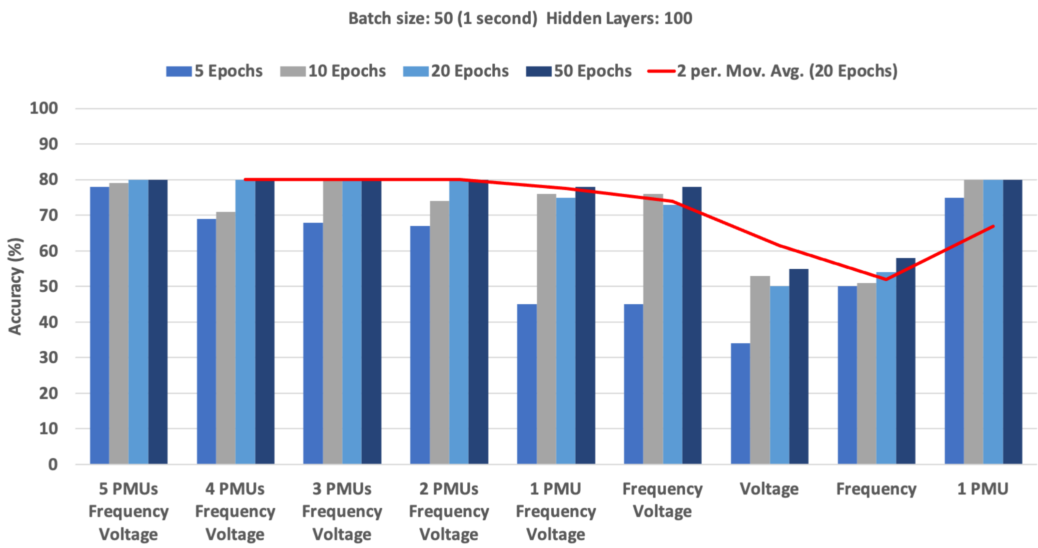

Figure 39: 1 Second window accuracy measure with a comparison over the number of PMUs and numbers of iterations during training (Epochs)

The 1-second window (50 data points), Figure 39, showed a reduction in the accuracy at low epochs as the number of PMUs or feature size reduced. Voltage and frequency, however, continued to show a more unsatisfactory performance with an accuracy of around 50%. The PMU closest to the event continued to show around 80% accuracy across all epochs.

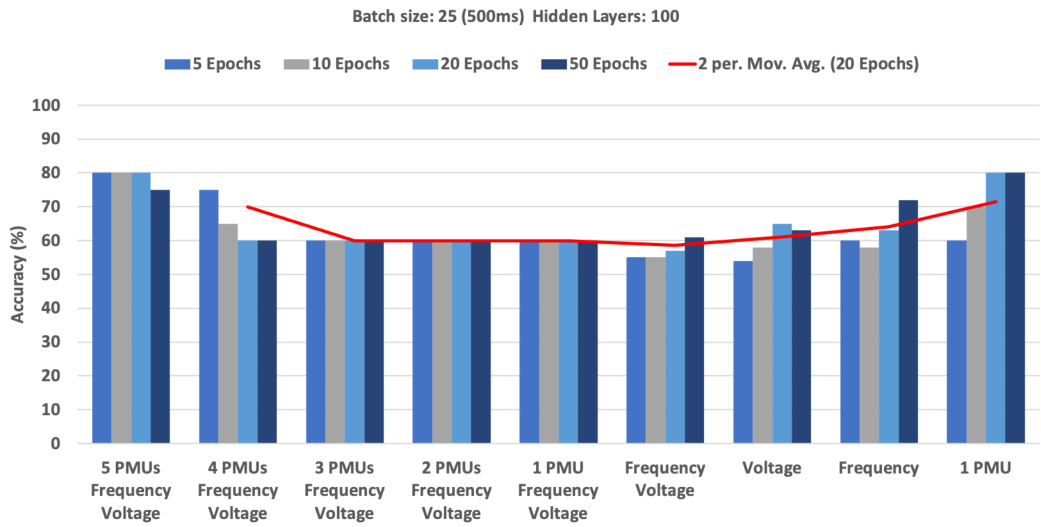

Figure 40: 500ms window accuracy measure with a comparison over the number of PMUs and numbers of iterations during training (Epochs)

The 500 milli-seconds (25 data points) was the shortest window to be investigated (Figure 40). This window showed dependency on a large PMU data, i.e. all PMUs in place. As the PMU devices were excluded, the accuracy also reduced across all the epochs but stabilised to around 60% accuracy. The PMU closest to the disturbance proved to accurately predict the disturbance with an accuracy of around 70% at ten epochs.

7.2 Eigenvalue Area Prediction

The initial phase where the LSTM model was tested proved that a prediction can be achieved based on the preceding window data to predict a labeled disturbance. The disturbances in that case, however, were labeled based on visual judgement. That is to say that knowledge of negatively damped (growing oscillations) and positively damped (declining oscillations) oscillations were used to label the various disturbances accordingly.

The next phase of the study was to perform the same LSTM model evaluation.

For instance, the disturbances are labelled according to their respective eigenvalue region on the eigenvalue plane described in the previous chapter (Chapter 6 section 6.5).

Eigenvalues were calculated using the Matrix Pencil Method. These eigenvalues were then labelled as regions on the eigenvalue plane, and the LSTM model was trained on these labelled regions. The regions were labelled into left and right regions that represented positively (stable) and negatively (unstable) damped oscillations. The LSTM model was then tested on five new events and the results are presented in Figure 41.

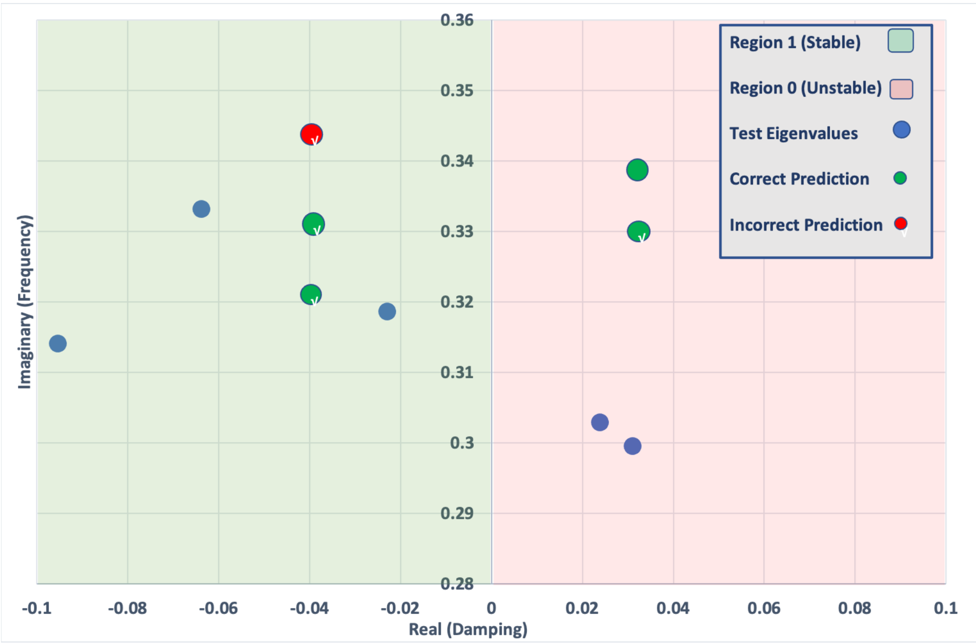

Figure 41: Eigenvalue plane showing the test eigenvalues and the predicted eigenvalues according their respective regions

In Figure 41, it can be seen that the LSTM was able to predict 4 out of the 5 events correctly. The test eigenvalues depict the exact location of the eigenvalue on the eigenvalue plane. These were, however, labelled as either being 1 (stable) or 0 (unstable). Therefore, the "Correct Prediction" points on the graph do not represent the exact eigenvalues but rather the 1 or 0 region. This is similar for the "Incorrect Prediction" point on the graph. The accuracy of 4 out of 5 amounts to

80% and this is similar to the average initial prediction test as demonstrated in the previous section.

