Power Quality State Estimation in Distribution Systems Based on Neural Networks

Master's Thesis
Master of Science in the study program *Technische Informatik*
Faculty of Information, Media and Electrical Engineering
Cologne University of Applied Sciences (TH-Köln)

Submitted by:	Markus de Koster
Address:	markus.de_koster@th-koeln.de
Submitted to:	Prof. Dr. Eberhard Waffenschmidt
	Prof. Dr. Beate Rhein

Cologne, October 30, 2023

Technology
Arts Sciences
TH Köln

Abstract

This thesis explores the suitability of physics-aware neural networks for power quality state estimation in distribution grids. For that, power quality data was generated in a simulated environment and used for training and evaluation of different neural network models. Comprehensive data analyses were carried out, focusing on optimal representations and processing of power quality data for neural network applications. Comparative assessments with traditional fully connected architectures demonstrated the superior capabilities of physics-aware models, which utilize the physical grid structure as a regularization mechanism. Despite increased computational complexity, effective methods were identified to address challenges posed by deep architectures and sparse connectivity. Best performing models achieved a mean squared error loss of 4×10^{-6}, significantly outperforming traditional models with a loss of 1.1×10^{-5}. The results strongly indicate that physics-aware neural networks are suitable for power quality state estimation tasks. Promising avenues for expanding developed models include incorporation of physical laws in the learning process, potentially further constraining the network to physically possible states.

Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	Meaning
AC	Alternating Current
BNN	Bilinear Neural Network
CDF	Cumulative Distribution Function
CVNN	Complex-Valued Neural Network
CSV	Comma Separated Value
DNN	Dense Neural Network
DSO	Distribution System Operator
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
FNN	Feedforward Neural Network
GNN	Graph Neural Network
GCNN	Graph Convolutional Neural Network
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
KVL	Kirchhoff's Voltage Law
LV	Low Voltage
MAE	Mean Average Error
MAPE	Mean Absolute Percentage Error
MC	Monte Carlo
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
MV	Medium Voltage
PANN	Physics-Aware Neural Network
PDE	Partial Differential Equation
PDF	Probability Density Function
PINN	Physics-Informed Neural Network
PMU	Phasor Measurement Unit
PQ	Power Quality
PV	Photovoltaic
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RMS	Root Mean Squared
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RVNN	Real-Valued Neural Network
THD	Total Harmonic Distortion

1 Introduction

Power grids are currently undergoing a major change in both power production and consumption. While in the past power production was conducted centralized in large power plants such as coal and nuclear, nowadays, a shift towards decentralized production has begun. Small-scale photovoltaic (PV) systems are being integrated into low voltage (LV) and wind turbines into medium voltage (MV) distribution grids, while nuclear and coal plants are decommissioned. At the same time, the ratio of power electronics with non-linear loads is increasing with the prevalence of LED lights, computers and the proliferation of electric vehicle supply equipment (EVSE).

A well-known challenge caused by this shift is matching power production to consumption due to altered demand patterns such as charging of electric vehicles and the dependency on weather for renewable energy production.

However, new challenges arise in terms of power quality (PQ), influencing the safe operation of power grids. Here, deviations in voltage magnitude, in frequency and from a perfect sinusoidal wave may cause failure of electrical equipment, reduce longevity of grid components or reduce system efficiency. Therefore, monitoring power quality throughout the whole grid is crucial for a safe operation. However, installing measurement devices is costly and not possible at all locations, especially in residential areas with underground lines. Consequently, the demand for algorithms that are capable of calculating or estimating power quality at unmonitored nodes arises.

Several model-based algorithms exist that are typically unfit in cases of data scarceness caused by low measurement penetration in the power grid. On the other hand, data-driven algorithms make use of historical or simulated data to compensate for the lack of measurements at operation time. This approach will be implemented in this work by using neural networks to estimated voltages at unmonitored nodes.

At first, a deeper understanding of power quality data is developed. For this, section 2 details power quality including power quality errors, their causes, mathematical modelling and regulations. Afterwards, section 3 deals with the technical foundations required for this work. There, required knowledge of power grids, algorithms for state estimation, power grid modelling tools and basics of machine learning are conveyed. In section 4 related work and existing methods for state estimation are described. Then, in section 5 a solution concept is developed tackling the previously described issues. Here, different representations of the complex voltages are evaluated to identify the optimal input form for neural networks. Additionally, several neural network models are implemented and evaluated. While a basic dense neural network model serves as the benchmark for performance comparison, physics-aware neural networks are implemented that incorporate the underlying power grid structure into the neural network structure. Moreover, models that utilize physical laws are evaluated for their applicability to power quality issues. Finally, the implemented models are improved in their ability to generalize and to deal with possible measurement noise. This section is followed by a portrayal of the implemented system architecture of the simulation and physics-informed framework in section 6. Finally, the

results are presented in section 7 and a summary and outlook on future work are provided in section 8.1.

Out of scope is the development of complex-valued neural networks that require a full rewrite of the basis of neural networks, and graph neural networks that were developed in a different thesis at the same time. Also, state estimation is performed based on frequency-domain data. Because of that, some errors such as short-lived transients may not be fully captured here.

2 Power Quality

Power quality refers to the degree to which electrical power conforms to the desired characteristics. Those include a sinusoidal wave form of voltage and current, a consistent voltage and stable frequency. Maintaining a stable and consistent power grid is crucial for the save operation of electrical equipment that is designed to function within specific voltage and frequency limits. Electrical equipment critical to grid operations, such as transformers or capacitors may experience reduced performance, overheating, or complete failure if subjected to poor power quality. In the worst case, such incidents could escalate to system outages. Simultaneously, customer devices, such as household appliances, EVSE, and industrial machines, are also susceptible to poor power quality conditions.

In this section, power quality disturbances, their causes, consequences and regulations are described.

2.1 Power Quality Disturbances

In most parts of the world and also in this work's project the power grid is operated at a frequency of 50 Hz and the low-voltage (LV) distribution section works with a line-to-neutral root-mean-square (RMS) value of 230 V with RMS calculated as $V_{\mathrm{RMS}} = \frac{V_{\mathrm{peak}}}{\sqrt{2}}$. Without disturbances, the voltage forms a perfect sine wave at the given frequency and voltage amplitude. Any deviation from this perfect sine wave may be classified as power quality disturbance.

Disturbances can be caused by a variety of electrical equipment and system states. Large power plants using turbines are closely monitored and controlled generally producing a near-sinusoidal wave with consistent voltage and controllable frequency. On the other hand, non-synchronous generators such as PV systems with inverters that operate in a grid-following pattern may result in deviations from the desired frequency. In these types of inverters, the voltage magnitude and phase angle of the power grid are tracked, and the output is then synchronized. Non-linear loads such as LED lights, EVSE or computers may cause harmonics. Additionally, lightning strikes and switching events can result in transients.

Because of that, the power grid and electrical equipment is built with sufficient leeway defined through norms and regulations.

2.1.1 Steady-State Frequency and Voltage Deviations

If the RMS value or the frequency in a steady state of the system, i.e. when they do not change over time, differ from the expected value, this is classified as steady-state frequency or voltage deviation. While most electronics are designed with margins such that minor disturbances do not cause equipment failure, major disturbances may result in outages. Typically, these margins are within the $\pm 10\%$ range for voltages and most equipment is usually not effected heavily by frequency deviations [1]. However, regulations still exist that define frequency deviation limits as shown in table 1 separated into islanded and grid-connected mode for low-voltage distribution grids.

Grid-Connected Operation			Islanded Operation		
Minimum	**Maximum**	**duration**	**Minimum**	**Maximum**	**duration**
49.5 Hz	50.5 Hz	during 99.5% of the year	49 Hz	51 Hz	during 95% of the week
47 Hz	52 Hz	permanently	42.5 Hz	57.5 Hz	permanently

Table 1 Maximum allowed frequency deviations. Adapted from [2, EN 50160]

2.1.2 Harmonics

In this work harmonic analysis is performed using neural networks. Therefore, it is paramount to gain an understanding of what harmonics and their causes are. Harmonics are described as sinusoidal waves at frequencies that are integer multiples of the fundamental frequency [1]. Figure 1 shows a sine wave at 50 Hz with amplitude 1, one at 150 Hz with amplitude 0.3, a third at 250 Hz with amplitude 0.2 and the waveform that results from overlaying the former. On the bottom, the corresponding Fourier transform is shown. Using Fourier transform, a time-domain signal can be converted into its constituent frequencies. Hereby, a complex signal with multiple distortions can be simplified by looking at its spectral components. This method will be used multiple times throughout this work.

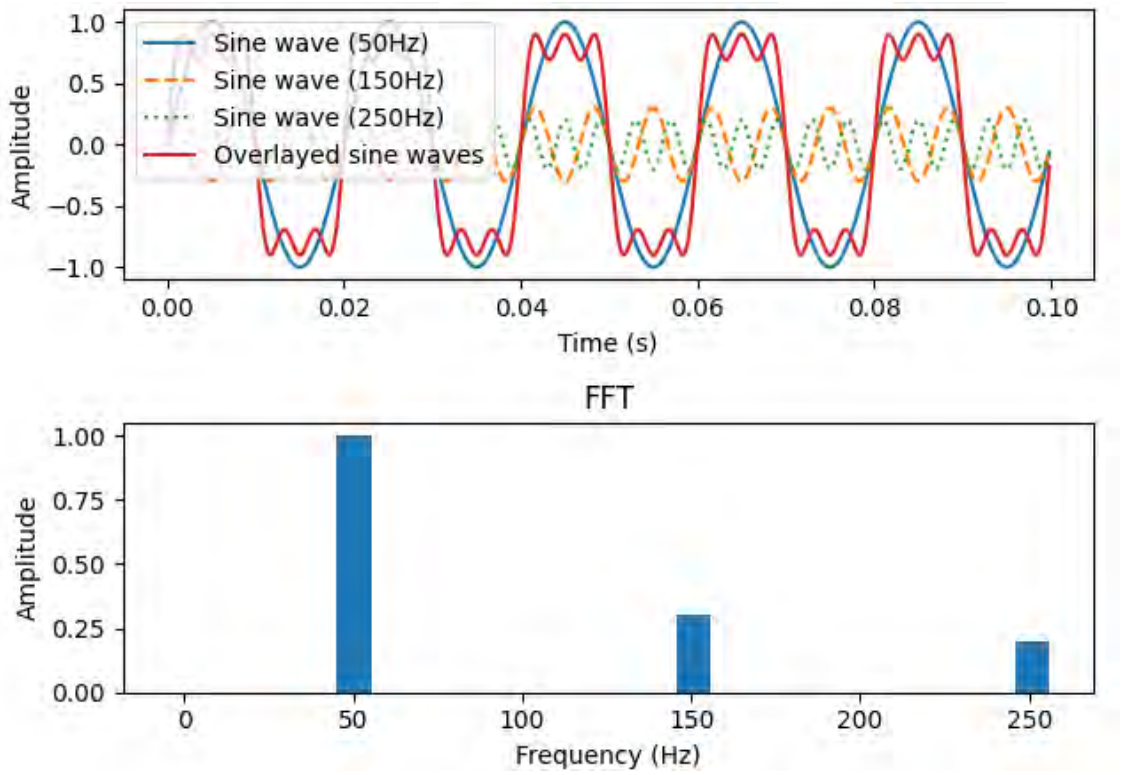


Figure 1 Overlay of sine waves at 50 Hz, 150 Hz and 250 Hz. Source: own illustration

A single perfect sine wave consists solely of one component at its fundamental frequency. In contrast, overlaying multiple sine waves generates a composite waveform with components at each of their respective frequencies, each contributing to the mixed signal's spectrum

with amplitudes corresponding to their original amplitudes.

In the case of a square wave as depicted in figure 2 harmonics at all uneven multiples of the fundamental frequency exist. This means that the signal consists of sine wave components at exactly those frequencies. Finally, an unbalanced waveform that is unsymmetrical

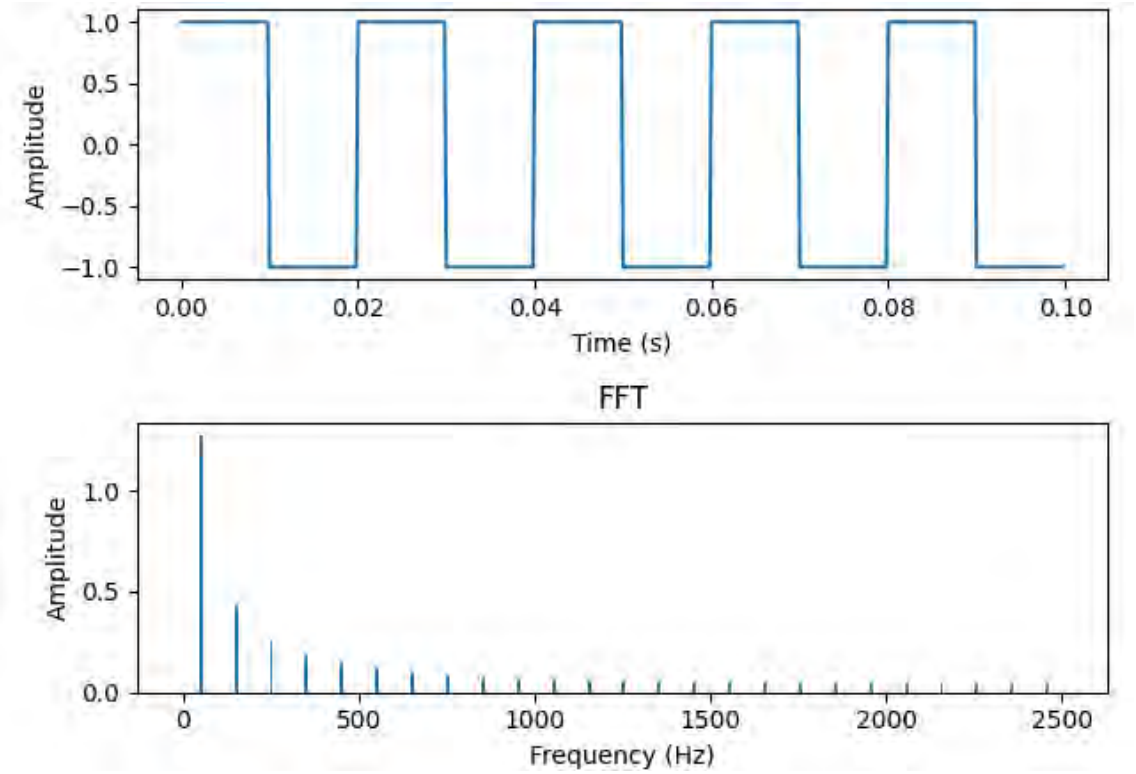


Figure 2 Square waves are built from sine waves at uneven harmonics as shown in the FFT spectrum. Source: own illustration

across the y-axis as shown in figure 3 results in even harmonics. In theory, the voltage wave is not expected to be unbalanced and therefore even harmonics are expected to be lower in amplitude than uneven harmonics. Additionally, interharmonics with components at frequencies that aren't integer multiples of the fundamental frequency and subharmonics with those frequency components below the fundamental frequency exist but will not be a part of this work. Finally, superharmonics shall be mentioned as a term that is commonly used for very high frequencies or *overtones*. Usually, harmonics make up most of the spectrum that builds the voltage wave and thus interharmonics, sub- and superharmonics play an inferior role in state estimation.

Total harmonic distortion (THD) is commonly used to describe the overall percentage of harmonics relative to the fundamental frequency. For that, the derivative of the sum of the squared harmonics is calculated as shown in equation 1 where harmonics up to N are considered and V_n is the RMS value of the n-th harmonic. Then, the result is divided by the RMS value of the fundamental frequency.

$$\mathrm{THD} = \frac{\sqrt{\sum_{n=2}^{N}(V_n^2)}}{V_1} \tag{1}$$

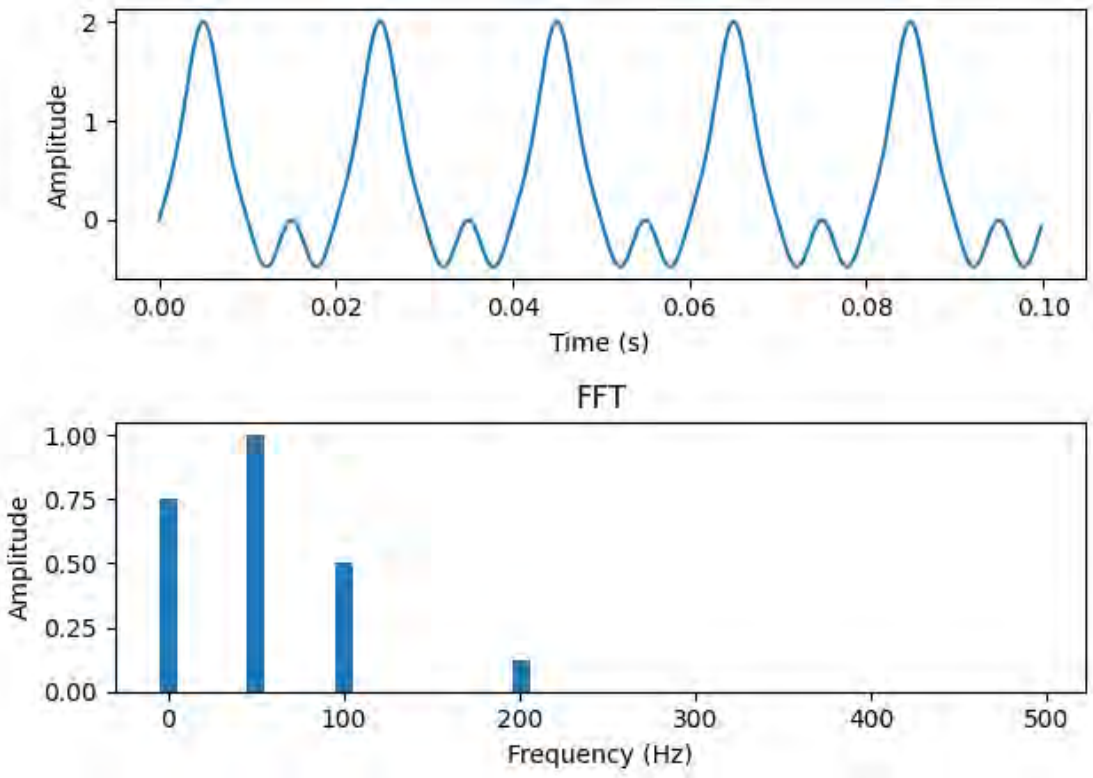


Figure 3 FFT of an unbalanced waveform with even harmonics visible in the FFT spectrum. Source: own illustration

A more detailed view on harmonics is presented in section 5.1.2 where harmonic spectra are measured in a laboratory setting.

International standards govern the harmonic current emissions of electrical equipment, with the International Electrotechnical Commission (IEC) issuing guidelines in the form of the 61000 series. Specifically, parts 3-2 and 3-12 outline limits for harmonic current emissions of equipment with input currents of $\leq 16\,\mathrm{A}$ and $> 16\,\mathrm{A}$, $\leq 75\,\mathrm{A}$ per phase, respectively [3] [4]. Concurrently, regulations exist for managing harmonic distortions in the power grid. Within Europe, the European Committee for Standardization has issued standards addressing voltage characteristics [2]. Table 2 gives the maximum allowed harmonic distortions for harmonics up to the 25th harmonic. For harmonics above the 25th and for interharmonics no limit is yet defined.

For three-phase AC systems, the phase angle of harmonics is calculated using equation 2 where n is the n-th harmonic, ϕ is the phase angle of the fundamental frequency and A, B, C represent the phases.

$$\phi_{n,A} = n\phi \mod 360^\circ \tag{2a}$$

$$\phi_{n,B} = n(\phi - 120^\circ) \mod 360^\circ \tag{2b}$$

$$\phi_{n,C} = n(\phi - 240^\circ) \mod 360^\circ \tag{2c}$$

One can see that if $n = 3$ the displacement between phases, i.e. 120° for phase B and 240° for phase C becomes a multiple of 360°. In figure 4 the alignment of phase angles

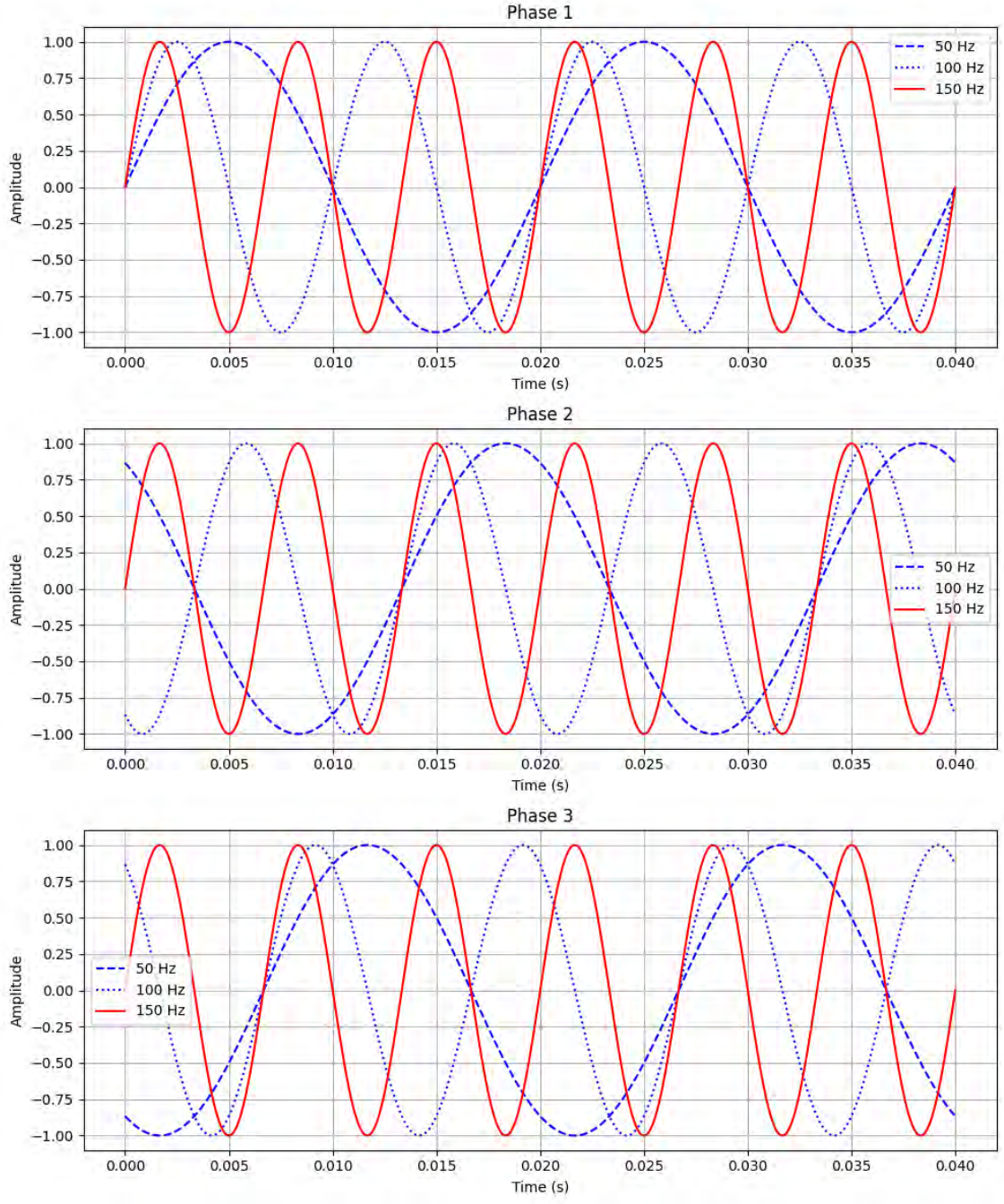


Figure 4 Harmonic phase alignment for harmonic at multiples of three of the fundamental frequency; third harmonics in red. Source: own illustration

Uneven harmonics		Multiples of 3		Even harmonics	
order	**relative amplitude**	**order**	**relative amplitude**	**order**	**relative amplitude**
5	6%	3	5%	2	2%
7	5%	9	1.5%	2	1%
11	3.5%	15	1.0%	6 to 24	0.5%
13	3%	21	0.75%		
17	2%				
19	1.5%				
23	1.5%				
25	1.5%				

Table 2 Maximum allowed harmonic distortions at low voltage supply points as percentage of the RMS value for fundamental frequency. Adapted from [2, EN 50160]

at multiples of three can be observed which is caused by phase displacement described in equation 2. While at 50 Hz all phases add up to zero, at 150 Hz all phases align in phase angle and add on top of each other.

2.1.3 Effective Value Fluctuations

Effective value fluctuations can be categorized into voltage sags and voltage swells. Here, the voltage amplitude decreases or increases for multiple cycles. As shown in figure 5

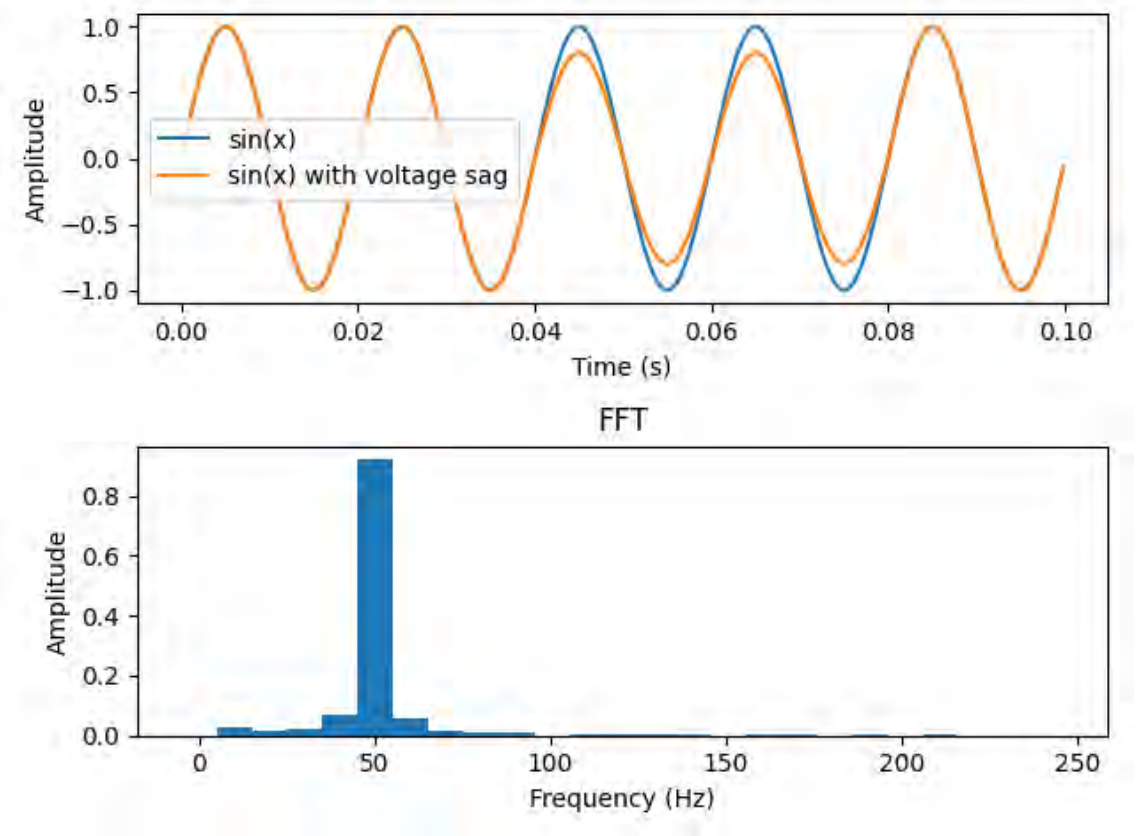


Figure 5 Voltage sag and associated FFT spectrum. Source: own illustration

voltage sags can mostly be identified through the altering amplitude, however, also cause some form of harmonic distortion mostly in subharmonics and interharmonics due to the altered wave shape. In the European norm [2, EN 50160] voltage sags and swells are classified based on increase or decrease in amplitude and duration of the sag or swell.

2.1.4 Transients

Transients are short disturbances that are caused by sudden changes in the power system such as short-circuits. Firstly, impulsive transients are typically caused by events like lightning strikes. Typically, the effects of impulsive transients are very short-lived spikes with a quick return to a steady state.

Secondly, oscillatory transients often occur due to the energizing or de-energizing of inductive or capacitive loads, or due to switching operations like opening or closing of circuit breakers. For these transients, oscillation occurs around a specific value and the effect may last up to several cycles [5]. The spectral components of a transient depend on the

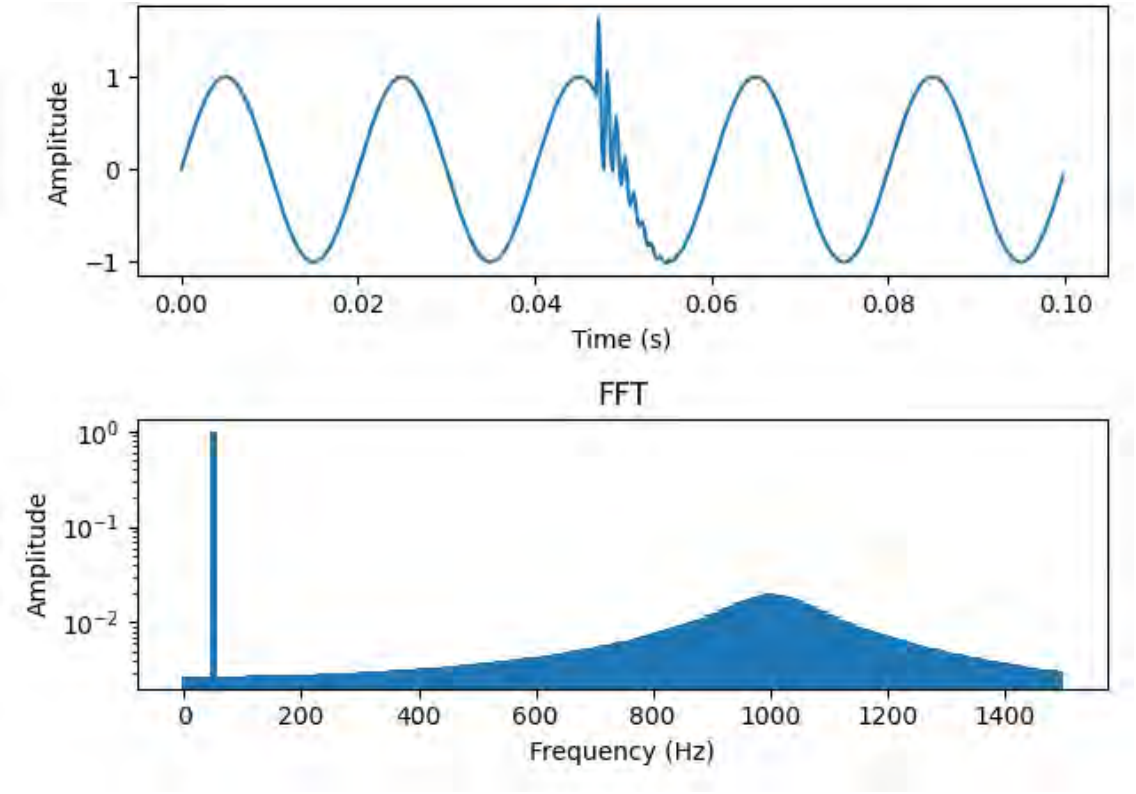


Figure 6 Simplified model of an oscillatory decaying transient with a main frequency component around 1 kHz with associated FFT on a logarithmic scale. Source: own illustration

transient source and may involve frequencies up to a few mega-Hertz [6]. A simplified mathematical model of an oscillatory transient with a main frequency component around 1 kHz is shown in figure 6.

2.1.5 Flicker

Contrary to the previously mentioned power quality disturbances, the term flicker describes the effect the disturbance can have on specific equipment, more precisely lighting systems

that change in brightness so that it is visible to the human eye. Here, the voltage fluctuates in a way that causes equipment to be affected so that visual disturbances can occur. However, flickers are not limited to lighting systems but rather described as a voltage drop with short fluctuations that may also affect other equipment. Flickers are mostly caused by large arc furnaces, electric motors or reactive power compensators [1]. Figure 7 shows

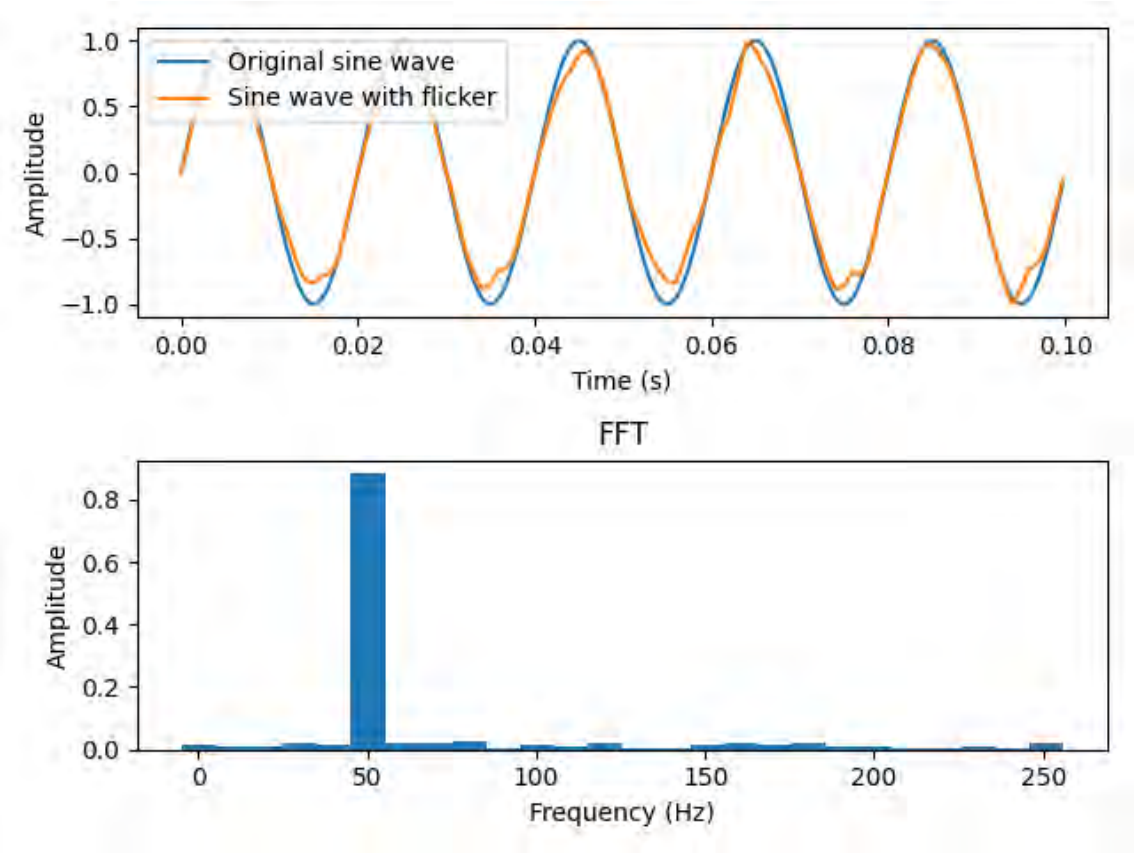


Figure 7 Simplified model of rapid voltage fluctuations that could cause flicker and associated FFT. Source: own illustration

the voltage waveform under rapid voltage fluctuations combined with an overall voltage sag that could cause flicker disturbances.