7.2.1 Real World Application

The prediction methods described in this document show good performance in predicting oscillatory disturbances on the network. Different window lengths were investigated. However, the application in the real power system control scenario would prove somewhat challenging. It would not offer the controller sufficient time to take decisive action to prevent or avert the oscillation from happening. Switching in or tripping relevant devices can help prevent threatening oscillations. A 10-second buffer between the disturbance and the training data was implemented to address this concern. This 10 seconds buffer zone data is excluded in the training data and will be considered the time for any decisive action taken by the controller. The 10-second buffer is illustrated in Figure 42.

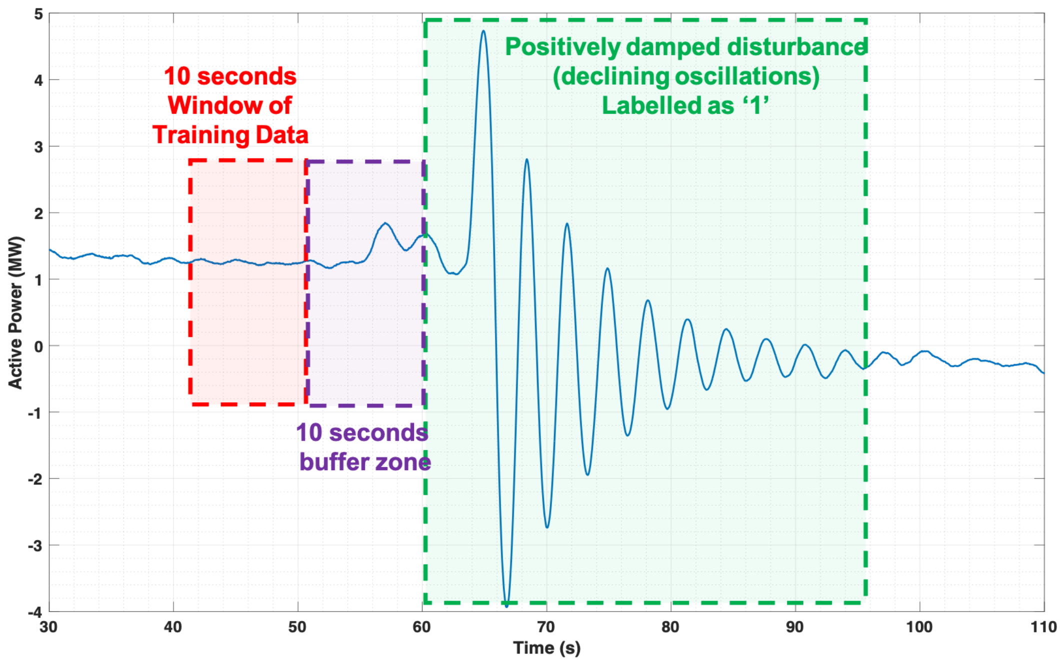

Figure 42: Active Power signal with a 10-second buffer zone between the training data window and the labeled disturbance window

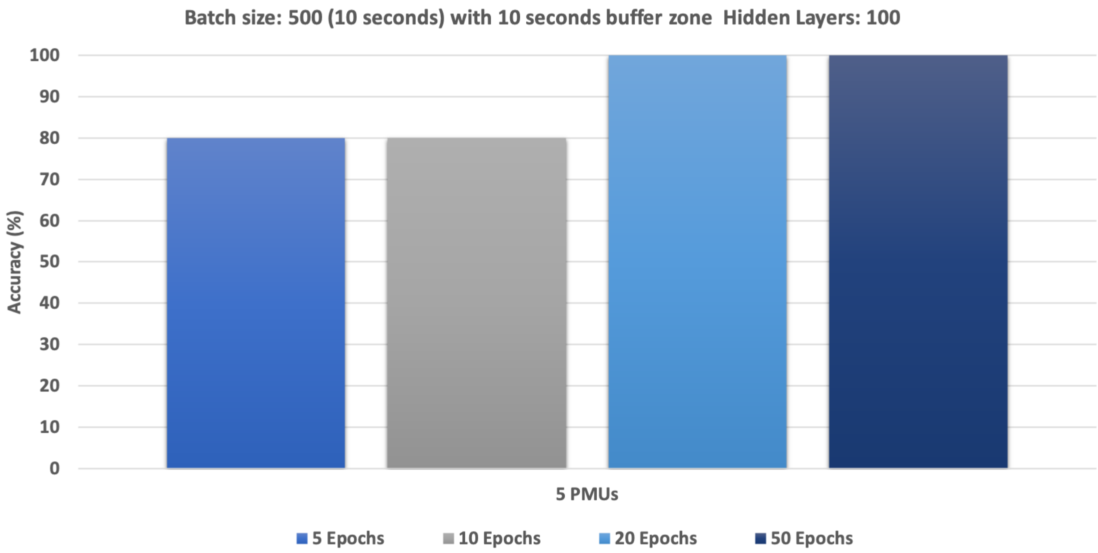

Figure 43: 10 seconds buffer accuracy measurement

This model was further tested using new unseen data and it showed good performance as shown in Figure 43 at ten epochs the model was able to achieve an 80% accuracy.

7.3 Discussion

The results have shown the significance of high-resolution PMU data in the power systems network's stability prediction. The initial part of the study considered a window length or 20 seconds or batch size of 1000 data points and 36 features from 5 geographically vast PMU devices for the machines's training. The aim was to prove the prediction ability of the LSTM model.

The 20-second window's performance is shown in Figure 36 where the positive damped (1) and the negatively damped (0) disturbances were correctly predicted. Shorter window lengths were examined and their performance is depicted in Figures 37,38, 39 and 40. These results were able to provide the following findings:

- The accuracy of the model reduced as the window lengths became shorter.

Short window sizes do not provide the model with long enough dependencies and therefore neglecting important information. Large windows will also result in overfitting as can be seen with the 20-second window.

- Epoch size directly impacts the accuracy of the model. The trade off with a large epoch size required a great amount of processing power and time to process. Ten epochs showed good performance across all window lengths and reduced features.

- Batch size has a direct impact on the accuracy of the model, however, with a few conditions. A large batch size or feature size will result in an accurate prediction. Voltage and frequency (together or individually) are poor predictors of power system oscillation events during long window frames. Their improvement in accuracy is observed over a short window frame of 500ms with an accuracy of around 60%.

- The PMU located in Limpopo showed, across all epochs and window lengths, to be an accurate predictor of disturbances. The focus of this study was on the inter-area oscillations of 0.3 Hz in nature. These are naturally best observed at the inter-connection point, which is where the Limpopo PMU is situated. This is proved from these results.

The studies can then adopt the optimal window length to be 10 seconds at 10 epochs with all the five PMUs in place. Shorter windows showed a much faster computation than the longer one but were unstable in their accuracy.

The eigenvalue region prediction using LSTM and the 10 second window length was evaluated to answer the research question further. The results showed the model accuracy in predicting where the possible eigenvalues are located on the eigenvalue plane for the inter-area modes of oscillation or oscillations that exhibit the mode frequency of around 0.3 Hz. The success of the LSTM model can be attributed to its memory function.

7.4 Conclusion

This Chapter presented the simulation results of the LSTM model. The model performance was evaluated in terms of the window lengths and a reduced dataset. Reduction of the dataset refers to the number of PMU devices that are used for the study. The prediction of the actual eigenvalue region was also evaluated and further tested on new unseen data. A real-world application method was also introduced, which implemented a buffer zone that would afford a small time frame for any decisive action to avert an oscillation disturbance.

The following Chapter concludes this dissertation and makes final remarks on future work.

8 Conclusions

8.1 Summary

This dissertation introduced the concept of Power system stability which is an important aspect of the integrated power system network's secure operation. In particular, small-signal stability, caused by small disturbances on the network, can result in local and inter-area power oscillations. These oscillations, if not controlled or damped, have the potential of causing a system blackout. A complex non-linear power system's dynamic behaviour was also described in terms of eigenvalues, which consist of damping and frequency.

Access to synchrophasor measurements from PMUs has provided the ability to monitor the network's performance in near real-time. PMUs have been able to provide high-resolution time-stamped data which has brought about the possibility of dynamic monitoring, post-event analysis, model bench marking, and improved state estimation of the network. PMUs in the South African context have shown significant benefits in post-event analysis where the traditional SCADA system came short in resolving the measurements.