2.2 Propagation in the network

Power quality disturbances are not limited to the point of occurrence. Instead, they propagate through the network depending on the network topology, impedance and existence of transformers [7]. Hereby, equipment failures can occur even at nodes further away from the original point of failure. However, it also allows for estimating and locating the cause even if no measurement is nearby.

In section 3.1 the propagation of harmonics across delta-wye transformers will be described. The propagation of voltage sags between the primary and secondary side also depends on the type of transformer, characterized by its winding connections [8].

2.3 Monitoring

Power quality measurement devices assess multiple parameters including effective values, harmonics, phasors and waveform tracking. While waveform tracking constitutes only one component of these measurements, it is the main focus of this work. Here, the monitoring devices use an envelope at ±10% around the anticipated waveform for fault detection. If the envelope is breached the monitoring device is capable of saving a fault record to fully capture the event including the previous time steps for later cause detection and following time steps to monitor fault recovery. Figure 8 depicts an envelope breach caused by a voltage sag. As dotted lines, a 10% envelope around the undisturbed sine wave is shown that is breached multiple times by the disturbed wave.

However, some power quality disturbances may remain unnoticed if the envelope is not

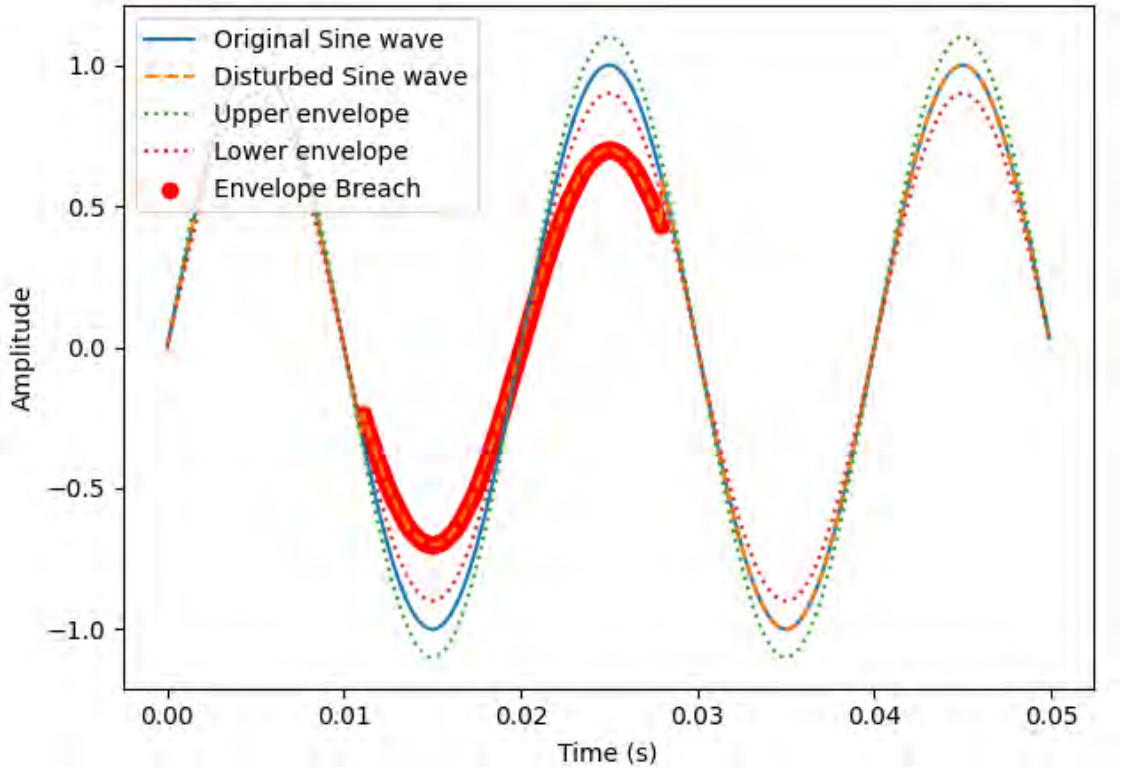


Figure 8 Envelope breach caused by a voltage sag. Source: own illustration

breached. Especially for low measurement penetration in a given grid, faults may remain unnoticed since the envelope is not breached at the closest monitoring point. Hereby, critical system states may cause equipment failure without the distribution system operator noticing and being able to intervene. Therefore, estimating power quality at unmonitored nodes as accurately as possible is crucial in a safe operation of the power grid.

3 Technical Foundations

In this section, the technical foundations are described on which the rest of this work is based on. At first, knowledge required to understand power grids is conveyed in section 3.1. Then, in section 3.2 the goal of state estimation and the differences between power flow and power quality state estimation are described. Finally, in section 3.4 those concepts of neural networks are introduced that will be modified in later chapters.

3.1 Power Grids

While the transmission grid is responsible for transmitting electricity through large distances, generally from large power plants to substations, the distribution grid distributes electricity to end users. In this work, the focus lies on medium and low voltage distribution grids that are currently undergoing changes caused by the shift towards decentralized renewable energy sources and non-linear consumers.

Those distribution grids consist of different types of components. Power lines can either be installed underground or overhead, consist of different materials and diameters, each associated with different parameters in terms of resistance, reactance and capacitance. Additionally, in AC circuits, two types of connections exist, delta and wye connection. While in delta connection, the three phases are connected in a triangle with each other, in wye connection, all three phases are connected together to a neutral wire.

Different types of transformers and connections exist. As seen earlier, third harmonics are phase aligned and would normally add up. Normally, this would result in the highest harmonic distortions. Because of that, delta-wye transformers are built in such a way that third harmonics are caught within the delta loop. In these cases, a short circuit current runs in the delta loop and ultimately gets converted to heat in the iron core of the transformer. This effect is shown in figure 9 with a primary delta winding and secondary wye winding and the short circuit current marked in red.

Small scale generators such as PV systems and wind turbines are directly connected to the distribution grid. Residential households and commercial buildings consume power in a mostly predictable pattern and are thus classified as static loads. On the other hand, industrial machines show variable consumption patters and are commonly classified as dynamic loads. Finally, measurement devices capture the power grid system state. In residential buildings, power consumption is monitored at slow intervals using traditional meters or nowadays more frequently smart meters. Additional measurement units can be installed by distribution system operators (DSO) at critical spots that require surveillance of additional parameters. Phasor measurement units (PMU) offer synchronized measurements of complex voltage and current phasors with frequent data transmission to DSOs. While traditionally more prevalent in transmission networks, these units are becoming increasingly relevant and widely deployed in distribution grids.

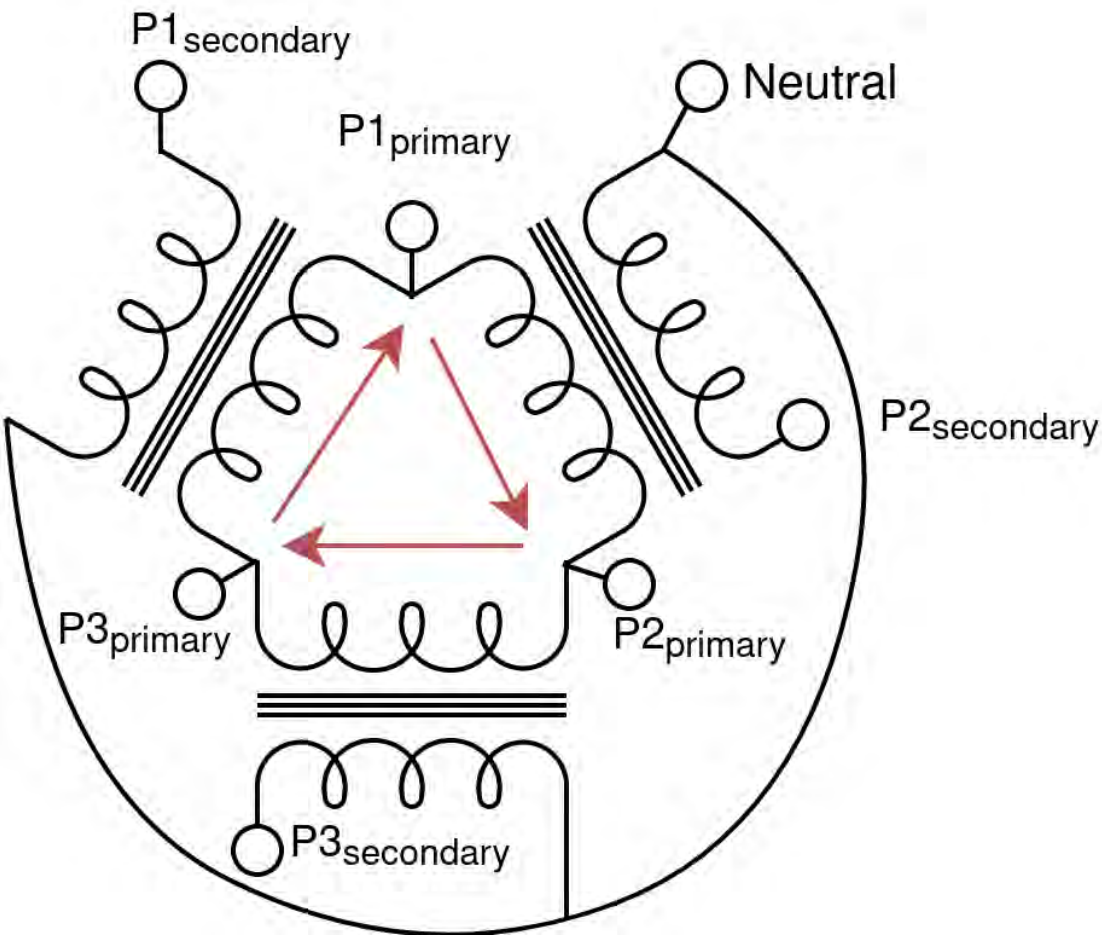


Figure 9 Delta-wye transformer with short circuit current for third harmonics shown in red. Source: adapted from [9]

3.1.1 Fundamental Laws and Equations

Electrical circuits including power grids can be described through equations and laws that will be utilized throughout this work. Ohm's law in complex AC circuits, shown in equation 3, describes the relationship between bus voltages V, injected currents I that are related through the nodal admittance matrix Y.

$$\underline{I} = \underline{Y} \cdot \underline{V} \tag{3}$$

Admittance, composed of conductance and susceptance, is the inverse of impedance, a measurement of resistance and reactance in AC circuits. Harmonic voltage disturbances, as described in the previous chapter 2.1.2, are caused by nonlinear loads that draw distorted current waveforms. These non-sinusoidal currents flow through the network's, i.e. power lines' or other components', impedance and thus generate harmonic voltages. For non-fundamental frequencies, Ohm's law also applies with voltages, currents and admittance calculated individually for the respective harmonic.

Kirchhoff's voltage law (KVL), which states that the sum of all voltages around any closed loop in a circuit must equal zero, serves as a foundational principle for deriving power balance equations. These equations establish the interrelations between power injections, nodal voltages, and the admittance matrix in an electrical network. The power balance equations are formulated as shown in equations 4 and 5 with P_i and Q_i as the real and reactive power injections at node i, respectively, V_i and θ_i as the voltage magnitude and angle at node i, respectively, Y_{ij} and α_{ij} as the magnitude and angle of the element in the i-th row and j-th column of the admittance matrix, respectively, and n as the number of nodes in the system.

$$P_i - V_i \sum_{j=1}^{n} V_j |Y_{ij}| \cos(\theta_i - \theta_j - \alpha_{ij}) = 0 \tag{4}$$

$$Q_i - V_i \sum_{j=1}^{n} V_j |Y_{ij}| \sin(\theta_i - \theta_j - \alpha_{ij}) = 0 \tag{5}$$

This principle will later be used in related work to confirm adherence of models to physical laws.

3.2 State Estimation

State estimation algorithms in power systems aim to achieve multiple objectives including accurate estimation of the current system state, fault detection, optimal resource utilization and ensuring safe operation of the power grid. System states of power grids that can not be fully observed are estimated using various algorithms. Even though this thesis focuses on power quality analysis, power flow estimation will be described first as it offers well-established algorithms and literature that, with some modification, can be applied to power quality analysis. Algorithms for state estimation can be coarsely categorized into model-based, forecasting-aided and data-driven algorithms [10]. Further, model-based and forecasting-aided state estimation can be divided into node-voltage-based and branch-current-based estimations depending on the used state vector [10]. An overview of existing

methods for power flow state estimation is given in table 3.

Model-based	**Forecasting-aided**	**Data-driven**
weighted least squares	Kalman filters	dense neural networks
least absolute value	extended Kalman filters	physics-aware neural networks
least trimmed squares	unscented Kalman filters	physics-informed neural networks
least median of squares		graph neural networks
generalized maximum likelihood		

Table 3 State estimation algorithms. Adapted from [10]

3.2.1 Power Flow Analysis

In general, traditional model-based power flow state estimation can be described by a mapping of measurements z to the state vector x upon which a measurement function h is called. Then the residuals r that results from subtracting the aforementioned result from the measurements z are calculated as shown in equation 6 [10].

$$r = z - h(x) \tag{6}$$

Afterwards, the residuals are minimized using a minimization or cost function. The difference between each of the abovementioned model-based algorithms lies within their respective cost functions. Notably, traditional model-based state estimations with weighted least squares (WLS) as the most commonly used algorithm require a fully known state vector. If measurements are not available for all nodes to build full vector, replacement values which are either virtual or pseudo measurements are used [11] [12]. For virtual measurements the voltage drop at a given node and hereby also the load at a specific node are assumed to be zero which is also known as zero-injection [13]. However, for pseudo measurements historical data is used to fill in gaps. As one can easily see, these algorithms suffer from numerical instability in case of lack of measurements. The aforementioned algorithms are well-suited for transmission system state estimation as transmission systems are usually well-observed and comprised of few unknowns. For distribution system state estimation data-driven algorithms are becoming more popular due to their robustness against missing data and computational speed post-training phase.

3.2.2 Power Quality Analysis

Power quality state estimation can be categorized into fundamental frequency, harmonic, transient and voltage sag or swell state estimation [14]. Fundamental frequency state estimation captures the system state at the fundamental frequency as described in the previous chapter. In harmonic state estimation, the system state at non-fundamental frequencies is estimated using limited available harmonic data, that might be corrupted with measurement noise. Model-based algorithms also exist for harmonic analysis, however are restricted by their inability to deal with underdetermined equation systems for equation 6

[15]. Voltage sag or swell state estimation attempts to estimate voltage dips and swells at unmonitored nodes from available measurements. Arguably, a voltage sag or swell state estimation is included in a fundamental frequency state estimation's voltage magnitudes. Moreover, if the fundamental frequency is added to the harmonic state estimation, a combination of these three state estimations can be performed at once. Thus, in this work, power quality state estimation will be performed on defined harmonics and the fundamental frequency at once.

Transients, although short-lived, are not entirely elusive in the frequency domain, particularly when high-frequency measurement devices are employed. At a sufficiently high sampling rate, the device is capable of capturing transient events effectively. Even though transients are short-lived, they significantly distort the voltage waveform. In cases, a measurement device is installed at the transient source or the propagation is large enough that an envelope breach is triggered at the next monitored node, GPS-synchronized measurements can be evoked to fully capture the event. However, as explained in section 2.1.4 spectral components of transients vary widely.

Furthermore, variable frequency drives may cause disturbances at switching frequencies typically in ranges between 1 kHz and 20 kHz and interharmonics may also be significant [16]. Hence, power quality analysis that focuses only specific multiples of the fundamental frequency in the range of 50 Hz to 1 kHz is insufficient for a full analysis of power quality. Nevertheless, multiple disturbances are visible in this range and for this work the above-mentioned range was chosen as a proof of concept and based on availability of harmonic spectra.

3.3 Open Distribution System Simulator (OpenDSS)

Open Distribution System Simulator (OpenDSS), developed by the Electric Power Research Institute (EPRI), is a simulation tool for electric power distribution systems. It combines various use cases including harmonic and interharmonic analysis in the frequency domain which are used in this work. Also, it works with both radial and meshed topologies for single-phase and multi-phase grids and was hence chosen as simulation framework for this work [17].

In OpenDSS a network description is represented by scripts that can be saved to and read from disk. For each grid component, its type, identifier and key parameters are defined in human-readable text form. Even though this allows easy modification of parameters by humans, manual recreation of power grids, especially for complicated architectures, is cumbersome and prone to errors. Hence, in this work a tool was expanded, that automatically converts a network defined in *pandapower*, a power system analysis tool introduced in [18], to OpenDSS. Previous work built for conversion of a specific network was expanded to a generalized converter application that will be presented in chapter 6.

For harmonic analysis, loads and generators are equipped with harmonic spectra whose specifics will be discussed in section 5.1.2. Then, a power flow calculation is performed as initialization for the harmonic analysis. Finally, OpenDSS solves the circuit for all fre-

quencies present in harmonic spectra or defined by the users. For each monitor defined in the OpenDSS script, the associated measured values are saved to a comma separated value (CSV) file. Since for each time step and for each monitor a separate CSV file is generated, many input-output operations are performed and the required disk space increases. Even though single generated files are comparatively small, given the vast amount the file system may run out of allocated disk space or out of allocated space for file identifiers. Without modification of the OpenDSS source code, it is unfortunately not possible to define different more data efficient file formats.

3.4 Neural Networks

In this work, multiple custom adaptions to neural networks are implemented. Because of that, a deep understanding of the mathematical background of feedforward neural networks is paramount. Figure 10 shows a simple architecture of a multi layer perceptron (MLP)

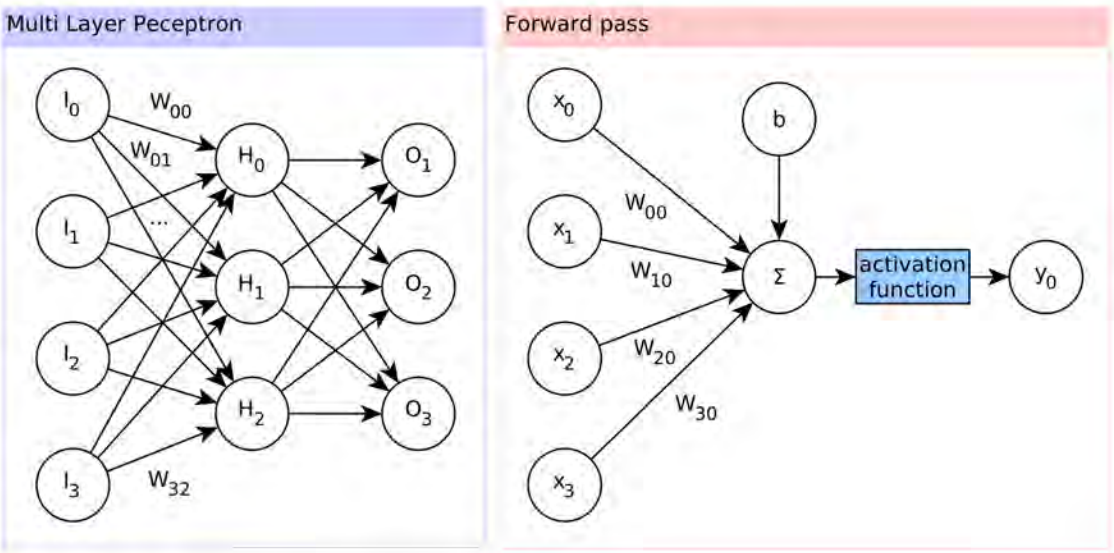


Figure 10 Simple multi layer perceptron architecture with input (I) hidden (H) and output (O) layer on the left side and forward pass for a single neuron activation on the right side. Source: own illustration

that consists of an input layer, a hidden layer and an output layer. The layers are fully connected with each connection associated with a weight. On the right side, a forward pass of a single neuron for a feedforward neural network is shown. Here, the output estimation y_0 of the neuron is determined by the input X_i, weights for each connection $W_{i,j}$ and a bias b. The weighted sum is calculated by multiplying the input activations x_i of the previous layer with the weights between node i and j as described in equation 7.

$$s_j = \sum_i x_i \cdot W_{i,j} \tag{7}$$

Then, the bias is added and the activation function f is applied to obtain the activations y_j as shown in equation 8.

$$y_j = f(s_j + b) \tag{8}$$

In section 5.3.1 a custom layer is introduced in which some connections are manually pruned by altering the weight and input matrix multiplications.

After the forwards pass is complete, the loss function $L(y, \hat{y})$ measures the difference between predicted output $\hat{y}$ and expected output y. In this project, a regression task is tackled and for that mean absolute error (MAE), mean squared error (MSE) are considered as loss functions with equations 9 and 10 respectively.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{12}$$

Further, many related papers calculate the loss as root mean squared error (RMSE) or mean absolute percentage error (MAPE) given in equation 11 and 12, respectively. In section 4.1.3 and 5.3.3 the usage of customized loss functions that incorporate physical information will be discussed.

After the loss computation, gradients are calculated and propagated in the backward pass. The internals of the backpropagation algorithm and gradient computation will not be altered in this thesis, are well-described in literature and thus will not be further elaborated on here [19].

During each iteration a batch of input data is passed to the network. Larger batch sizes can result in more stable gradients towards a local minimum at the cost of high memory usage and increased computational complexity. Due to the high dimensionality of the input data in this work's use case memory usage will become relevant in future sections.

4 Related Work

In this section, existing literature on topics regarding this work is reviewed. While section 4.1 describes state-of-the-art models and model architectures for power flow and power quality state estimation, section 4.2 reviews literature regarding usage of complex value representations in neural networks. Finally, in section 4.3 related domains are described that will not be part of this work, however need to be considered when integrating the system into a live environment or provide other interesting insights.

4.1 State Estimation Algorithms

While numerous papers exist on power flow state estimation, few focus on power quality in distribution grids. Hence, at first a few selected papers on power flow are presented with innovative approaches that tackle the lack of available measurements or incorporate physical information.

4.1.1 Model-Based and Forecasting-Aided Algorithms

For model-based algorithms, it is well-known that at least as many variables are known as unknown variables exist in the system equation 6. In [12] a WLS algorithm was used to show that under simplified circumstances reasonable results for worst case assumptions with model-based algorithms can be achieved as long as the ratio of unknown nodes to measurement nodes is not bigger than 2.5. While for power flow power consumption measurements may be available through smart meters installed in households, power quality analysis requires more sophisticated measurement devices that are capable of measuring harmonics and waveforms for all three phases. Therefore, in this work's use case a much larger ratio of unknown nodes to measurement nodes is to be expected and model-based algorithms are no longer feasible.

Forecasting-aided algorithms, comprised of Kalman filters and their extensions, utilize historical data by performing a weighted average of the predicted state and current measurements. To some degree, this usage of the predicted state enables them to compensate for missing measurements [10]. While Kalman filters can solve linear problems, their unscented and extended variations also solve non-linear problems [20]. However, they also suffer from system unobservability under severe lack of measurements, especially if measurements are unavailable for several time instances in a row. Numerous extensions exist for model-based and forecasting-aided algorithms. Measurements are weighted based on their expected accuracies with observed nodes receiving high weights in contrast to pseudo-measurements. Physical laws are incorporated by adding constraints to Kalman filters in [21] and to WLS in [22] and [23]. This, however, increases the computational complexity even further. Additionally, bounds for specific variables are used in [12] to incorporate information about consumers and producers into the equation. However, this poses a problem if the bounds are falsely set and unexpected system states remain unnoticed due to the ill-conditioned initialization of the algorithm.

4.1.2 Data-Driven Algorithms

In [24] a joint topology and power flow state estimation algorithm is proposed. At first, probability density functions (PDF) of power injections are calculated using historical smart meter data, which is assumed to be available at all nodes. Then, an approximate solution to three-phase power flow through random Monte Carlo (MC) sampling using the best-fit PDFs is performed. For all samples, the associated topologies are saved and used to train a DNN for topology identification. Finally, the state estimation DNN is trained using PMU measurements for the currently sampled topology. If during real-time operation the identified topology is not consistent with the trained topology, transfer learning is used to apply the model to the new topology. In the IEEE 34-node system with two installed PMU using a DNN an MAE of 0.13 for voltage phase and an MAPE of 0.26 for voltage magnitude was achieved. For the 240-node distribution network of Midwest U.S., with six installed PMU, an MAE of 0.15 for voltage phase and an MAPE of 0.25 for voltage magnitude was achieved.

4.1.3 Incorporating Physical Information

Several approaches exist that incorporate physical information into neural networks. In literature, Physics-Aware Neural Networks (PANN, sometimes PAWNN) and Physics-Informed Neural Networks (PINN) are described. Even though the nomenclature seems equivalent, the term PANN is usually used for networks incorporating physical information into the neural network topology while PINN utilize physical information in their loss function [25][26]. The authors of [27] and [28] propose a data-driven model that incorporates a system state formulation in the model's weights which are determined during the training phase, therefore calling it 'physics-aware'. More precisely, the pseudo-inverse of the model matrix which itself is derived from measurements is built and multiplied with weights. After the training, unknown voltages are computed from known active and reactive power flow. To compensate for potentially missing measurements a recurrent neural network (RNN) is used that generates pseudo-measurements for the next time step from the estimations of the previous one. Skip connections are added to avoid vanishing or exploding gradients and measurements are corrupted with Gaussian noise with standard deviation of $\sigma = 0.2$ for power flows and $\sigma = 0.1$ for voltage magnitudes. Then, the authors test the hybrid algorithm against the IEEE 57-bus and 118-bus system resulting in $RMSE = 2.97 \times 10^{-4}$. This approach is not feasible in this work's use case as insufficient measurements are available to fully capture the system state or to create meaningful pseudo measurements.

In [29] physics-aware neural networks are proposed that incorporate the graph structure of the neural network in the neural network architecture. Here, the connections between each of the layers are pruned by multiplying weights with either zero or one based on adjacency information. To reduce overfitting, the connections are pruned deterministically before training so that fewer trainable parameters exist. Further, an algorithm for μ-phasor-measuring units (μPMU) placement is developed, that separates the underlying

graph structure into compartments. At the locations of separation μPMUs are installed, and it is shown that the width of each compartment is equivalent to the minimum required layers in the PANN. Verification of the proposed model is conducted by placing five μPMU in the IEEE-37 bus reference network and measuring the squared L2-norm which is effectively equivalent to MSE. Additionally, seven real-valued measurements of current magnitudes and 26 complex pseudo measurements are included [29]. For optimal μPMU placement an MSE of 1.273×10^{-3} is achieved which increases significantly for suboptimal μPMU placement and in case too few layers are used.

The papers [30] and [31] build upon the previously mentioned work [29] by additionally pruning connections between those nodes that lay in different compartments. Here, training data for power flow estimation is generated using OpenDSS for the IEEE 123 bus feeder. Three scenarios are constructed with different measurement errors and penetration, and the performance is measured using "average estimation accuracy" [31]. Even though this term is neither defined in the quoted work nor a commonly used term for regression tasks, it is assumed to be the same definition as in [29] which makes it equivalent to MSE. In case of sufficient measurements with power measurements at 85 and PMUs at 2 of the total 123 nodes, a WLS algorithms, albeit slower, performed better than PANN. However, with fewer measurements and increased measurement noise, the WLS performed significantly worse while PANN performance remained similar. Further, the additional pruning did not improve results.

Physics-informed neural networks were first introduced in [25] as a novel approach to integrate the governing physical laws, described by general nonlinear partial differential equations (PDE), directly into the neural network training process. Here, the loss is defined as sum of the model loss and a physics-informed loss that arises from the discrepancy between the neural network's output and the expected behavior as defined by the associated PDE. The authors of [32] differentiate between physics-informed loss function, physics-informed designs of architecture, which for the context is of this thesis is termed physics-aware, and physics-informed initialization. According to their definition physics-informed initialization also comprises the initial training of neural networks with simulated data and later fine-tuning with measured values from the real world and therefore can be compared to transfer learning. Overall, the authors utilize a broader definition for physics-informed models that also includes verifying DNN output with physical equations or engineering input data according to its physical relevance. Additionally, the authors classify graph neural networks (GNN) as a category of physics-informed neural networks due to their incorporation of grid structure in the model. They adapt a generalized formulation for physics-informed loss as a sum of conventional loss, parametric regularization terms and physical regularization $R_{\text{phy}}(X, \hat{y})$ that depends on the physical equations X and the model output $\hat{y}$. In section 5.3.3 several equations are evaluated in regard to their relevance for this work's use case. The authors' quite recent literature review lists several papers on power flow but no paper on power quality. Hence, the application of physics-informed neural networks to power quality will be first described here.

In [33] automatic differentiation of the swing equation is used to estimate frequency and

rotor angle of a generator with varying mechanical power. It is shown, that PINN are efficient in estimation of target parameters while also being able to derive unknown system parameters.

Finally, in [34] power flow is estimated through calculation of the voltage prediction loss of a neural network and the power injection reconstruction loss. For this, a neural network reconstructs power injections from solved voltages. Then, the partial derivative of the power balance equation formed by Kirchhoff's law, described here in equations 4 and 5, with regard to the voltages is used as joint loss function for the neural networks. Three different designs are developed for the verification model, an MLP, a bilinear neural network (BNN) and a BNN with additional topology information through weight multiplication with the adjacency matrix, akin to PANN. The output of the first neural network, that estimates voltages based on a mixture of partially available power injection, voltage magnitude and angle measurements is then physically validated in the second model. Additionally, a Gaussian noise with standard deviation $\sigma = 0.01$ is added to simulate measurement errors. Further, an optimal ratio of prediction loss and physics-informed loss was identified at $10 : 1$, therefore valuing the prediction quite strongly. Overall, the model that includes topology information performs best with an RMSE of 6.27×10^{-4} for active power in the IEEE 57 bus system. However, the authors assumed each bus provides at least two measurement properties out of active and reactive power injections, voltage magnitudes and voltage angles. Because of that, a well-defined power flow equation with as many unknowns as measurements is available.

Finally, graph neural networks (GNN) also leverage physical information by making use of the graph-like structure of power grids. Graphs are usually represented by edges and nodes, often called vertices. In GNN, nodes are associated with feature vectors that in this work's case are measured values. In power grids, edges could represent power lines, transformers and switches, and may as well associated with edge features such as impedance. Then, messages are passed from one node to another and feature vectors are updated based on the input to each node. Repeating the message passing step multiple times results in a further spread of node information. This process is described for power flow state estimation in [35].

The authors of [36] develop a generic graph convolutional neural network (GCNN) for power flow estimation. In the first layer, they only connect the most active inputs to outputs for which the highest correlation between each other exists. Hereby, adjacency is estimated, and therefore the approach works without information about grid topology while also reducing amount of parameters. The pruned layer is followed by several graph convolutional layers and finally by another pruned layer. While in their test cases a fully connected neural network achieved better results for the IEEE 39 bus system, the GCNN showed better performance in larger grids. Best results were achieved in the Texas grid system with 1350 loads and an MSE of 0.0244. However, in all cases more inputs were used than outputs estimated.

The work in [37] generalizes GNN so that even networks of sizes that were not seen during

training can be estimated with a reasonable efficiency. For this, the adjacency matrices are used as input to the GNN along with production and consumption measurements. Then, the measurements are embedded to the size of a hyperparameter d and the adjacency matrices directly incorporated into the network architecture. Using this method to estimate power flow in transmission systems, they achieve a median MSE of 0.0715 between the 20th and 80th percentile for a random topology of a size already seen in training.

In [38] and [39] a GNN model with attention mechanism is proposed for power flow state estimation. Whereas [39] only takes PMU measurements into account, [38] also uses legacy measurements that include active and reactive power flow and injections and magnitudes of branch current and bus voltage. A GNN is trained and evaluated once with all measurements available and then with percentages of measurements removed. On the IEEE 30 bus system with 3 bus voltage PMUs, 8 branch current PMUs and 100 legacy measurements, an MSE of 1.233×10^{-5} was achieved. Similar results were accomplished for the IEEE 118 bus system as shown in table 4. However, the losses significantly increased with randomly removed measurements. At 60 percent excluded measurements, the MSE loss was at $> 10^{-3}$ and $> 10^{-2}$ in case of the IEEE 118 and the IEEE 30 bus system, respectively.

In [40] the authors expanded the test cases of [38] to networks with more nodes and evaluated their performance with regard to amount of training samples. Table 4 shows an overview of reviewed papers that offer comparable results. Notably, all presented papers performed power flow analysis for distribution systems or in case of [37] for transmission systems. Nevertheless, comparing different approaches proves difficult due to differences in chosen grids, amount of measurement nodes, measurement points and lastly different loss methods. Generally, one can see that low measurement penetration is detrimental to all algorithms. Moreover, combining different types of measurements such as PMU with traditional measurements appears to be a standard approach that leverages as much data as available.