This dissertation provided the background on the power system stability and the operation of the PMU. It further explored the different measurement-based eigenvalue extraction methods and their respective drawbacks. These included the FFT which used a curve fitting method, the Prony analysis, which bears a close relation to ARMA and the Matrix Pencil Method which fits a sum of damped sinusoids to evenly sampled PMU data. Prony and the Matrix Pencil Method used mathematical models in the form of Singular Value Decomposition to estimate the eigenvalues from the measured signals. It was established that the Matrix Pencil Method is more stable and less prone to noise than the other two methods.

High-resolution PMU data was acquired from five PMU devices spread across the Eskom network for this study. The PMU devices were strategically selected due to their locations being on the critical areas on the power system. The data was pre-processed in the form of data standardization, Real and Reactive Power calculations were carried out in preparation for the simulations' machine learning aspect.

Some machine learning algorithms were discussed in detail, and the most applicable algorithm for this time-series type of data was a Deep Learning Neural Network in the form of LSTM. The memory aspect of the LSTM was shown to be ideal in the pattern recognition of the data. The LSTM method was applied to PMU data. It showed good performance in predicting positive or negatively damped disturbances on the network using a 20-second window before the disturbance. Other window lengths were also explored but showed lower performance. The trade-off was between the number of epochs and feature size or the number of PMUs required for a prediction.

LSTM further showed good performance in predicting the eigenvalue area on the eigenvalue plane for the various events. The dissertation proposed a 10-second buffer window between the disturbance and a 10-seconds window of training data. In the real world, this method would be able to provide the controllers with some time to take decisive action to prevent the oscillation. This method showed good performance in the testing where an accuracy of 80 % was achieved at 20 epochs and all five PMUs in place.

8.2 Conclusion

In this dissertation, the problem of power system stability, particularly small signal stability disturbances was addressed. Synchrophasor technology through the use of PMUs are well positioned to monitor these disturbances on the network. Case studies across the globe and in South Africa have shown great benefits that have been derived from the use of PMUs in the real-time monitoring of the power system dynamic behaviour.

Mathematical models such as FFT, Prony Analysis and the Matrix Pencil Estimation method were involved in a comparative study to extract eigenvalue parameters from the measured PMU signals or data. The study showed the Matrix Pencil Estimation method to be more stable and less prone to noise. It was therefore used in a measurement-based eigenvalue analysis for the machine learning disturbances.

The research's main focus was to use real measured PMU data from vastly located PMUs on the power system, in a supervised machine learning algorithm, to predict small-signal disturbances occurring on the network. The proposed approach was to label the actual disturbance according to its position on the eigenvalue plane and use a window of PMU data before the disturbance to train and test the prediction accuracy. Several machine learning algorithms were investigated, and the Neural Network Long Short Term Memory (LSTM) proved to be efficient in time-series prediction applications. The LSTM and eigenvalue region prediction using PMU data showed promising results achieving an 80% accuracy.

A further improvement on the initial approach suggested an introduction of a buffer zone which would be the data window located in between the disturbance and the training data window frame. This buffer zone would serve as a time period for any decisive action to be taken by the controller to prevent the occurrence of a disturbance. This improvement showed good results in the accuracy performance achieving an accuracy of 80% at low iterations.

8.3 Future Work

This dissertation's main focus was on eigenvalue area prediction using PMU data and only utilized five PMU devices to achieve this. Eigenvalue area is related to the small-signal stability determination of the network. Future research could consider an increased number of PMU devices for machine learning. This would be able to provide a more concrete prediction and increased reliability of the power network. Future networks are becoming more intelligent through the use of synchrophasor measurements in the protection and control of the network [63]. Over and above the network experiences stability problem, other matters of concern that can be addressed under the machine learning prediction scope would be the system inertia estimation. This is on the backdrop of the high penetration of renewable energy of the modern power system.

9 References

[1] Prabha Kundur. *Power System Stability and Control.* The EPRI Power Systems Engineering Series. McGraw-Hill, 1994. ISBN: 0-07-035958-X.

[2] N. Mithulananthan, D.Thakur and K. Prasertwong. 'Understanding Low Frequency Oscillation in Power Systems'. In: *International Journal of Electrical Engineering and Education* 47.3 (July 2010), pp. 248–262. DOI: 10.7227/IJEEE.47.3.2.

[3] I. Erlich et al. 'Assessment of the Small Signal Stability of the European Interconnected Electric Power System Using Neural Networks'. In: *IEEE Explore* (Feb. 2001), pp. 158–161. DOI: 10.1109/LESCPE.2001.941643.

[4] IEEE. 'Proposed Terms Definitions for Power System Stability'. In: *IEEE Transactions on Power Apparatus and Systems* PAS-101.7 (July 1982), pp. 1894–1898. ISSN: 0018-9510.

[5] P. Kundur et al. 'Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions'. In: *IEEE Transactions on Power Systems* 19.3 (Aug. 2004), pp. 1387–1401. ISSN: 0885-8950.

[6] Dr B Kaylan Kumar. 'Power System Stability and Control'. In: 2004.

[7] A.G. Phadke, M. Ibrahim and T. Hlibka. 'Fundamental basis for distance relaying with symmetrical components'. In: *IEEE Transactions on Power Apparatus and Systems* 96.2 (Mar. 1977), pp. 635–646. ISSN: 10.1109/T-PAS.1977.32375.

[8] Bindeshwar Singh et al. 'Applications of phasor measurement units (PMUs) in electric power system networks incorporated with FACTS controllers'. In: *International Journal of Engineering Science and Technology* 3.3 (July 2011), pp. 64–82. DOI: 10.4314/ijest.v3i3.68423.

[9] A. G. Phadke and J. S. Thorp. *Synchronized Phasor Measurements and Their applications.* Vol. 1. 2008.

[10] Psymetrix. *PhasorPoint User's Guide Software Version 6.0.* Dec. 2012.

[11] V Vignesh, S. Chakrabarti and S. C. Srivastava. In: *An experimental study on the load modelling using PMU measurements*. Apr. 2014, pp. 1–5. DOI: 10.1109/TDC.2014.6863487.

[12] S. Kolluri et al. 'Island formation in entergy power grid during Hurricane Gustav'. In: July 2009, pp. 1–5. DOI: 10.1109/PES.2009.5275340.

[13] A.H. Nielson et al. 'PHASOR MEASUREMENT UNITS IN THE EASTERN DANISH POWER SYSTEM'. In: *CIGRE Conference 2006*. CIGRE Conference 2006 Conference date: 01-01-2006. CIGRE (International Council on Large Electric Systems), 2006, pp. 1–5. ISBN: 2-85873-020-2.

[14] M. Wald. *New Tools for Keeping the Lights On - The New York Times*. URL: http://www.nytimes.com/2013/08/01/business/%20new-tools-for-keeping-the-lights-on.html?ref=science. (accessed: 01.11.2020).

[15] C. Lu et al. 'Advancing China's Smart Grid: Phasor Measurement Units in a Wide-Area Management System'. In: *IEEE Power and Energy Magazine* 13.5 (Sept. 2015), pp. 60–71. ISSN: 10.1109/MPE.2015.2432372.

[16] R. Leandro et al. 'Monitoramento on-line de oscilações eletromecânicas no SIN utilizando sincrofasores'. In: May 2014.

[17] E. Martinez. 'Wide area measurement and control system in Mexico'. In: (Apr. 2008), pp. 156–161. ISSN: 10.1109/DRPT.2008.4523394.

[18] Teboho Machabe, Ellen De Mello Koch and Ken Nixon. 'A Review of Power System Instability Prediction Methods Using Phasor Measurement Unit Data'. In: (2020), pp. 1–6. DOI: 10.1109/SAUPEC/RobMech/PRASA48453.2020.9041084.

[19] B. Pal and B. Chaudhuri. *Robust Control in Power Systems*. U.S: Springer, 2005. ISBN: 978-0-387-25950-5.

[20] HyungSeon Oh. 'Situational Awareness with PMUs and SCADA'. In: (June 2017).

[21] Duc Nguyen et al. 'Smart Grid Line Event Classification Using Supervised Learning Over PMU Data Streams'. In: 2015, pp. 1–8. DOI: 10.1109/IGCC.2015.7393695.

[22] Bo Yang et al. 'Big Data Analytic Empowered Grid Applications - Is PMU a big data issue?' In: May 2015, pp. 1–4. DOI: 10.1109/EEM.2015.7216718.

[23] Judith Hurwitz and Daniel kirsh. *Machine Learning for Dummies.* John Wiley and Sons, Inc, 2018. ISBN: 978-1-119-45495-3.

[24] Brian Boucheron and Lisa Tagliaferri. *Machine Learning Projects.* Digital Ocean, May 2019. ISBN: 9780999773024.

[25] IAdrian Stetco et al. 'Machine learning methods for wind turbine condition monitoring: A review'. In: *Renewable Energy* 133 (2019), pp. 620–635. ISSN: 0960-1481.

[26] Dean Abbort. *Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst.* John Wiley and Sons, Inc, May 2018. ISBN: 978-1-118-72796-6.

[27] Laura Edell. *Is Machine Learning the New EPM Black?* URL: https://scorecardstreet.wordpress.com%20/2015/12/09/is-machine-learning-the-new-epm-black/. (accessed: 17.08.2019).

[28] Reza Yousefian and S. Kamalasadan. 'A Review of Neural Network Based Machine Learning Approaches for Rotor Angle Stability Control'. In: *IEEE Explore* abs/1701.01214 (2017).

[29] Md Zahangir Alom et al. 'A State-of-the-Art Survey on Deep Learning Theory and Architectures'. In: *Electronics* 8.3 (5 March 2019).

[30] Ariruna Dasgupta and Asoke Nath. 'Classification of Machine Learning Algorithms'. In: *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 03.03 (March 2016), pp. 6–11.

[31] Leslie Pack Kaelbling, Michael L. Littman and Andrew P. Moore. 'Reinforcement Learning: A Survey'. In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285. URL: http://people.csail.mit.edu/lpk/papers/rl-survey.ps.

[32] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep Learning'. In: *Nature -Insight Review* 521 (28 May 2015), pp. 436–444.

[33] Zhou Ning, Jian Yin and Bora Akyol. 'Spectral Analysis of Power Grid PMU Data'. In: *Pacific Northwest National Laboratory* (June 2012).

[34] De Callafon Raymond and Charles Wells. 'Distributed Real-Time Electric Power Grid Event Detection and Dynamic Characterization'. In: *CIGRE US National Committee 2014 Grid of the Future Symposium* (2014).

[35] S. P. Teeuwsen, I. Erlich and M. A. El-Sharkawi. 'Small-Signal Stability Assessment based on Advanced Neural Network Methods'. In: *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491)*. Vol. 4. 2003, pp. 2349–2649. DOI: 10.1109/PES.2003.1271000.

[36] Changgang Li et al. 'Measurement Based Power System Dynamics Prediction with Multivariate AutoRegressive Model'. In: *2014 IEEE PES T D Conference and Exposition*. 2014, pp. 1–5. DOI: 10.1109/TDC.2014.6863288.

[37] Rui Zhang et al. 'Network Anomaly detection Using One Class Support Vector Machine'. In: *International Multiconference of Engineers and computer Scientists* 2168 (Mar. 2008).

[38] Y.Y. HSU, C.R. Chen and C.C. Su. 'Analysis of Electromechanical Modes using Artificial Neural Network'. In: *IEE Proceedings - Generation, Transmission and Distribution* 141.3 (1994), pp. 198–204. DOI: 10.1049/ip-gtd:19949872IETDigitalLibrary.

[39] N. I. Sapankevych and R. Sankar. 'Time Series Prediction Using Support Vector Machines: A Survey'. In: *IEEE Computational Intelligence Magazine* 4.2 (May 2009), pp. 24–38.

[40] W.C. Hong. 'Application of Seasonal SVR with Chaotic Immune Algorithm in Traffic Flow Forecasting'. In: *Neural Computing and Applications* 21.3 (Apr. 2012), pp. 583–593.

[41] Yuxiu Hua et al. 'Deep Learning with Long Short-Term Memory for Time Series Prediction'. In: *IEEE Communications Magazine* 57.6 (2019), pp. 114–119. DOI: 10.1109/MCOM.2019.1800155. URL: https://dblp.org/rec/journals/corr/abs-1810-10161.bib.

[42] S. Hochreiter and J. Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[43] Alex Graves and JÃrgen Schmidhuber. *Framewise phoneme classification with bidirectional lstm and other neural network architectures*. Vol. 18. 18(5):602-610 5-6. 2005, pp. 602–610. DOI: https://doi.org/10.1016/j.neunet.2005.06.042.

[44] Wojciech Zaremba, Ilya Sutskever and Oriol Vinyals. 'Recurrent Neural Network Regularization'. In: (2014). URL: https://arxiv.org/abs/1409.2329.

[45] Denis Smirnov and Engelbert Mephu Nguifo. 'Time Series Classification with Recurrent Neural Networks'. In: *Advanced Analytics and Learning on Temporal Data* 8 (2018).

[46] Sepp Hochreiter and Jürgen Schmidhube. 'Long short-term memory'. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[47] Dan Lin. 'Methods for Analyzing Power System Small Signal Stability'. PhD thesis. Memorial University of Newfoundland, 2015.

[48] A. Fernández Rodríguez et al. 'Coding Prony's method in MATLAB and applying it to biomedical signal filtering'. In: *BMC Bioinformatics* 19.451 (2018). DOI: https://doi.org/10.1186/s12859-018-2473-y.

[49] Javad Khazaei et al. 'Distributed Prony analysis for real-world PMU data'. In: *Electric Power Systems Research -Elsevier* 133 (2016), pp. 113–120. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2015.12.008..

[50] S. Avdakovic et al. 'Identification Of Low Frequency Oscillations In Power System'. In: *2009 International Conference on Electrical and Electronics Engineering - ELECO 2009*. 2009, pp. 103–107. DOI: 10.1109/ELECO.2009.5355325.

[51] B. Mohammadi et al. 'Online inter-area oscillation monitoring in power systems using PMU data and Prony analysis'. In: *International Journal of the Physical Sciences* 6(22) (Oct. 2011), pp. 5267–5272. ISSN: 1992 - 1950.

[52] B.Shayanfard, M.Dehghani and A.Khayatian. 'Optimal PMU Placement for Full Observability and Dynamic Stability Assessment'. In: *2011 19th Iranian Conference on Electrical Engineering*. 2011, pp. 1–6.

[53] Shuqing Zhang, Xiaorong Xie and Jingtao Wu. 'WAMS-based detection and early-warning of low-frequency oscillations in large-scale power systems'. In: *Electric Power Systems Research* 78.5 (Aug. 2008), pp. 897–906. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2007.06.008.

[54] Tapan K. Sarkar and Odilon Pereira. 'Using the Matrix Pencil Method to Estimate the Parameters of a Sum of Complex Exponentials'. In: *IEEE Antennas and Propagation Magazine* 37.1 (Feb. 1995), pp. 48–55.

[55] A. Allen et al. 'PMU Data Event Detection: A User Guide for Power Engineers'. In: (Oct. 2014). DOI: 10.2172/1160181. URL: https://www.osti.gov/biblio/1160181.

[56] François Sarrazin et al. 'Comparison between Matrix Pencil and Prony Methods applied on Noisy Antenna Responses'. In: *2011 Loughborough Antennas Propagation Conference*. Nov. 2011, pp. 1–4. DOI: 10.1109/LAPC.2011.6113978.

[57] Nazri Mohd Nawia, Walid Hasen Atomia and M. Z. Rehmana. 'The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks'. In: *Procedia Technology* 11 (2013), pp. 32–39. ISSN: 2212-0173.

[58] Jürgen Schmidhuber. 'Deep learning in neural networks: An overview'. In: *Neural Networks* 61 (Jan. 2015), pp. 85–117. ISSN: 0893-6080.

[59] Felix A Gers, Douglas Eck and Jürgen Schmidhube. 'Applying lstm to time series predictable through time-window approaches'. In: Aug. 2001, pp. 669–676. ISBN: 978-3-540-42486-4. DOI: 10.1007/3-540-44668-0_93.

[60] Ilya Sutskever, Oriol Vinyals and Quoc V Le. *Sequence to sequence learning with neural networks. In Advances in neural information processing systems.* 2014, pp. 3104–3112.

[61] Reimers Nils and Gurevych Iryna. 'Optimal hyperparameters for deep lstm-networks for sequence labeling tasks'. In: *arXiv preprint arXiv:1707.06799* (2017).

[62] Nielsen Jakob. *Usability Engineering.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 0125184050.