4.2 Complex Neural Networks

Conventional neural networks are designed to work with real numbers in their activation functions, their loss methods and backpropagation algorithms. Therefore, in most of the existing work, complex values are split into their real and imaginary component and then used as real numbers in neural networks. For that, either the input array is expanded by concatenation or its dimensionality is increased. Several representations for those real-valued neural networks (RVNN) exist that will be further examined in 5.2.1. However, multiplying inputs with weights results in an unnecessarily increased complexity for split-valued neural networks. This can be shown using a fully connected complex-valued neural networks (CVNN) with two neurons at each layer. For this network, four weights exist between each layer since the kernel shape is $(2, 2)$. A split-valued neural network requires four neurons per layer to express the same input and therefore the kernel shape is $(4, 4)$ resulting in 16 weights. This reduction in the degree of freedom through usage of CVNN is also described in [42]. Here it is shown, that between RVNN and CVNN the latter are better at generalization. It is argued that a high degree of freedom, which is the case for

Grid	**Measurements**	**Algorithm**	**Results**	**Ref.**
IEEE 123	2 PMUs, varying pseudo measurements	Pruned PANN	MSE 3.42×10^{-2}	[31, Tran 2021]
IEEE 37	5 PMUs, 7 current measurements, 26 pseudo measurements	Partitioned PANN	MSE 5.33×10^{-3}	[29, Zamzam 2019]
IEEE 34	2 PMUs	DNN	MAE 0.13 voltage angle, MAPE 0.26 voltage magnitude	[41, Azimian 2022]
240 node Midwest U.S.	6 PMUs	DNN	MAE 0.15 voltage angle, MAPE 0.25 voltage magnitude	[41, Azimian 2022]
IEEE 57	measurements at all nodes	PINN	RMSE 6.27×10^{-4}	[34, Hu 2021]
IEEE 37	unclear, more measurements than outputs	DNN	MSE 0.0316	[36, Bolz 2019]
IEEE 37	unclear, more measurements than outputs	GCNN	MSE 0.1106	[36, Bolz 2019]
Texas 1350 loads	unclear, more measurements than outputs	GCNN	MSE 0.0244	[36, Bolz 2019]
random topologies 10-110 nodes	measurements at all nodes	GNN	MSE 0.0715	[37, Donon 2019]
IEEE 30	3 bus voltage phasors, 8 branch current phasors, 100 legacy measurements	GNN	MSE 1.233×10^{-5}	[38, Kundacina 2022]
IEEE 118	7 bus voltage phasors, 26 branch current phasors, 500 legacy measurements	GNN	MSE 2.038×10^{-5}	[38, Kundacina 2022]

Table 4 Comparison of existing work on power flow state estimation under varying conditions.

RVNN and for networks with many neurons, leads to a larger generalization error. For neural networks it is a well-known fact that many parameters and a high degree of freedom can lead to overfitting since the model may be able to memorize the training data.

The authors of [43] review complex valued neural networks (CVNN) that by design work with complex values through custom activation functions, loss methods and backpropagation algorithms. Finally, in [44] CVNN are applied to graph convolutional neural networks and applied to power grid state forecasting, attack detection and localization.

4.3 Related Domains

In this section, several related domains are described that are not part of this work but need to be considered for later system integration. At first, the topology of the underlying grid is assumed to be known. Either the DSO provides the required information or topology estimation algorithms are employed.

Secondly, bad data detection describes the process of identifying flawed data which are mostly due to measurement errors. Several traditional detection algorithms exist, e.g. in the chi-square method, but also data-driven methods are nowadays used more commonly [10].

Furthermore, optimal measurement points need to be identified that capture the system state as well as possible. Due to the costs of measurement devices and practical limitations in installation locations, purely statistical algorithms may be insufficient. For example, the measurement placement algorithm developed in [29] disregards the issue of transformers hindering propagation of power quality issues. Thus, the simple separation into compartments based on minimizing the distance to measurement units may not be feasible. Additionally, measurement devices may not be able to installed at the chosen locations if the cable is underground. However, the algorithm could be modified by adding a condition that requires at least one measurement unit in each section separated by a transformer and another condition that limits the selectable placement locations.

5 Solution Concept

In previous chapters the insufficiencies of traditional model-based state estimation algorithms were shown. The expected low measurement penetration of the power grid either causes numerical instability of conventional algorithms or fully causes them to fail. Thus, this work's solution concept is based on a different set of algorithms, data-driven algorithms that utilize large amounts of data to train a model to predict system states. Since previous work showed successful approaches involving neural networks, multiple neural network models will be implemented and compared.

For that, the workflow can be split into two phases, a model training, including data generation, and an estimation phase. Figure 11 shows the phases and overall workflow used

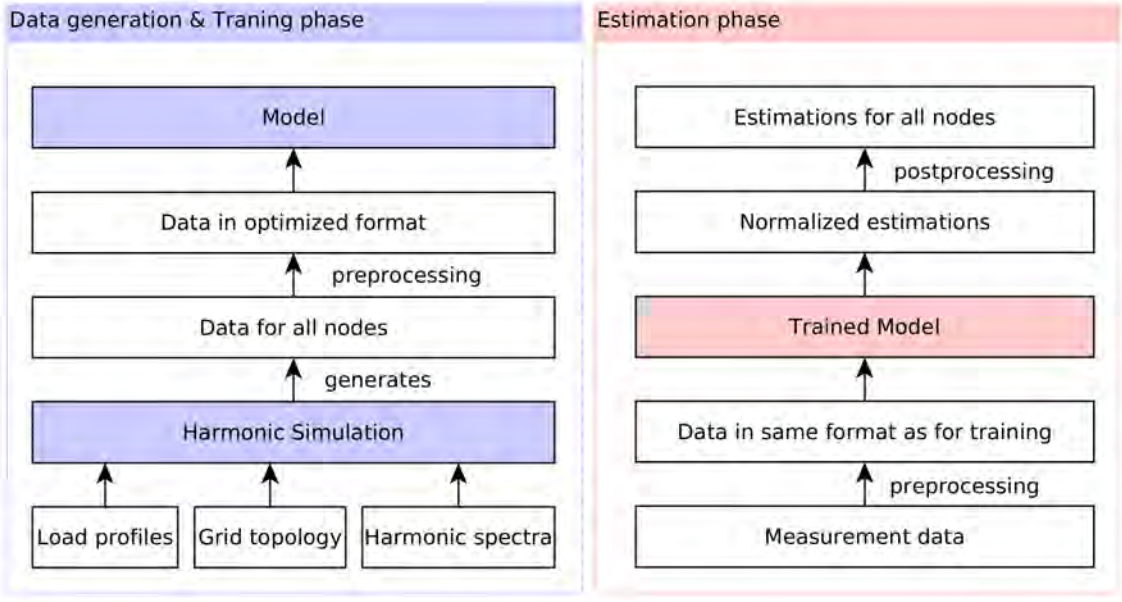


Figure 11 Power quality state estimation workflow during data generation, training and estimation phase. Source: own illustration

for power quality state estimation. During the first phase, harmonic data is generated in a simulation environment that will be further explained in the following section 5.1.

Then, the generated data is preprocessed so that its distribution is optimal for training of a neural network. Preprocessing is described in section 5.2. Different models are trained using the optimized data for all nodes as target and selected measurement points as input. Section 4.1.3 further showed that incorporating physical information in the neural network's architecture can be beneficial to its performance. Hence, different architectures are developed in section 5.3.1 and 5.3.3. Then, the developed models are improved by adding skip connections in section 5.3.4 and Gaussian Layers to combat measurement noise in section 5.3.5. Section 5.3.6 discusses methods of reducing model complexity by separating the grid into compartments and estimating each compartment individually. Finally, the overall model architecture is presented in section 5.3.7.

The second phase is the estimation phase in which the trained model is used in a live environment. Here, data is only available for measurement nodes and preprocessed in the

same way as during training. The normalized estimations are post-processed by inverting normalization so that voltage estimations for all nodes can be derived.

5.1 Harmonic Simulation

In order to generate power quality data, a harmonic simulation of a power grid is conducted using OpenDSS. As a reference grid the CIGRE low voltage distribution network, described in [45], was chosen as it contains a residential, a commercial and an industrial section. In OpenDSS, a digital twin of the underlying power grid is developed with renewable generators added in form of photovoltaic systems and monitors attached to each node. Those monitors capture voltage magnitude and phase angle measurements for predefined frequencies. During training of the neural network, the target data consists of measurements for all nodes while only three nodes are chosen as input data. Those three nodes will later be referred to as measurement points. The reference grid is shown in figure 12 with the residential subnetwork on the left, industrial section in the middle and commercial subnetwork on the right. These subnetworks are operated at 400 V line-to-line. On top,

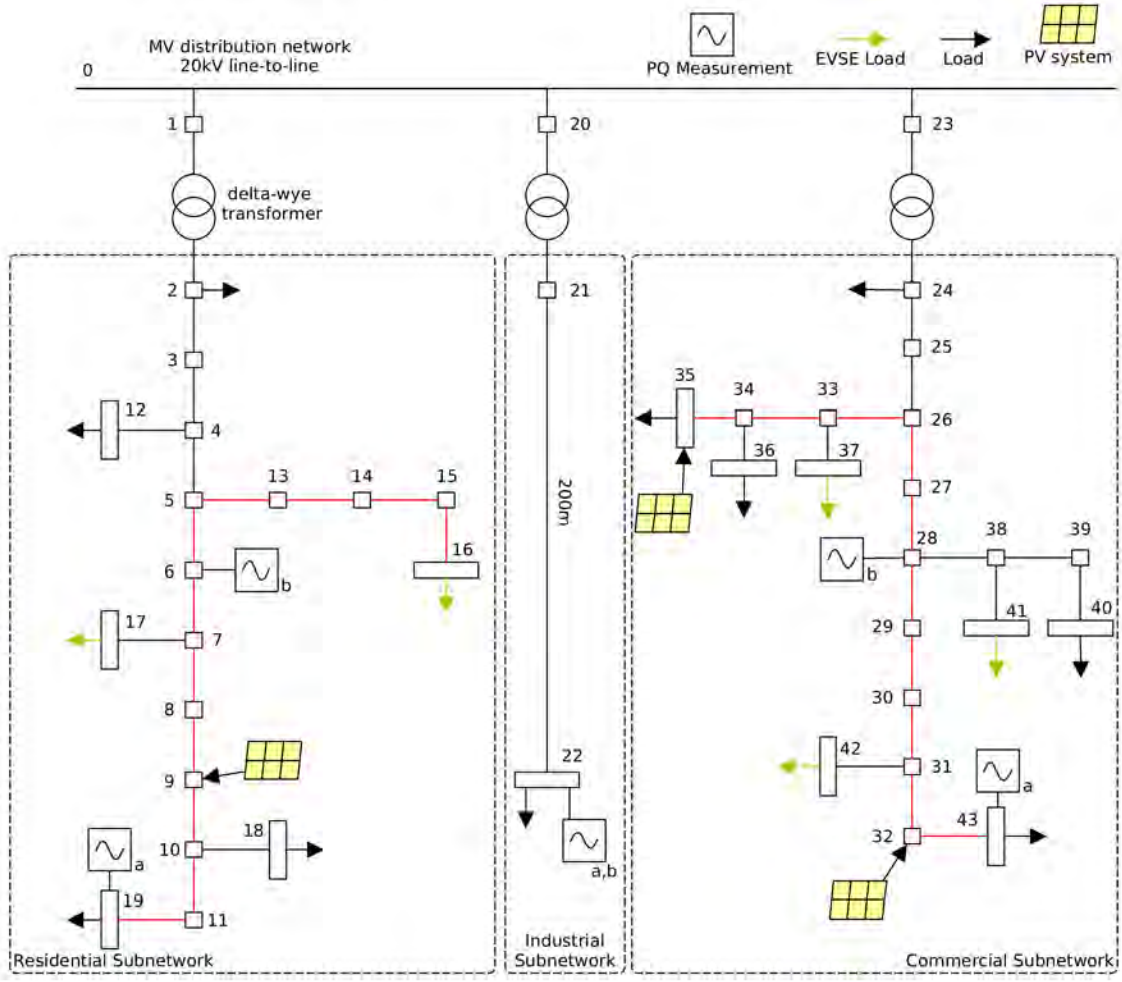


Figure 12 CIGRE low voltage distribution grid divided into three subnetworks with added photovoltaic (PV) systems, electric vehicle supply equipment (EVSE), loads and power quality measurement points. Adapted from [45]

the medium voltage (MV) distribution network operated at 20 kV connects the individual sections. Between the MV and LV section delta-wye transformers are installed with a

primary delta winding at the MV side and a secondary wye winding at the LV side. In the figure, selected measurement points are shown in two possible configurations a and b at the most distanced node from the transformer and in the middle of section, respectively. In red, the longest path within the same subnetwork from any node to the measurement node in configuration a, and in green EVSE are marked.

5.1.1 Load and Generator Profiles

For each simulation step, the load or generator receives its consumed or generated power from profile files which are read into memory at the program start. The training set is associated with different profiles than test and validation sets. Hereby, the generalization ability of the model shall be tested. Further, this prevents validation or test set samples that are similar to training set samples which may result in a falsely low prediction error of the model. While training data was artificially generated using the load profile generator described in [46], test and validation set load profiles are simplified versions of the *HTW Berlin* dataset [47]. The dataset consists of a combination of three-phase smart meter measurements measured over multiple years in 15 minute steps and active and reactive power measurements sampled every second throughout one week. Then, the authors combine both datasets to obtain high-resolution load profiles. For this work's use case, the data was fed to the simulation framework in 1 minute steps and averaged over all phases, assuming a simplified symmetrical load. Generator profiles of the test and validation set are based on collected data of exemplary PV systems, available in [48].

5.1.2 Harmonic Spectra

Harmonic spectra quantify harmonic distortion for each harmonic. In order to simulate harmonic distortions in power grids, OpenDSS allows loads and generators to be equipped with harmonic spectra. Here, loads require current spectra and generators can be provided with either voltage or current spectra [17]. For different types of equipment different spectra should be provided. While industrial loads may show the same or similar spectra over time, residential household equipment usually is only active for short periods of time and different equipment is used throughout the day. For example, tea kettles and stoves are used during the morning and evening, whereas television devices are typically used in the evening after dinner. Additionally, electric vehicle (EV) chargers in residential areas are typically used after work while those in commercial areas are used during work time. This shows the need for time dependent load spectra that offer a variety of spectra for different devices.

Due to time constraints, harmonic spectra were only used for inverters at those points with PV systems and some households and commercial buildings were assumed to be equipped with EVSE. In this simplified version, all other components were assumed to possess ideal spectra with no harmonic distortion. For PV inverters a measured spectrum of a 33 kW inverter at 100% power level was used [49]. The spectra used in simulation are shown in table 5 and also consist of measured data in case of EVSE. Here, the first EV is a 2014 model BMW i3 charged at 240 V with 15.5 A [50]. The second EV is a 2015 model Nissan

	PV Inverter		EV 1	EV 2	EV 3
Harmonic	**Mag.**	**Angle**	**Mag.**	**Mag.**	**Mag.**
1	99.956	76.957	100	100	100
2	0.32606	-94.822	-	-	-
3	0.45068	33.034	8.4	3.9	9.7
4	0.2183	146.21	-	-	-
5	1.7971	-178.53	2.1	2	8.8
6	0.038894	118.52	-	-	-
7	2.2295	101.36	3.6	1.6	4.0
8	0.23969	-101.19	-	-	-
9	0.12951	-17.256	1.3	0.9	2.7
10	0.16404	86.047	-	-	-
11	0.23619	72.842	1.2	1.4	4.8
12	0.010896	-124.28	-	-	-
13	0.08269	53.563	0.4	1.9	4.6
14	0.04204	-113.04	-	-	-
15	0.016148	5.4987	0.3	0.7	2.7
16	0.033937	78.594	-	-	-
17	0.017689	-98.512	0.3	0.3	2.1
18	0.0020779	-157.18	-	-	-
19	0.022024	165.55	0.01	0.2	0.9
20	0.010804	-80.658	-	-	-

Table 5 Harmonic spectra used in the simulation. Source [49], [50], [51], [52]

Leaf charged at 208 V with 16 A and the third EV is a 2015 Mercedes B-Class charged at 240 V with 15.4 A[51][52]. These spectra were applied to selected points in the grid with the spectrum of the BMW i3, i.e. EV 1, only applied to test and validation set and the others used solely on the training set. Hereby, the model's generalization ability shall be tested.

To increase understanding of harmonic spectra voltages and currents up to the 50th harmonic of power converters were measured in a laboratory setting. Here, three lead batteries are connected to an inverter and both charged and discharged with various loads. Figure 13 shows harmonic voltages excluding the fundamental frequency of 50 Hz. One can see that uneven harmonics especially those that are not multiples of three make up the majority of the spectrum. In color, the phase angle in radians is depicted which for multiples of three is around zero. Notably, the 7th harmonic's phase angle is close to π and for two values the phase angle exceeds π and thus wraps around to values slightly larger than $-\pi$. This is due to the inability of measurement devices to capture phase wraps as they only ever see the wave's position between $-\pi$ and $-\pi$. In later sections, especially for the complex value representation in section 5.2.1, this will become relevant.