[63] V. Terzija et al. 'Wide area monitoring protection and control of future electric power networks'. In: *Proceedings of the IEEE* 99.1 (2011), pp. 80–93. DOI: 10.1109/JPROC.2010.2060450.

10 Appendices

A - About MATLAB R2020a

MATLAB version R2020a was the primary tool used in conduction this research. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User
- Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science.

In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

B - FFT MATLAB Code

The MATLAB code presented in this section performs the FFT on a voltage signal. The signal is initially detrended and filtered for the relevant frequencies. FFT is then performed on the filtered signal and a FFT plot is generated. The final step is to estimate the parameters or coefficients that best describe the FFT plot through the curve fitting method.

```
%FFT Simulation
%this file has the for loop commented out
V_detrend = detrend(test3.V);
%%
Time = test3.Time;
%--------Set up the sampling rate for FFT------------------

Fs = 50;              % Sampling frequency
T = 1/Fs;               % Sampling period
L = length(V_detrend);            % Length of signal
t = (0:L-1)*T;          % Time vector
%----bandpass filter for frequencies between 0.02 and 0.4 ...
    Hz----------
%lowpass(x,150,fs)
[b a]=butter(2,[0.1 0.2],'bandpass');
H = freqz(b,a,floor(100/2));
filtered = filter(b,a,V_detrend);
```

```
%subplot(3,1,2)
%plot(filtered)
%title('Filtered Signal')
%xlabel('t (milliseconds)')
%ylabel('Voltage')
%xlim([0 1260]);
count = 0;
%%
%---perform FFT on the filtered signal--------------------
Y = fft(V_detrend);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
%plot(P1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;

%-------plot FFT signal------------------------------------------

%% Fit: 'untitled fit 2'.
[xData, yData] = prepareCurveData(f,P1);

% Set up fittype and options.
ft = fittype( 'a/sqrt((x^2-w^2)^2+(4*d^2*x^2))', ...
    'independent', 'x', 'dependent', 'y' );
%ft = fittype('a+1/(sqrt(((x)^2-(w)^2)^2+(4*(d)^2*(x+3)^2)))', ...
    'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Algorithm = 'Levenberg-Marquardt';
opts.Display = 'Off';
%opts.StartPoint = [0.588728084842466 0.915735525189067 ...
    0.792207329559554];
opts.StartPoint = [1 1 1];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts);
%subplot(3,1,2);
% Plot fit with data.
%figure( 'Name', 'untitled fit 2' );
%plot( fitresult, xData, yData );
```

```
%legend( h, 'P1 vs. f', 'untitled fit 2', 'Location', ...
    'NorthEast', 'Interpreter', 'none' );
% Label axes
%xlabel( 'f', 'Interpreter', 'none' );
%ylabel( 'P1', 'Interpreter', 'none' );
%grid on
%xlim([0 10]);
coefficients = coeffvalues(fitresult);
T = table(coefficients);
gof;
%[fn,dr] = modalfit(P1,f,Fs,1,'FitMethod','PP')
%x_recon = 3000*ifft(yData,3000,'nonsymmetric'); ...
    %reconstructed signal
%t = [0:1:length(x_recon)-1]/Fs; %recompute time index
%plot(t,x_recon,t,V_detrend,'r');%reconstructed signal
%plot(fitresult, xData, yData);
%fitresult

trial = fitresult(test3.V);
%subplot(2,1,1);
estimated = trial;

hold on
plot(Time,detrend(-10000*estimated));
%plot(estimated);

plot(Time,V_detrend,'r');
hold off
%subplot(2,1,2);
%plot(V_detrend,'r');
```

C - Matrix Pencil Estimation Method MATLAB Code

The Matlab code presented in this section performs the Matrix Pencil method of estimation. The code uses a function, **fMatrixPencil**.

```
%This code performs the Matrix Pencil method of estimation ...
    using the
%fMatrixPencil function. The length of the data must be ...
    divisible by 2.

y = NormalisedMasterDataS1.Power;
t = NormalisedMasterDataS1.Time;
L = length(y)/2;        %The length of the data must be ...
    divisible by 2.
y = detrend(y);
dt =1/50;
tol = 10e-3;
[amp,theta,freq,alpha_percent,y_hat,sigma,z]=fMatrixPencil(y',L,dt,tol);

T = table(z,sigma,alpha_percent,freq,amp,theta);
figure;
plot(t,y,t,y_hat,'r','LineWidth',1.5)
legend('Original Signal','Matrix Pencil Reconstructed')
xlabel('Time (s)')
ylabel('Real Power')
title('Event Test 5')
Plotted_values_Pencil = table(real(y_hat),y);
```

```

function [amp, theta, freq, alpha_percent, y_hat,sigma,z] = ...
    fMatrixPencil(y_noise, L, dt, tol)

%% Matrix Pencil Method
N = length(y_noise)
```

```
t = 0:dt:N*dt-dt;
% L = N/3:1:N/2; % pencil parameter, range n ≤ L ≤ N-n
% L = L(1); % choose which pencil parameter from the range to use
% Define new matrix [Y]
L=L;

Y = zeros(N-L, L+1);  %creates a zero matrix [Y]

for ii = 0:N-L-1                     % the loop propulates the ...
    diagonal
    n = ii + 1;                      % y_noise values
    for jj = 0:L
        Y(ii+1,jj+1) = y_noise(n);
        n = n + 1;
    end
end

%Y=Y;

% SVD to [Y]
[U,S,V] = svd(Y);
s = diag(S);
% figure;
% plot(s,'.-'), title('sigmas of matrix Y')

% tol = 10e-3;
for n = 1:length(s)-1
    if (abs(s(n+1)/s(1)) ≤ tol)
        break;
    end
end

n
% s_threshold = s(1)*0.1
% s(s<s_threshold) = [];
% n = length(s); % system order based on sigma

s = s(1:n);

```

```
V_prime = V(:,1:n);

V1_prime = V_prime(1:end-1,:);
V2_prime = V_prime(2:end,:);
%V2_prime'
%pinv(V1_prime')
A = V2_prime'*pinv(V1_prime');% size is nxn
z = eig(A);

Z = zeros(N,n);
for qt = 0:N-1
    Z(qt+1,:) = (z.').^qt;
end
Z = Z;

R = pinv(Z)*y_noise';
length(R)
amp = abs(R);
theta = atan(imag(R)./real(R));

sigma = log(abs(z))/dt
freq = atan(imag(z)./real(z))/(2*pi*dt)
alpha_percent = -sigma./sqrt(sigma.^2+(2*pi*freq).^2)*100; ...
    %damping ratio

A_matpencil= sortrows([amp, theta, freq, ...
    alpha_percent,sigma,z],1);

y_hat = Z*R;
```