Additionally, harmonic currents were measured in the experiment. Figure 14 shows the measurements for all harmonics including the fundamental frequency. As expected, the measured effective currents increase as the batteries are charged and discharged. Also, one can see that in the fundamental frequency the phase angle quickly diverges for negative and positive loads. However, for harmonics this behavior is not as obvious. Finally, in figure 15 the measured effective values were used to reconstruct sine waves for harmonic

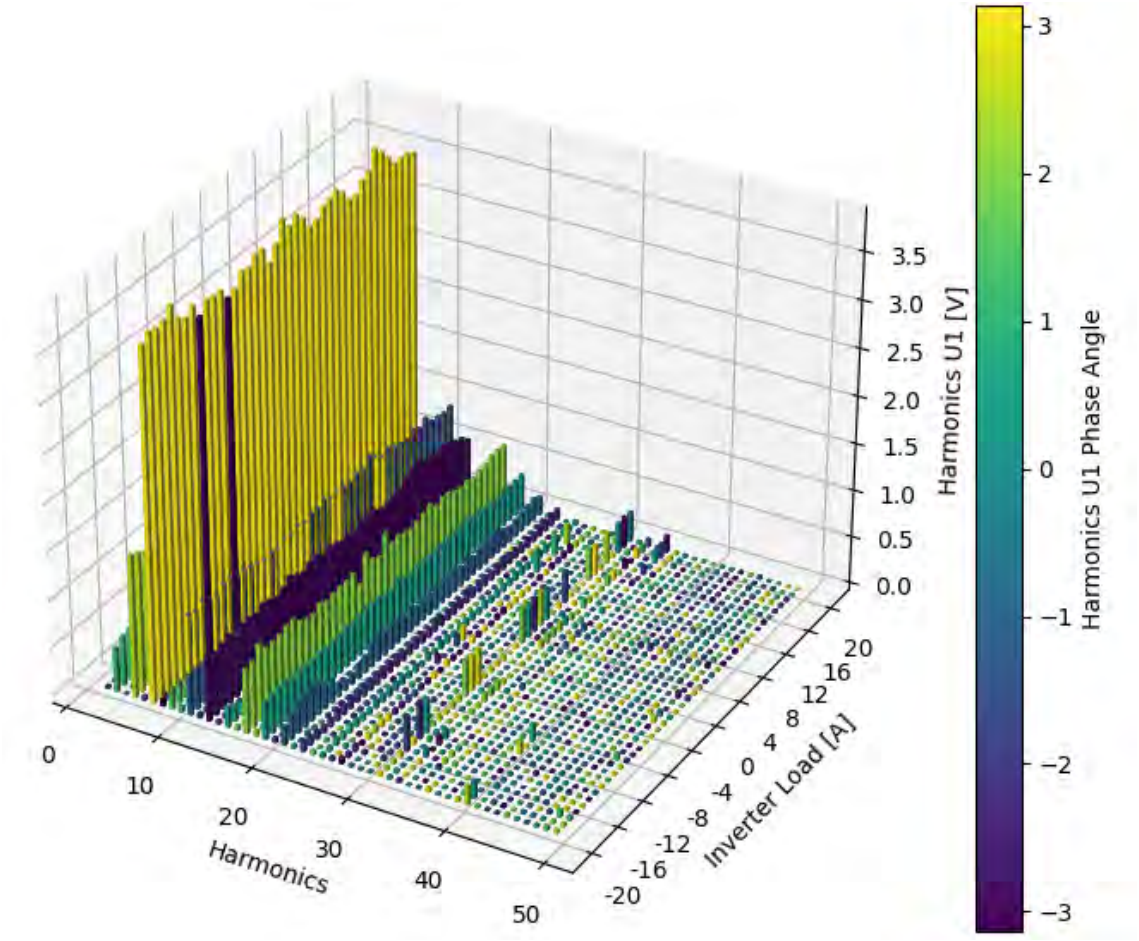


Figure 13 Measured effective values of harmonic voltages phase 1 without first harmonic for various loads connected to an inverter. Source: own illustration

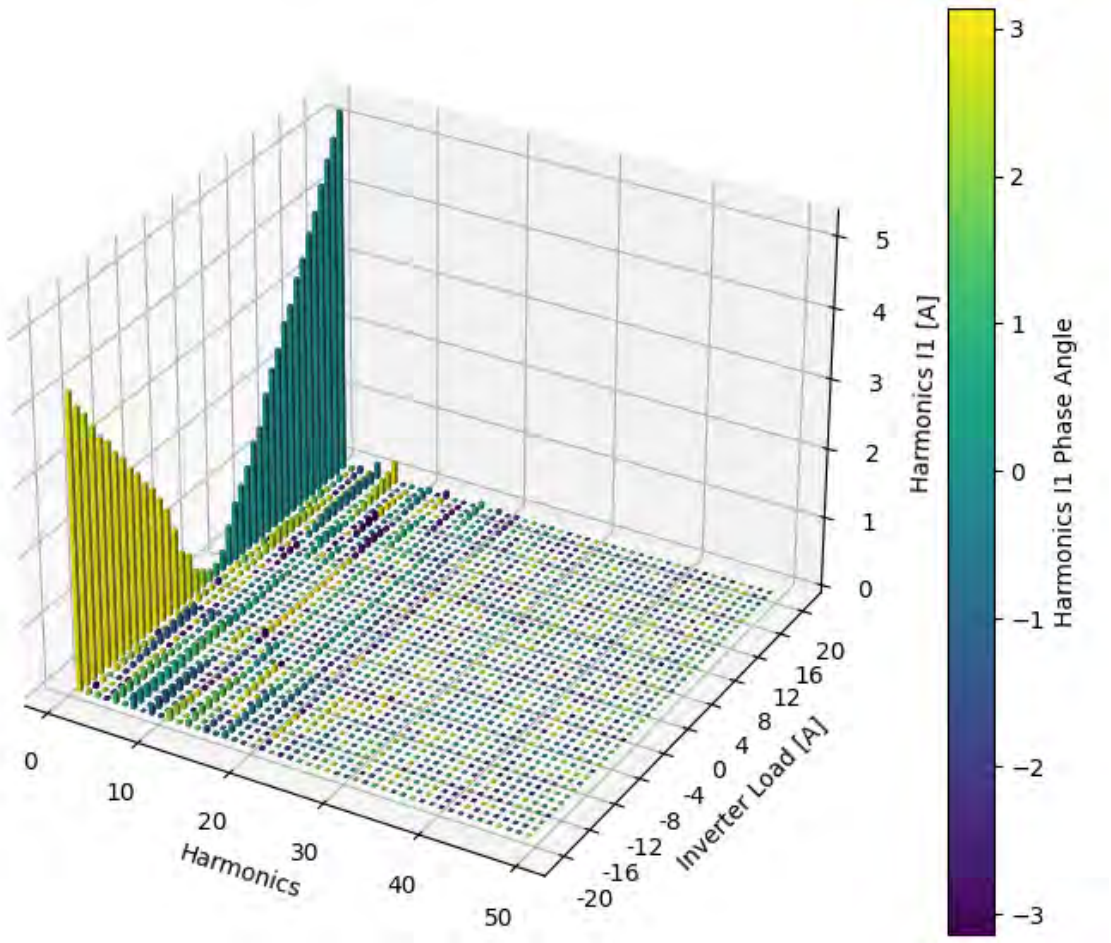


Figure 14 Effective values of harmonic currents phase 1 including first harmonic for various loads connected to an inverter. Source: own illustration

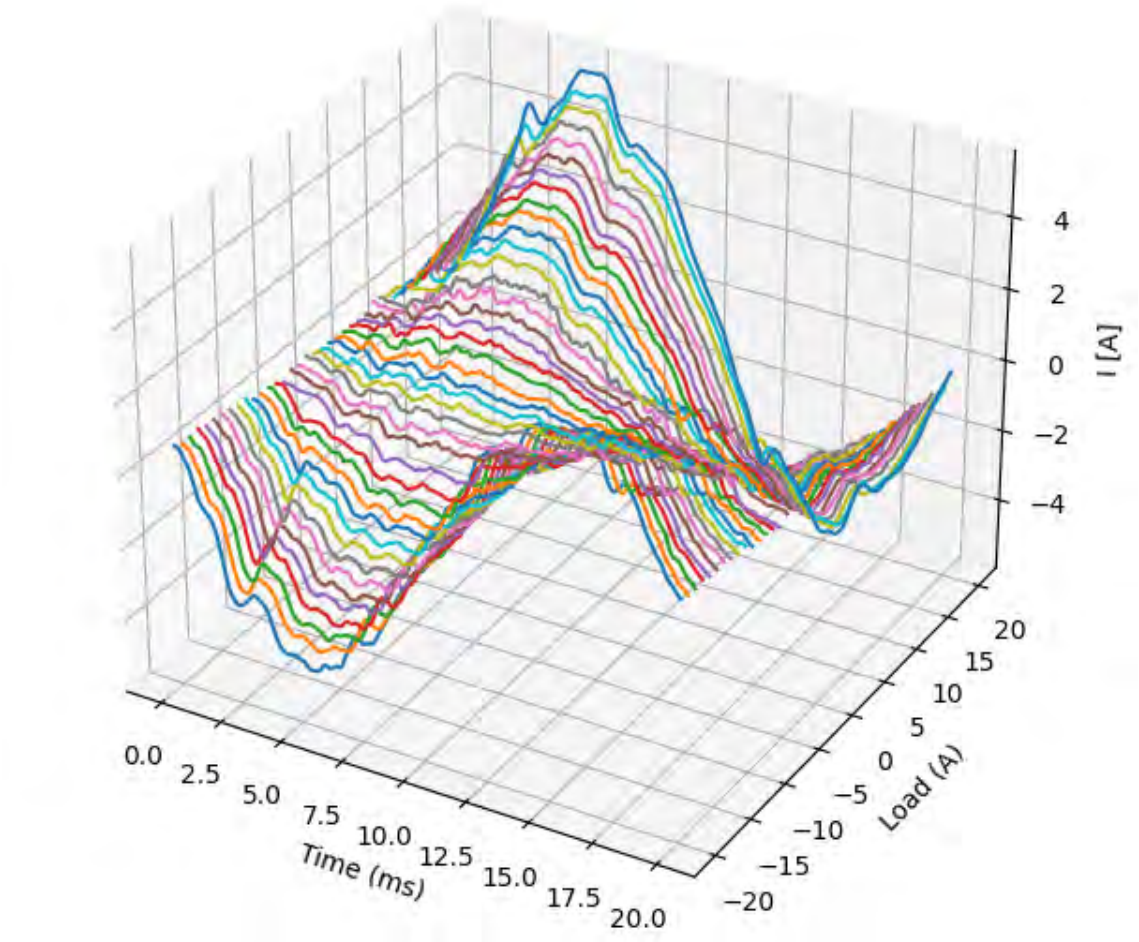


Figure 15 Reconstructed sine waves phase 1 for various loads connected to an inverter. Source: own illustration

currents. Here, the changing phase angles for various loads is clearly visible. Notably, the used measurement device captures current angles relative to voltage angles. Therefore, the change in phase angle for different loads is only visible in currents. All in all, this shows the necessity for providing spectra for various loads and dependent on time. However, this significantly increases the simulation complexity and was omitted in this work due to time constraints.

5.2 Data Preprocessing

In this work, multiple neural network models are compared and evaluated. For all models, a deep understanding of the underlying data is necessary in order to improve the performance of artificial neural networks.

5.2.1 Complex Value Representation

In alternating current (AC) circuits voltages are measured as complex values for which several representations exist. Section 4.2 showed that first concepts for complex-valued neural networks (CVNN) exist. However, implementations of these concepts are not yet widely available. Hence, a RVNN is used in this work in which the complex components are split into separate features. This section will evaluate the advantages and disadvantages of each representation.

At first, the training data for the neural network in Cartesian form is plotted in a scatter plot for each frequency. Figure 16 shows one million input data points plotted as scatter plot for each frequency. On the x-Axis the real part and on the y-Axis the imaginary part of the complex voltage is plotted. One can see that the data is mostly contiguous and in the harmonics curved patterns are visible. These curves are caused by the strong physical relationship between real and imaginary part of a complex number. If in polar representation the phase angle changes while the magnitude remains equal, in Cartesian representation both the real part and the imaginary part will change. Forcing neural networks to rebuild these relationships that offer no additional information might be unnecessary and hinder the network in learning more relevant patterns. Therefore, different kinds of complex value representations were evaluated.

In polar form a simpler linear and even distribution across the plane was observed. However, phase wrapping becomes an issue where phase values that are slightly bigger than π loop over to values slightly bigger than $-\pi$. In figure 17 this is visible for 400 Hz and in the input data this phenomenon also occured at 450 Hz, 500 Hz, 600 Hz and 650 Hz. This is problematic for two reasons. For once, values that are logically similar are shown far apart. Therefore, the neural network might not be able to learn their correlation. Secondly, as seen in the graphic many values are clustered close to each other while a large section in the middle is uninhabited. Hereby, differences between values that are now put very close to each other might not be as obvious to the neural network while a lot of space is unused.

Because of this, quantile transform was evaluated as technique to map the problematic data distribution to a uniform distribution by mapping input values to their corresponding

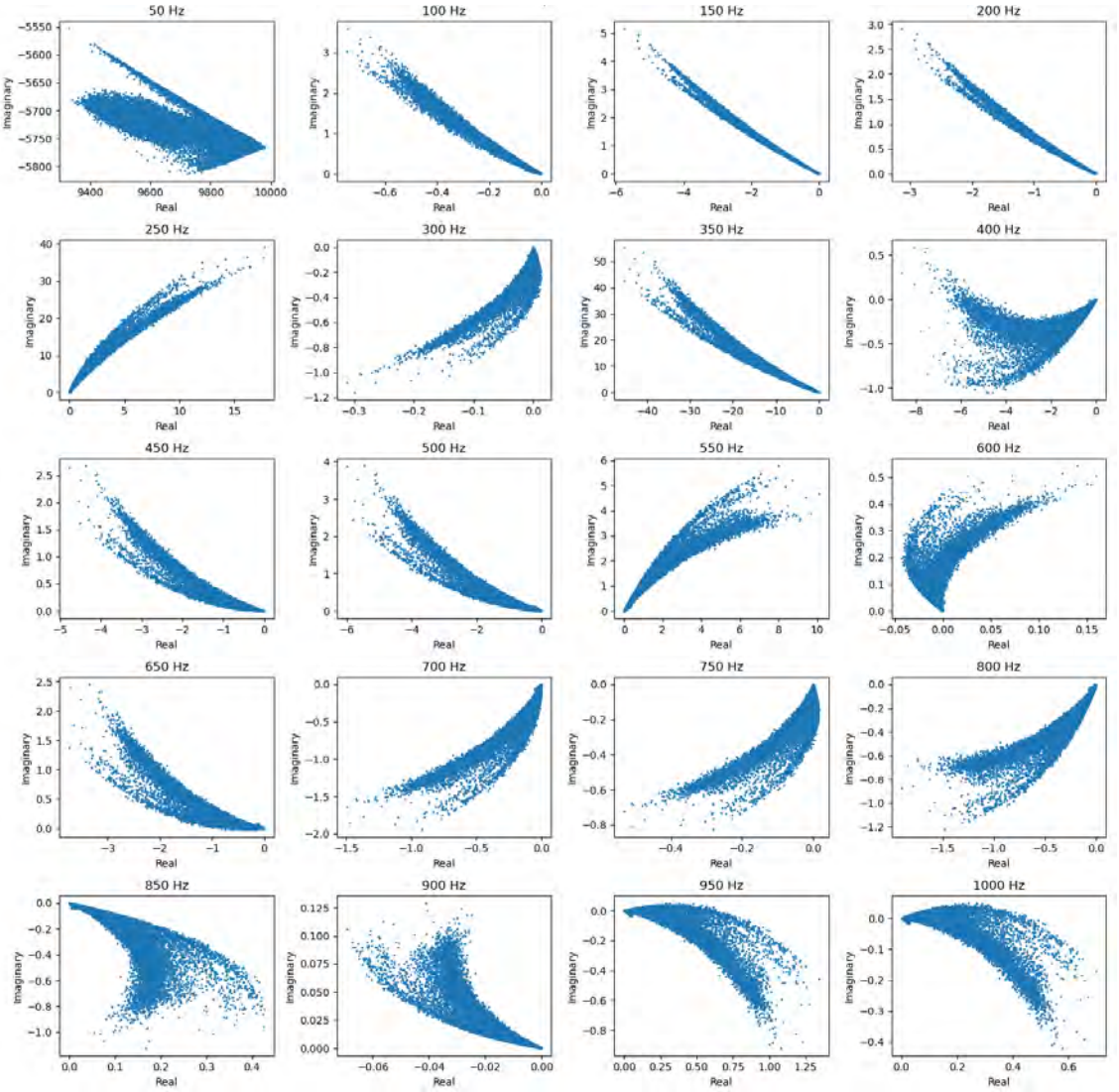


Figure 16 One million data points of input data in Cartesian form scattered over real and imaginary part for each frequency. Source: own illustration

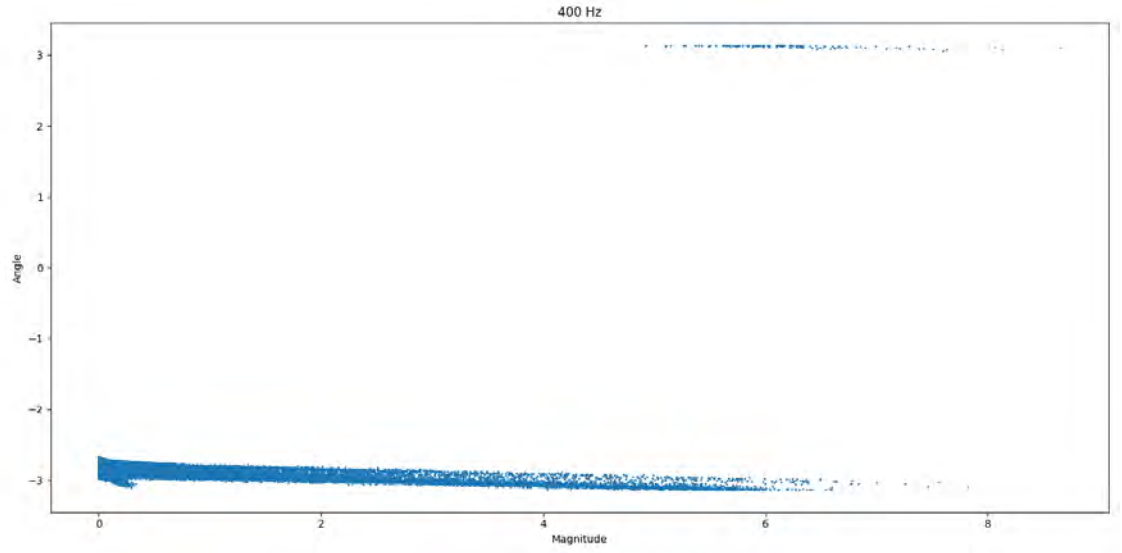


Figure 17 Scatter plot of input data showcasing the issue of phase wrapping in polar form. Source: own illustration

quantiles in the output which will be further described in section 5.2.2

Instead, the issue of phase wrapping can be circumvented by usage of the exponential form that puts the phase angle in the exponent of Euler's number as shown in equation 13.

$$re^{i\phi} \tag{13}$$

Now, the question arises of how to save the data in exponential form. While saving the magnitude and phase angle ϕ dicrectly is certainly possible, it then is equivalent to the polar form in terms of data spread. Further, solving $e^{i\phi}$ and saving the real and imaginary part along with the magnitude is feasible. However, hereby the dimensionality is increased and the issue of a strong relationship between real and imaginary part is reintroduced. Therefore, a partial exponential form is proposed where the magnitude is saved along with the imaginary part of the solved $e^{i\phi}$. Saving either the real or imaginary part is sufficient since the length of the vector $e^{i\phi}$ must be equal to 1 as it is later multiplied with the actual magnitude r. Nevertheless, the sign of the real part needs to be preserved. Since the magnitude is always positive, the sign can simply be appended to the magnitude without the need for introducing additional dimensions.

Furthermore, in figure 18 an additional problem is visible at 300 Hz that exists in both

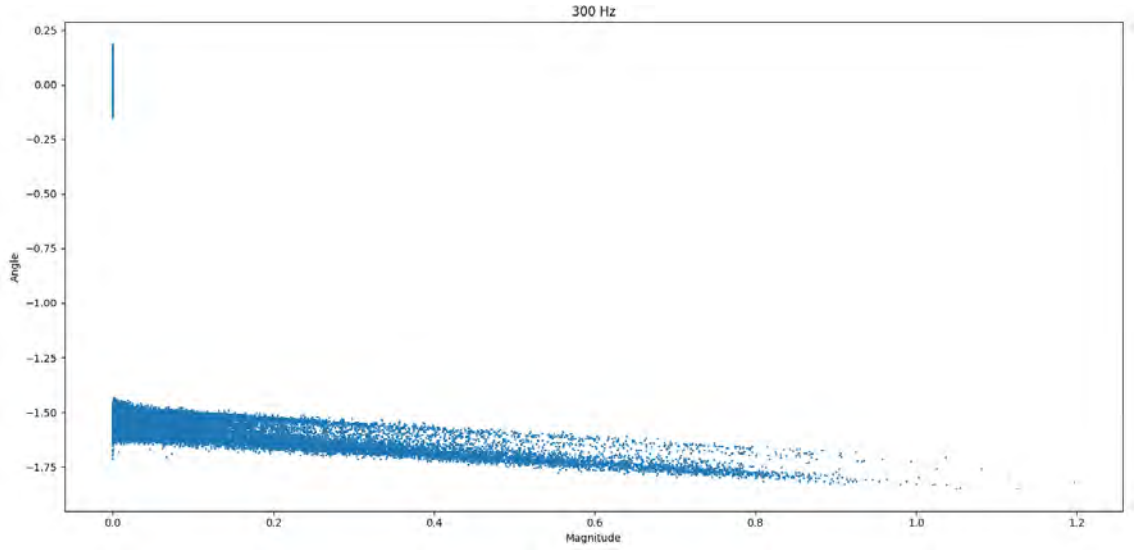


Figure 18 Scatter plot of input data in polar form showcasing the issue of unmeasurable angles at third harmonics. Source: own illustration

polar and exponential form. For frequencies that are multiples of three of the fundamental frequency, clustering behavior can be observed in both polar and exponential form with one cluster at magnitudes close to zero. Here, the phase angle can not be properly measured by measurement devices, because values are equal to or close to zero in magnitude. This is caused by the usage of delta-wye transformers in the OpenDSS simulation grid in which the delta-connected winding is on the medium voltage side and the wye-connected winding on the low voltage side. In the reference network 12 the effect occurred in the medium voltage and the industrial section. Even though the industrial grid section is wye-connected, it does not show any third harmonics, as no harmonic sources such as an inverter was modelled at

this stage. Figure 19 shows the data distribution in exponential form with the magnitude

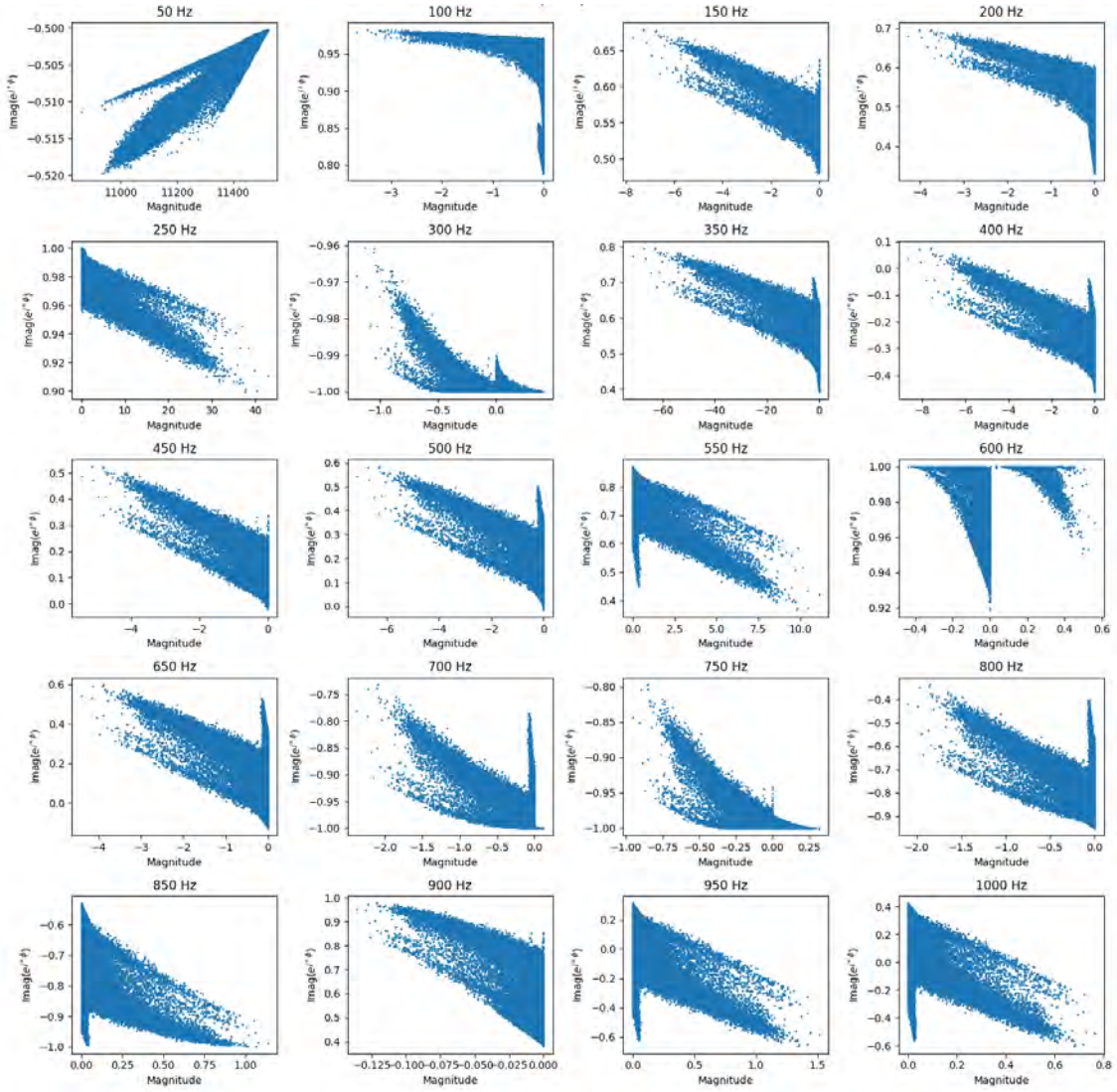


Figure 19 Scattered input data in exponential form after moving phase angles for zero magnitude values. One million data points are shown with magnitude and $imag(e^{j\theta})$ for each frequency. Source: own illustration

on the x-Axis and $imag(e^{j\theta})$ on the x-Axis. Here, for those third harmonic values where the magnitude is close to zero the phase angle was manually set to the average of all phase values in preprocessing. Hereby, the data is more evenly spread while no physically relevant information is lost as the phase angle has no meaning for magnitudes at or very close to zero.

5.2.2 Normalization

Even though neural networks can work with data in any range, using different scales for features results in artificial weighting where for those features with smaller values the losses are underestimated while those with large numerical values disproportionately influence the gradients. Mathematically, this can be seen in the eigenvalues of the hessian matrix which will also be on dissimilar scales. The hessian matrix is a second-order partial derivative of the loss function, and its eigenvalues provide information about the convexity of the

loss function. Generally, a convex loss landscape is preferable [53]. Graphically, the loss landscape would be stretched along the feature with larger values and thus look elliptical in contrast to circular for similar scales. Due to the different shape, the gradient will not always point towards a minimum. Instead, with features on a similar scale weight updates are consistent across all input dimensions and therefore the gradient descent algorithms converge more rapidly.

In this work's use case the feature scales differ widely. While the fundamental frequency shows voltage magnitudes up to or even above the respective voltage level of the underlying power grid, harmonics are undesired and typically much smaller.

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{14}$$

Equation 14 shows the formula for Min-Max normalization bringing all values in range [0..1]. Here, X represents the dataset, $\min(X)$ and $\max(X)$ the minimum and maximum values in X, respectively. For multidimensional data that is to be normalized along specified axes only, this formula can be generalized by computing the minimum $\min_{\text{axes}}$ and maximum $\max_{\text{axes}}$ along the relevant axes. Additionally, values can be rescaled to an arbitrary range $[a..b]$ by generalizing equation 14 to 15.

$$X' = a + \frac{(X - \min_{\text{axes}}(X))(b - a)}{\max_{\text{axes}}(X) - \min_{\text{axes}}(X)} \tag{15}$$

Standardization, also commonly referred to as Z-score normalization, scales the data so that it has a mean $\mu = 0$ and a standard deviation $\sigma = 1$. The equation is given in 16.

$$X_{std} = \frac{X - \mu}{\sigma} \tag{16}$$

Here, each feature must be standardized separately to deal with different scales and maintain independence. However, even with separate standardization to a mean of 0 and standard deviation of 1 for each feature, the respective values will still reside in different ranges.

Lastly, quantile transform redistributes values to follow a uniform probability distribution and was therefore considered to tackle non-uniform distributions of input data as shown in the previous section 5.2.1. While in the case of Min-Max scaling, a fixed minimum and maximum value is computed for each feature in the training set and subsequently used to normalize the test set, quantile transform is non-linear. In quantile transformation, each data point is mapped to its percentile rank based on the cumulative distribution function (CDF) of the training set. This means that when applying the same transformation to the test set as derived from the training set, one must make an assumption that the test set follows a similar distribution to the training set. If values in the test set exceed the maximum values seen during transformation of the training set, the transformation is not well-described and prone to errors [54].

Even though a uniform probabilistic distribution is desirable, the non-linearity of the trans-

formation and reliance on specific characteristics in the input data make this approach impractical.

In [55] the technique of batch normalization is introduced in which the layer inputs are normalized. Contrary to the previously mentioned normalization techniques, the normalization is performed for each batch of training data as part of the model. Initial tests of batch normalization did not lead to improved results. Hence, min-max normalization was adopted for this work and a method implemented that normalizes across specified axes.

5.3 Neural Networks for State Estimation

As seen in previous sections, power grid data is often non-linear, and its intricate relationships can be rather complex. Because of that a model is required that is capable of handling non-linear data. Additionally, the input data may include measurement noise or fully missing data points. Artificial neural networks are well-suited to these requirements. In this work several models and architectures are evaluated and techniques for model improvement are applied.

A simple dense neural network with fully connected layers will be used as benchmark to compare more sophisticated architectures against.

5.3.1 Physics-Aware Neural Networks

Confining the model to the physics of the underlying power grid can be advantageous in many regards. For once, the model complexity is reduced through pruning of connections that represent no physical properties. Secondly, the degree of freedom is reduced which may improve the model's capability to generalize as discussed earlier. Nevertheless, manual confinement also comes with caveats. A model that is too limited in its degree of freedom may not be able to fully capture the complexity of the problem, resulting in underfitting. Additionally, the required mathematical operations can be memory intensive especially in the case of high-dimensional data.

In this thesis, a generalized approach to PANN is implemented that works for all kinds of grid topologies, for multidimensional input and in case of incomplete input vectors. This work builds upon the papers previously described in section 4.1.3, especially [29], [30] and [31].

At first, the nodal admittance matrix is calculated for all frequencies based on the *pandapower* model of the power grid. It contains all grid topology information required for building PANN.