D - Prony Estimation MATLAB Code

The Matlab code presented in this section performs the Prony Estimation method. The method utilises the **polynomialmethod** function.

```
%This code performs the prony analysis using the polynomial ...
    method.
%Also prints the table of modes
x = NormalisedMasterDataS1.Power;
t = NormalisedMasterDataS1.Time;
%x = detrend(test3.V);
%t = test3.Time;
L = length(x);  % length of the signal
Ts =1/50;  %50Hz signal
p = 1445;
[Amp,alfa,freq,theta,y_hat] = polynomial_method(x,p,Ts,'classic');
plot(t,x,'LineWidth',1.5);
hold on
plot(t,y_hat,'r','LineWidth',1)
hold off;
legend('Original Signal','De-noised Prony Reconstruction')
xlabel('Time (s)')
ylabel('Voltage')

T = table(Amp,alfa,freq,theta);
%plot(alfa,freq,'*');
%hold on;
% Plotted_values_Prony = table(real(y_hat),x);
% y_hat_real = real(y_hat);
% avrg_of_x = mean(x);
% avrg_of_y_hat = mean(y_hat_real);
% error_cal = (abs(y_hat_real-x))/abs(x)
```

```
%----This function performs the prony estimation of the mode ...
    parameters using
%the polynomial method from PronyAlg.m
```

```

function[Amp,alfa,freq,theta,y_hat] = ...
    polynomial_method(x,p,Ts,method)
%method: 'classis', 'ls' or 'tls' (case sensetive)
%define the solving methods
CLASSIC = 0;
LS = 1;
TLS = 2;

N = length(x)

if strcmpi(method,'classic')
    % if N≠2*p;
    %     disp('ERROR: length of x must be 2*p samples in ...
           classical method.');
    %     Amp = [];
    %     alfa = [];
    %     freq = [];
    %     theta = [];
    %     return
    % else
            solve_method = CLASSIC;
     %end
elseif strcmpi(method,'LS')
     solve_method = LS;
elseif strcmpi(method,'TLS')
     solve_method = TLS;
else
     disp('ERROR: error in parsing the argument "method".');
     Amp = [];
     alfa = [];
     freq = [];
     theta = [];
     return;
end

%%step 1

T = toeplitz(x(p:N-1),x(p:-1:1));
```

```

switch solve_method
    case{CLASSIC, LS}
        a = -T\x(p+1:N);
    case TLS
        a = tls(T,-x(p+1:N));
end

%check for indeterminate foms
indeterminate_form = sum(isnan(a) | isinf(a));
if (indeterminate_form)
    Amp = [];alfa = [];freq = []; theta = [];
    return;
end

%%step 2
c = transpose([1;a]);
r = roots(c);
e = length(r);
alfa = log(abs(r))/Ts;
freq = atan2(imag(r),real(r))/(2*pi*Ts);

%in case alfa equals to +/- Inf the signal will not be ...
    recovered for n=0

alfa(isinf(alfa)) = realmax*sign(alfa(isinf(alfa)));

%%step 3
switch solve_method
    case CLASSIC
        len_vandermonde = N; %exact case (N=2p) find h with p ...
            samples
    case LS
        len_vandermonde = N; %overdetermined case (N>2p) find ...
            h with N samples
    case TLS
        len_vandermonde = N; %overdetermined case (N>2p) find ...
            h with N samples
end
```

```
Z = zeros(len_vandermonde,p);
for i = 1:length(r)

    Z(:,i) = transpose(r(i)).^(0:len_vandermonde-1);

   % qt = 0:N-1
    %Z(i+1,:) = (r(i).').^i;
end

rZ = real(Z);
iZ = imag(Z);
%here Inf values are substituted by realmax values

rZ(isinf(rZ)) = realmax*sign(rZ(isinf(rZ)));
iZ(isinf(iZ)) = realmax*sign(iZ(isinf(iZ)));

Z = rZ+1i*iZ;
Z = Z;
switch solve_method
    case {CLASSIC,LS}
        h = Z\x(1:len_vandermonde);
        %h = pinv(Z)*x;
    case TLS
        %if exists nan values SVD won't work
        indeterminate_form = sum(sum(isnan(Z) | isinf(Z)));
        if (indeterminate_form)
            Amp = [];
            alfa = []
            freq = [];
            theta = [];
            return;
        else
            h = tls(Z,x(1:len_vandermonde));
        end
end
Amp = abs(h);
theta = atan2(imag(h),real(h));

```

```
y_hat = Z*h;
```

E - Power Calculation and Standardization Script

The measurements provided by the PMU are voltage magnitude, voltage angle, current magnitude and current angle. The Microsoft Visual Basic Macro presented in this section calculates the Active Power and Reactive Power from the aforementioned PMU measurements using equations 41 and 42.

$$P = V \times I \cos\left(\theta_V - \theta_I\right) \tag{41}$$

$$Q = V \times I \sin\left(\theta_V - \theta_I\right) \tag{42}$$

The script further performs standardisation of the data to address the uneven data challenge using equation 43.

$$z_j = \frac{x_j - \overline{x_j}}{\sigma_x} \tag{43}$$

```
Sub calculation()

Dim lastRow As Long
lastRow = Range("A" & Rows.Count).End(xlUp).Row
Dim cell As Range

'Power calculation1

    Range("P2").Select
    ActiveCell.Formula = "P1"
    Range("P3").Select
    ActiveCell.Formula = "=F3*H3*(180/PI())*COS(RADIANS(G3-I3))"
    Range("P3").Select
    Selection.AutoFill Destination:=Range("P3:P" & lastRow)
'end power calc
'normalization1

    Range("Q2").Select
    ActiveCell.Formula = "NormP1"
```

```
    Range("Q3").Select
    ActiveCell.FormulaR1C1 = _
        "= STANDARDIZE(RC[-1],AVERAGE(R3C16:R21002C16),
        STDEV(R3C16:R21002C16))"
    Range("Q3").Select
    Selection.AutoFill Destination:=Range("Q3:Q" &lastRow), ...
       Type:=xlFillDefault

    'Reactive calculation1

    Range("R2").Select
    ActiveCell.Formula = "Q1"
    Range("R3").Select
    ActiveCell.Formula = "=F3*H3*(180/PI())*SIN(RADIANS(G3-I3))"
    Range("R3").Select
    Selection.AutoFill Destination:=Range("R3:R" & lastRow)
'end reactive calc
'normalization reactive1

    Range("S2").Select
    ActiveCell.Formula = "NormQ1"
    Range("S3").Select
    ActiveCell.FormulaR1C1 = _
        "= STANDARDIZE(RC[-1],AVERAGE(R3C18:R21002C18),
        STDEV(R3C18:R21002C18))"
    Range("S3").Select
    Selection.AutoFill Destination:=Range("S3:S" & lastRow), ...
       Type:=xlFillDefault
  '______________________________
    'Power calculation2

    Range("T2").Select
    ActiveCell.Formula = "P2"
    Range("T3").Select
    ActiveCell.Formula = "=F3*J3*(180/PI())*COS(RADIANS(G3-K3))"
    Range("T3").Select
    Selection.AutoFill Destination:=Range("T3:T" & lastRow)
'end power calc
'normalization2
```

```

    Range("U2").Select
    ActiveCell.Formula = "NormP2"
    Range("U3").Select
    ActiveCell.FormulaR1C1 = _
"= STANDARDIZE(RC[-1],AVERAGE(R3C20:R21002C20),
STDEV(R3C20:R21002C20))"
    Range("U3").Select
    Selection.AutoFill Destination:=Range("U3:U" & lastRow), ...
       Type:=xlFillDefault

    'Reactive calculation2

    Range("V2").Select
    ActiveCell.Formula = "Q2"
    Range("V3").Select
    ActiveCell.Formula = "=F3*J3*(180/PI())*SIN(RADIANS(G3-K3))"
    Range("V3").Select
    Selection.AutoFill Destination:=Range("V3:V" & lastRow)
'end reactive calc
'normalization reactive2

    Range("W2").Select
    ActiveCell.Formula = "NormQ2"
    Range("W3").Select
    ActiveCell.FormulaR1C1 = _
        "= STANDARDIZE(RC[-1],AVERAGE(R3C22:R21002C22),
        STDEV(R3C22:R21002C22))"
    Range("W3").Select
    Selection.AutoFill Destination:=Range("W3:W" & lastRow), ...
       Type:=xlFillDefault
  '________________________________________
    'Power calculation3

    Range("X2").Select
    ActiveCell.Formula = "P3"
    Range("X3").Select
    ActiveCell.Formula = "=F3*L3*(180/PI())*COS(RADIANS(G3-M3))"
    Range("X3").Select
```

```
    Selection.AutoFill Destination:=Range("X3:X" & lastRow)
'end power calc
'normalization3

    Range("Y2").Select
    ActiveCell.Formula = "NormP3"
    Range("Y3").Select
    ActiveCell.FormulaR1C1 = _
"= STANDARDIZE(RC[-1],AVERAGE(R3C24:R21002C24),
STDEV(R3C24:R21002C24))"
    Range("Y3").Select
    Selection.AutoFill Destination:=Range("Y3:Y" & lastRow), ...
        Type:=xlFillDefault

    'Reactive calculation3

    Range("Z2").Select
    ActiveCell.Formula = "Q3"
    Range("Z3").Select
    ActiveCell.Formula = "=F3*L3*(180/PI())*SIN(RADIANS(G3-M3))"
    Range("Z3").Select
    Selection.AutoFill Destination:=Range("Z3:Z" & lastRow)
'end reactive calc
'normalization reactive3

    Range("AA2").Select
    ActiveCell.Formula = "NormQ3"
    Range("AA3").Select
    ActiveCell.FormulaR1C1 = _
      "= STANDARDIZE(RC[-1],AVERAGE(R3C26:R21002C26),
      STDEV(R3C26:R21002C26))"
    Range("AA3").Select
    Selection.AutoFill Destination:=Range("AA3:AA" & lastRow), ...
        Type:=xlFillDefault

'_____________________________
 'Power calculation4

    Range("AB2").Select
```

```
    ActiveCell.Formula = "P4"
    Range("AB3").Select
    ActiveCell.Formula = "=F3*N3*(180/PI())*COS(RADIANS(G3-O3))"
    Range("AB3").Select
    Selection.AutoFill Destination:=Range("AB3:AB" & lastRow)
'end power calc
'normalization4

    Range("AC2").Select
    ActiveCell.Formula = "NormP4"
    Range("AC3").Select
    ActiveCell.FormulaR1C1 = _
    "= STANDARDIZE(RC[-1],AVERAGE(R3C28:R21002C28),
    STDEV(R3C28:R21002C28))"
    Range("AC3").Select
    Selection.AutoFill Destination:=Range("AC3:AC" & lastRow), ...
        Type:=xlFillDefault

    'Reactive calculation4

    Range("AD2").Select
    ActiveCell.Formula = "Q4"
    Range("AD3").Select
    ActiveCell.Formula = "=F3*N3*(180/PI())*SIN(RADIANS(G3-O3))"
    Range("AD3").Select
    Selection.AutoFill Destination:=Range("AD3:AD" & lastRow)
'end reactive calc
'normalization reactive4

    Range("AE2").Select
    ActiveCell.Formula = "NormQ4"
    Range("AE3").Select
    ActiveCell.FormulaR1C1 = _
   "= STANDARDIZE(RC[-1],AVERAGE(R3C30:R21002C30),
   STDEV(R3C30:R21002C30))"
    Range("AE3").Select
    Selection.AutoFill Destination:=Range("AE3:AE" & lastRow), ...
        Type:=xlFillDefault
End Sub
```

```
=STANDARDIZE(D3,AVERAGE($D$3:$D$21002),STDEV($D$3:$D$21002))
=STANDARDIZE(H3,AVERAGE($H$3:$H$21002),STDEV($H$3:$H$21002))
```

F - LSTM Machine Learning MATLAB Code

The Matlab Code in this section presents the machine learning algorithm, LSTM. The machine is trained on historical events which are labelled accordingly. The algorithm then predicts the outcome on new unseen events.

```
Traindata = readtable('Train20SecsLSTM.xlsx');% read train data
%Traindata{:,end} =  fillmissing(Traindata{:,end}, 'previous');
Testdata = readtable('Testdata20Secs.xlsx'); %read test data
batchSize = 1000;  %windowlength
batchSize2 = 1000;
groupIdx = 1:batchSize:height(Traindata);  %groupIdx stores ...
    the number of eventsin terms of batches
groupIdtest = 1:batchSize2:height(Testdata);

% Determine how many batches of each event are recorded in data:
numEvent0Batches = sum(Traindata{:,end} == 0)/batchSize;
numEvent1Batches = sum(Traindata{:,end} == 1)/batchSize;

numCellRows = numEvent0Batches + numEvent1Batches; %total ...
    events recorded as rows
% Create a cell array for predictors and response
xdata = cell(numCellRows,1); % predictors or Features
ydata = cell(numCellRows,1); % response

xtest = cell(numel(groupIdtest),1);
ytest = cell(numel(groupIdtest),1);

% Populate cell arrays
for i = 1:numel(groupIdx)

    batch = Traindata{groupIdx(i):groupIdx(i)+batchSize-1, ...
        1:end-1}';
    event = ...
        categorical(Traindata{groupIdx(i):groupIdx(i)+batchSize-1,end}');

    xdata{i} = batch;
```

```
    ydata{i} = event;

end

for j = 1:numel(groupIdtest)
   batch2 = ...
       Testdata{groupIdtest(j):groupIdtest(j)+batchSize2−1,1:end−1}';% ...
       the 2 is just because the last batch size 999 and not ...
       1000 rows
   xtest{j} = batch2;
   eventRes = ...
       categorical(Testdata{groupIdtest(j):groupIdtest(j)+batchSize−1,end}');
   ytest{j} = eventRes;

end

% Define LSTM architecture
numFeatures = 36; % Sequence size (4 features per event − each ...
    one is represented in a row in the data)
numHiddenUnits = 20; % Something you need to play around with
numClasses = 2; % 0 or 1

layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits,'OutputMode','sequence') % Input ...
        gate/tanh layer (calculation of c bar)
    fullyConnectedLayer(numClasses) % C_t
    softmaxLayer
    classificationLayer];

% Specify Training Options:

options = trainingOptions('adam', ...
    'MaxEpochs',80, ...
    'GradientThreshold',2, ...
    'InitialLearnRate',0.005, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',200, ...
    'LearnRateDropFactor',0.2, ...
```

```
    'Verbose',0, ...% Logical to display training data in the ...
        command window
    'Plots','training-progress'); %To plot training progress ...
        during training

% Train the LSTM:
net = trainNetwork(xdata,ydata,layers,options);

% Make Predictions with LSTM:
YPred = classify(net,xtest); % Need to split the dat
%acc = setdiff(YPred,ytest);

%acc = sum(YPred == ytest)./numel(ytest);

%  for i = 1:length(YPred)
%      figure(i)
%   cm = confusionchart(ytest{i},YPred{i},'title','Training ...
    Accuracy');
%   cm.ColumnSummary = 'column-normalized';
%   cm.RowSummary = 'row-normalized';
%  end
tot =0;

  for i = 1:length(YPred)
    % figure(i)

      acc = sum(YPred{i} == ytest{i})./numel(ytest{i});
      tot = tot+acc;

  end
  accuracy = (tot/5)*100
  bar(accuracy)
```

G - Eigenvalue estimation for train and test disturbances

This section depicts the plots that were derived from the Matrix pencil estimation method for all the train and test events utilised in the study. The plots include five test events and 19 events that were used for training.

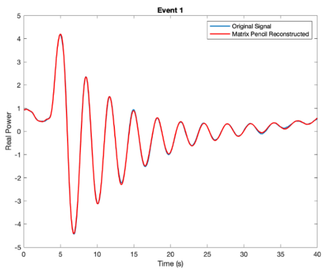

Figure G.1: Test 1 Disturbance

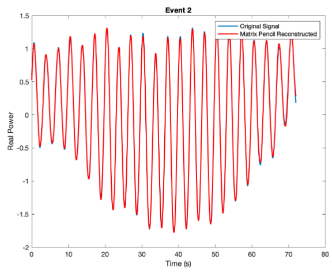

Figure G.2: Test 2 Disturbance

	sigma	alpha percent	frequency	amp	theta
Test 1	-0.0953262	4.82573814	0.31402355	2.25413728	-1.2433434

	sigma	alpha percent	frequency	amp	theta
Test 2	0.02387626	-1.2545833	0.30286761	0.34687041	-1.2008215

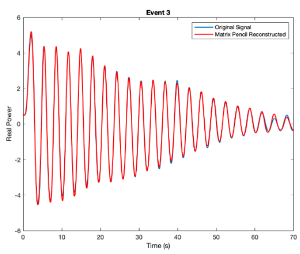

Figure G.3: Test 3 Disturbance

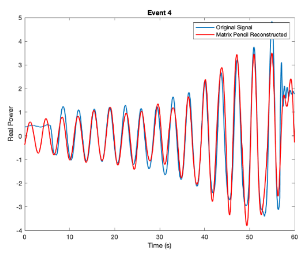

Figure G.4: Test 4 Disturbance

	sigma	alpha percent	frequency	amp	theta
Test 3	0.03107238	-1.6507461	0.2995402	0.16270163	0.39109909

	sigma	alpha percent	frequency	amp	theta
Test 4	-0.0229247	1.14497297	0.31863971	2.71899912	-1.4259355

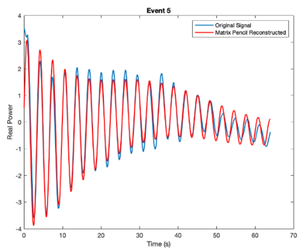

Figure G.5: Test 5 Disturbance

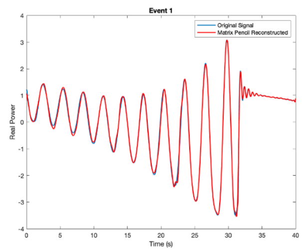

Figure G.6: Train 1 Disturbance

	sigma	alpha percent	frequency	amp	theta
Test 5	-0.0637904	3.04641904	0.33310722	0.72864193	-0.0550908

	sigma	alpha percent	frequency	amp	theta
Event 1	0.15231869	-7.6628495	0.31543087	0.01559711	0.36652894

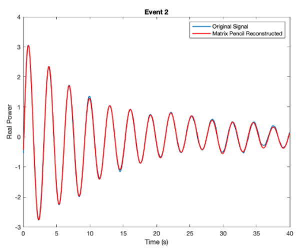

Figure G.7: Train 2 Disturbance

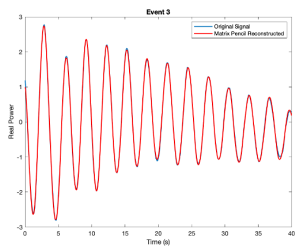

Figure G.8: Train 3 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 2	-0.0778482	3.77952044	0.32758313	1.6129822	1.54765036

	sigma	alpha percent	frequency	amp	theta
Event 3	-0.0326528	1.58813655	0.32718813	1.42663614	0.02891168

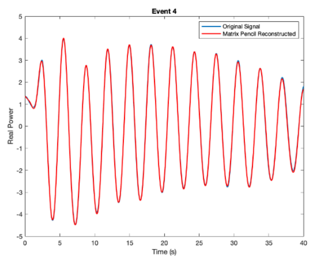

Figure G.9: Train 4 Disturbance

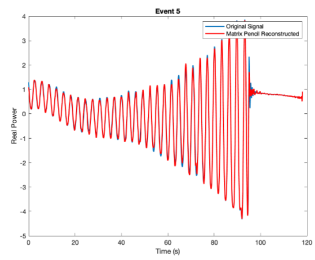

Figure G.10: Train 5 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 4	-0.0139032	0.69321201	0.31919775	2.09342789	1.44623702

	sigma	alpha percent	frequency	amp	theta
Event 5	0.00851186	-0.4224061	0.32070865	0.28809931	1.00780417

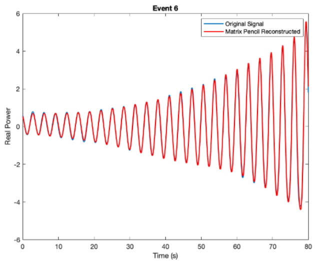

Figure G.11: Train 6 Disturbance

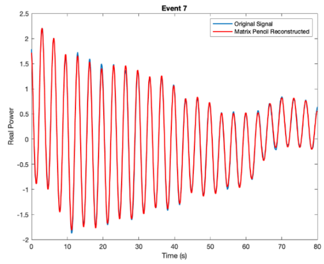

Figure G.12: Train 7 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 6	0.13335993	-6.9248674	0.30576673	9.47E-05	1.31524571

	sigma	alpha percent	frequency	amp	theta
Event 7	0.05815133	-2.9758684	0.31086634	0.09618018	-1.1627389

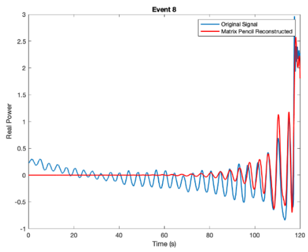

Figure G.13: Train 8 Disturbance

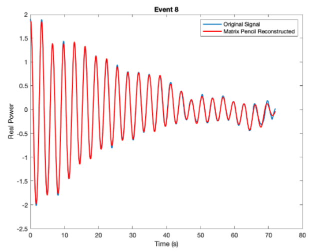

Figure G.14: Train 9 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 8	0.24305683	-12.508098	0.30684038	0.00274789	0.71913364

	sigma	alpha percent	frequency	amp	theta
Event 9	-0.0098715	0.46143706	0.34047641	1.41267856	-0.5469873

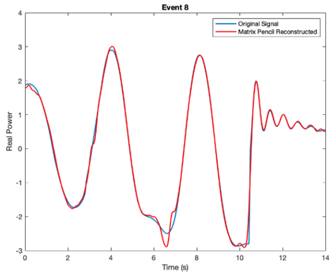

Figure G.15: Train 10 Disturbance

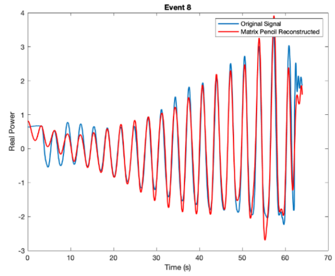

Figure G.16: Train 11 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 10	-0.0205944	0.97220953	0.33712412	1.34000746	1.08521376

	sigma	alpha percent	frequency	amp	theta
Event 11	0.02984964	-1.5126445	0.31403114	0.24291435	0.72328664

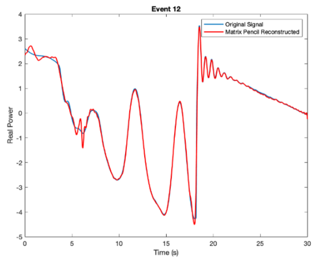

Figure G.17: Train 12 Disturbance

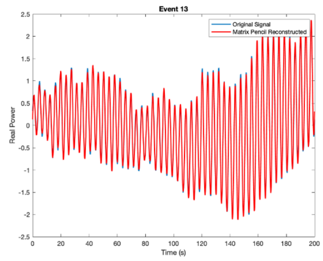

Figure G.18: Train 13 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 12	-0.0161708	0.86574753	0.29726546	1.16420821	1.07876581

	sigma	alpha percent	frequency	amp	theta
Event 13	-0.0872349	4.71959118	0.29384735	0.29262423	1.2643155

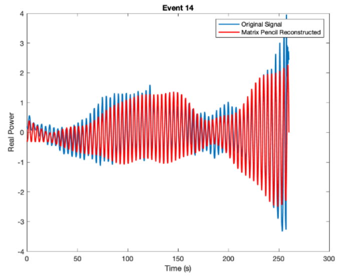

Figure G.19: Train 14 Disturbance

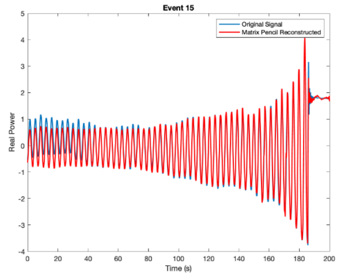

Figure G.20: Train 15 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 14	0.06133573	-3.8714242	0.25196325	0.93657435	-0.1950584

	sigma	alpha percent	frequency	amp	theta
Event 15	0.0198113	-0.8908257	0.35393473	0.00334003	0.3515

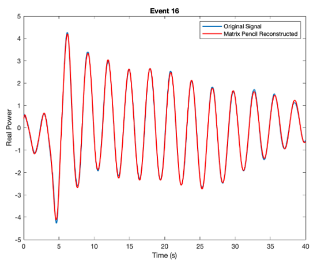

Figure G.21: Train 16 Disturbance

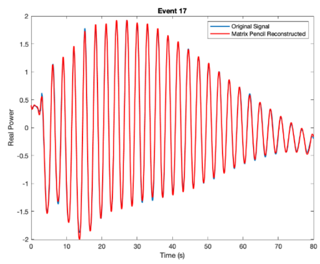

Figure G.22: Train 17 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 16	0.02877055	-1.5276359	0.29970762	0.01820833	0.04738624

	sigma	alpha percent	frequency	amp	theta
Event 17	0.00473411	-0.2694419	0.2796349	0.31568855	0.41726421

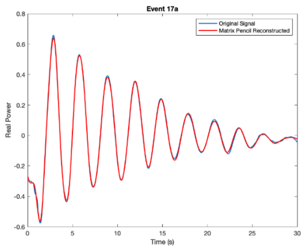

Figure G.23: Train 18 Disturbance

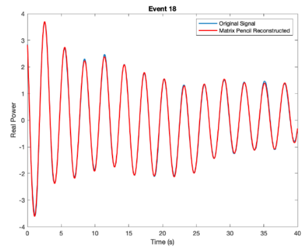

Figure G.24: Train 19 Disturbance

	sigma	alpha percent	frequency	amp	theta
Event 18	-0.1750212	8.61463748	0.3221487	0.67361868	1.32669609

	sigma	alpha percent	frequency	amp	theta
Event 19	-0.1251338	6.31642022	0.31467017	1.71868069	0.88424588

The Matrix Pencil method showed accurate estimation with most of the events depicted. The generated estimated eigenvalues from these plot were used in the eigenvalue position prediction.