Figure 20 shows the admittance matrix of the exemplary grid shown in figure 21. In the following sections and figures this grid will function as simple basis for explanations. From this admittance matrix Y, the adjacency matrix is built using equation 17.

$$A_{i,j} = \begin{cases} 1 & \text{if } |Y_{i,j}| > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

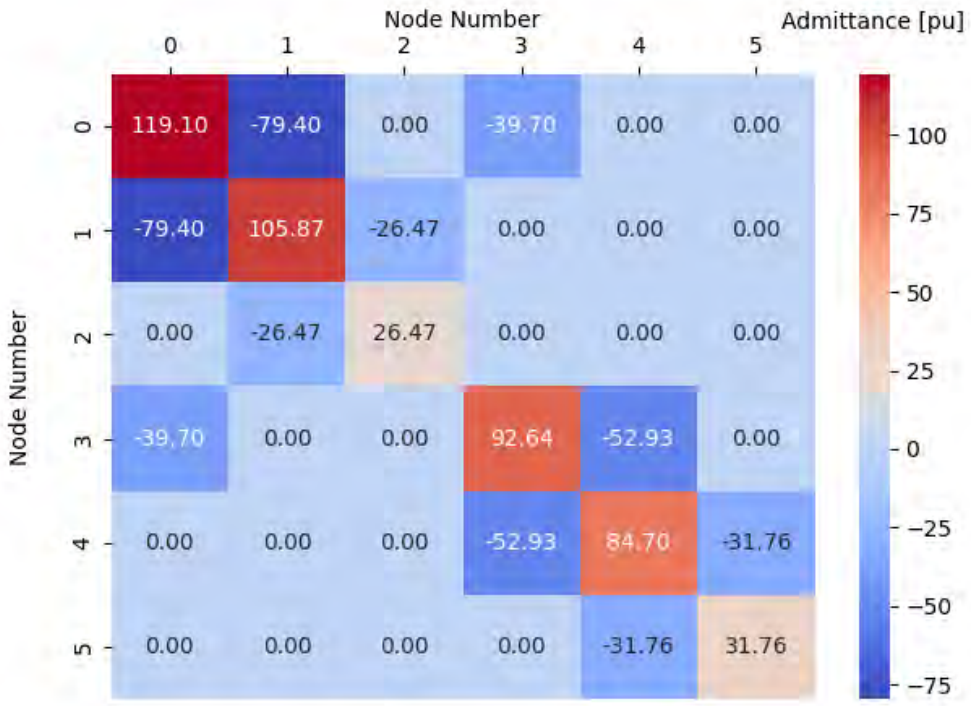


Figure 20 Real part of an exemplary admittance matrix at 50 Hz in per-unit (pu). Source: own illustration

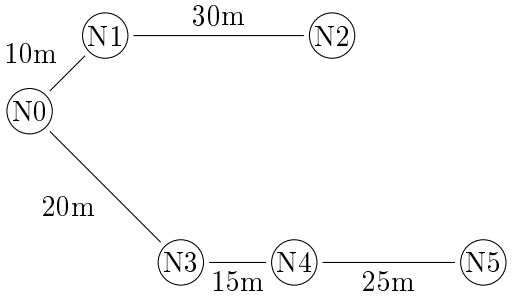


Figure 21 Example grid with six nodes and associated line lengths. Source: own illustration

Starting with a fully connected dense neural network, connections are pruned based on the adjacency information. This results in the network topology given in figure 22 if all nodes of the power grid are input nodes, i.e. if all nodes either provide measurement or replacement values. Notably, the connections from one neuron to itself will always exist

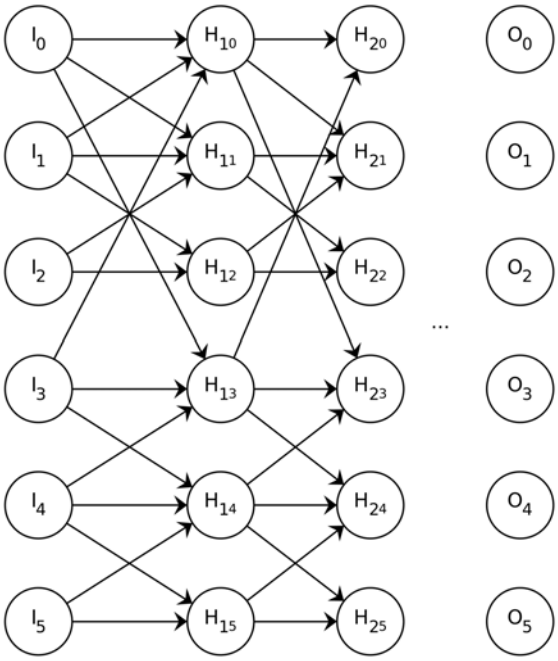


Figure 22 Neural network with multiple inputs pruned via adjacency in the underlying example grid in figure 21. Nodes in the input, hidden, and output layers are denoted by I, H, and O respectively. Source: own illustration

unless the node is fully isolated. One can easily see that neurons in hidden layers are only affected by inputs of adjacent nodes in the underlying grid topology. On the other hand, figure 23 shows the network topology if only one measurement value exists at node $N0$. In this case, in the first hidden layer those neurons corresponding to nodes $N2$, $N4$ and $N5$ are not provided with any input at all. Even in the second hidden layer one node without input remains at node $N5$. Those nodes without any input are marked as red and connections from these nodes are marked with dashed lines in figure 23. Additionally, the longest required path for all neurons being affected by the input is marked in red. By comparing this path to the grid topology, one can see that the length is equal to the longest distance between any node and input node $N0$. From this, we can derive that in PANN at least L layers are required so that all outputs are affected by the input. This can be generalized into equation 18 where for each node in the set V the minimal distance is found between itself and a set of measurement nodes n_m by the distance function d. Out of all the results, the maximal value is equal to the required layers.

$$L = \max_{v \in V} \min_{u \in n_m} d(v, u) \tag{18}$$

Pruning neural networks while preserving their capability to capture physical laws poses a non-trivial challenge due to the opaque nature of hidden layers. Unlike in image classification tasks where the inputs usually consist of raw pixel data and activated neurons

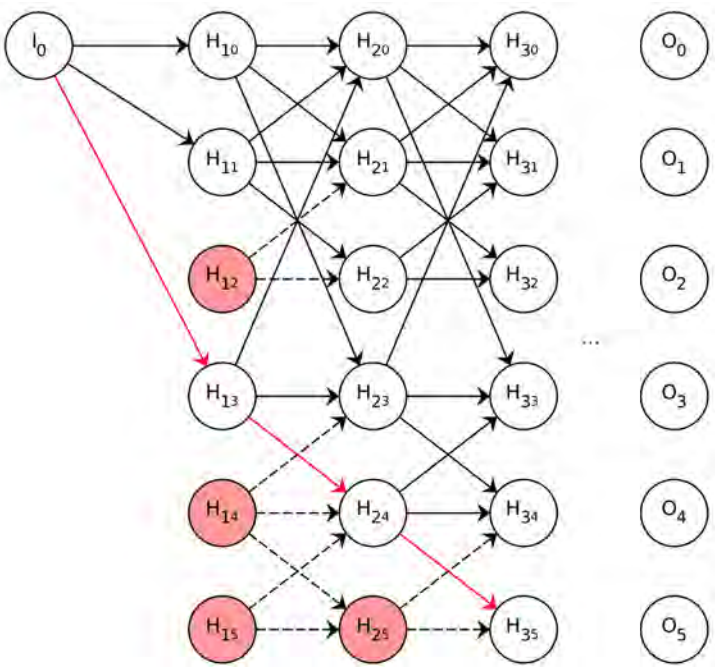


Figure 23 Neural network with single input pruned via adjacency. Red nodes remain unresponsive to the input, while red arrows delineate the maximal propagation path required for all outputs to be influenced by the input. Source: own illustration

in deeper layers may correspond to recognized shapes, the mapping between neurons in hidden layers and the physical characteristics of power grid systems is less straightforward [56].

During the highly iterative process of model selection and fine-tuning, pruning is associated with a lot of effort and can be circumvented. Even though by pruning the model size is reduced and fewer weights need to be trained, these optimizations can be employed after a suitable model has been identified and tested. Instead, custom layers are developed in which those weights corresponding to non-existent connections are set to zero.

This approach requires matrix multiplications with altered versions of the admittance matrix. In this work, two different custom layers are implemented, adjacency-pruned layers and admittance-weighted layers. While adjacency-pruned layers are based on the adjacency matrix and in a binary fashion multiply weights either by one or by zero, admittance-weighted layers multiply the weights by the respective admittance. Masked weights M for adjacency-pruned layers are calculated as shown in equation 19.

$$M_{i,j} = W_{i,j} \cdot A_{i,j} \tag{19}$$

However, in the admittance-weighted layer further factors need to be considered. Firstly, the absolute value of the admittance matrix is normalized using min-max normalization as described in equation 14. Then, a logarithmic redistribution of values is applied that brings the values closer to 1. This approach alleviates the risk of vanishing gradients which might occur if weights are multiplied by values close to zero. Hence, a direct comparison between a model trained with a redistribution factor $k = 50000$ and one model without redistribution is shown in figure 24. Here, two models with four consecutive admittance-weighted layers

were trained once with and once without redistribution of admittance values prior to multiplication with the weights. Then the gradients of the trained models were extracted and put into 500 bins showing their respective incidence rate. Under optimal conditions, gradients should approximate a Gaussian distribution and exhibit minimal variation across different layers. For both architectures under consideration, it is observed that gradients receive updates in the final layer. However, in the model trained without redistribution, gradients in all preceding layers progressively approach zero values. This phenomenon indicates a suboptimal performance in the backpropagation of gradients, commonly referred to as the vanishing gradients problem. The problem of vanishing gradients may also be

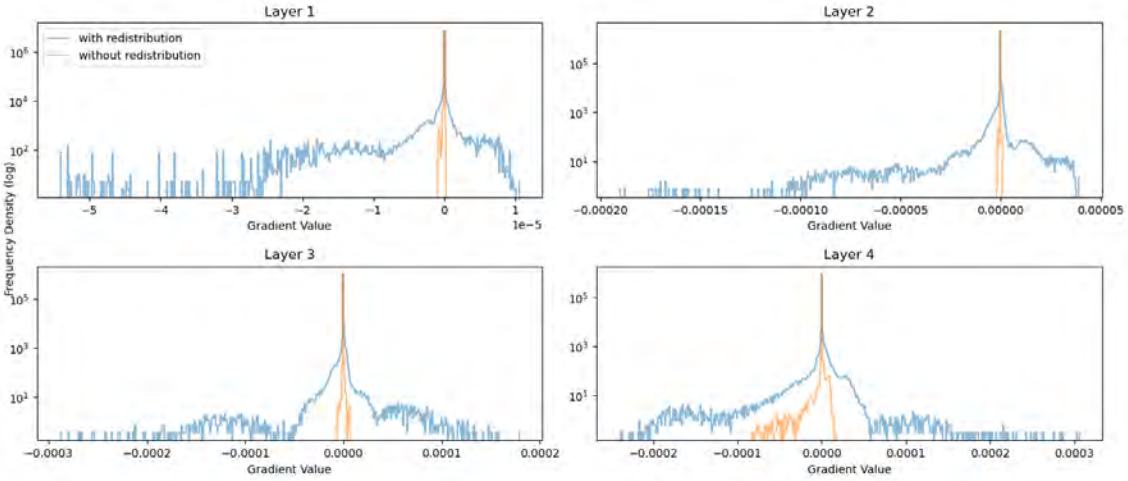


Figure 24 Gradient comparison showcasing vanishing gradients if no redistribution of admittance values is conducted in admittance-weighted layers. Source: own illustration

averted by usage of skip connections which will be featured in section 5.3.4. All applied transformations together are described in equation 20 with k as scaling factor and mmn as min-max normalization.

$$Y'_{i,j} = \begin{cases} 0 & \text{if } Y_{i,j} = 0, \\ \frac{\log(1+k\times \mathrm{mmn}(|Y_{i,j}|))}{\log(1+k)} & \text{otherwise} \end{cases} \tag{20}$$

Then, equation 19 is altered such that the adjacency matrix is replaced by the transformed admittance matrix like so $M_{i,j} = W_{i,j} \cdot Y'_{i,j}$. Finally, the weighted sum s_j of output node j is calculated by altering equation 7 by replacing the weights $W_{i,j}$ with masked weights $M_{i,j}$ as described in equation 21.

$$s_j = \sum_i x_i \cdot M_{i,j} \tag{21}$$

5.3.2 Expansion to multidimensional data

While pruning using weight multiplication may seem as easy as simple matrix multiplications, it increases in complexity when the multidimensionality of the input is considered. Each input feature that may have an influence on another feature needs to be added to the latter's weighted sum as depicted in equation 22 in which

- s is the weighted sum,

- i is the index of the input node,
- j is the index of the output node,
- n_1 to n_m represent the input features,
- k_1 to k_m represent the output features,
- X is the input activation matrix of the previous layer,
- W is the weight matrix.

$$s_{k_1,k_2,\ldots,k_m,j} = \sum_{i=1}^{i} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \cdots \sum_{n_m=1}^{N_m} X_{n_1,n_2,\ldots,n_m,i} \cdot W_{n_1,n_2,\ldots,n_m,i,k_1,k_2,\ldots,k_m,j} \tag{22}$$

In this work's use case, for each feature, i.e. voltages for each frequency and complex component, the connections need to be established if the respective nodes are connected to each other. Therefore, equation 22 is modified by adding batch index b, using masked weights M instead of weights W and replacing feature indices as shown in equation 23 with n_1, k_1, for frequency and n_2, k_2 for complex component for input and output respectively.

$$s_{b,k_1,k_2,j} = \sum_{i=1}^{i} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} X_{b,n_1,n_2,i} \cdot M_{n_1,n_2,i,k_1,k_2,j} \tag{23}$$

Additionally, due to the altered kernel shape the adjacency or admittance matrix needs to be further modified so that equation 19 holds. Moreover, if the custom layer is the first layer of the neural network, the input shape may not correspond to the shape of the admittance matrix in case measurements or replacement values are only available for selected nodes. Therefore, the known indices K need to be extracted from the admittance matrix as shown in equation 24.

$$Y'_{n_1,n_2,\ldots,n_m,i,j} = Y_{n_1,n_2,\ldots,n_m,i,j} \quad \text{for} \quad i \in K \tag{24}$$

Then, the modified matrix is repeated over m new axes where m is the amount of features so that its shape corresponds to the kernel shape.

Finally, an integer neuron scaling factor is included that allows for modification of neurons per layer by adding a dimension of size of the factor to the kernel. The code for equation 23 and 24 is shown in code snippet 1. Notably, each additional feature increase the matrix complexity and results in additional memory usage.

5.3.3 Physics-Informed Neural Networks

Physics-informed neural networks (PINN) leverage the information contained in equations describing the system to ensure adherence to physical laws. In section 4.1.3 existing literature regarding the incorporation of physical information into the neural network training process was reviewed. In this section, the applicability to power quality state estimation is discussed.

The generalized formula $R_{\text{phy}}(X, \hat{y})$ presented in 4.1.3, however, allows for consideration

of multiple physical laws. At first, Kirchhoff's voltage law is considered.

$$V_n = V_{\text{source}} - \sum_{i=1}^{n} V_{\text{drop},i} \tag{25}$$

For a path in a radial network of length n KVL can be described as shown in equation 25 where V_{source} is the voltage at the source or reference node and $V_{\text{drop},i}$ is the voltage drop across the i-th element in the path from the source to node n. The voltage drop $V_{\text{drop},i}$ is equal to $I_i \cdot Z_i$, with I_i as the current through the i-th element and Z_i as the impedance of the i-th element in the path. However, this requires either voltage drops or node currents I_i and impedances to be available as target data during training. Calculating the voltage drop by subtracting node voltage from source voltage $V_{\text{drop},i} = V_{\text{source}} - V_i$ obviously lets equation 25 hold true and therefore does not leverage any additional physical properties or relationships.

Power balance equations 4 and 5 use the same principle and thus also require either voltage drops or power injections. Therefore, the approach discussed in [34] is not applicable here without introducing additional parameters. Nevertheless, if calculated separately for each frequency and taking frequency-dependent impedances into account, PBE are generally also applicable for power quality analysis. Further, lack of available state variables also prevents the application of Ohm's law, equation 3, as physical law for PINN. While the admittance matrix is available, node currents are missing.

Hence, in future work saving additional system state variables solely for the sake of physical validation may be considered. However, using the known target values available from the simulation to validate the physical model during training arguably can only result in the target state as the only possible solution and thus a redundancy of the direct residuals.

All in all, it remains unclear whether a benefit is provided since the prediction of additional parameters also introduces uncertainties. Nevertheless, the cost of calculating additional parameters during simulation is low and therefore, in future work, PINN should be considered for power quality using PBE or Ohm's law.

5.3.4 Skip Connections

In deep neural networks vanishing and exploding gradients are issues that can occur due to poor weight initialization, activation function choice or deep model architectures. During backpropagation, the gradients are updated in such a way that gradients become close to zero, i.e. 'vanish', or become increasingly large, i.e. 'explode'. For deep architectures the issue is especially prominent since each layer contributes to the effect. Hence, in [57] the concept of residual networks (ResNet) was introduced. Layers are grouped into blocks, so-called residual blocks, for which the input is added to the output through a residual connection. Since this connection essentially skips the layers within the residual block, it is often called skip connection. As described in section 5.3.1, physics-aware neural networks require architectures as deep as the maximum distance of any node to the nearest measurement node. Moreover, manual pruning and weighting contributes to the vanishing

gradient problem as previously shown in figure 24. Thus, large power grids especially those with low measurement penetration or suboptimal placement are prone to the degradation problem. Further, in [53] it is shown that skip connections improve the convexity of the loss landscape. Therefore, skip connections were used in this work's architecture.

5.3.5 Measurement Errors and Noise Layers

Phasor measurement units (PMU) that monitor voltage or current magnitude and phase are an essential part of smart grids due to their ability to accurately capture the system state synchronized and in short time steps. However, PMU measurements are prone to errors and inaccuracies caused by internal device components. For those errors, more commonly referred to as noise, [58] shows that it follows a Gaussian distribution with zero mean in a steady state. However, more recent work suggests that real measurement errors are unlikely to follow a Gaussian distribution [59] [60] [61]. Even though the noise may follow a non-Gaussian distribution, the authors of [61] suggest that Gaussian distribution can still be a meaningful choice for error modeling. Additionally, modeling realistic errors whose relationships are only partially known and described in existing work is difficult. In [59] amplitude modulation, phase modulation and frequency ramp tests were performed for two different types of PMU. The standard deviation in all test cases was $\sigma < 0.01$ for phase angles and $\sigma < 0.005$ for magnitudes. Hence, for this work a Gaussian noise with $\sigma = 0.01$ will be used to model errors in the input data. Authors of related work [27] and [28] as well utilize Gaussian noise with $\sigma = 0.01$ in case of voltage magnitudes and $\sigma = 0.02$ for power flow to simulate measurement errors.

In traditional model-based algorithms like WLS, the weight matrix compensates for measurement noise by providing estimates of the accuracy for each measurement unit. Hereby, those points with high measurement accuracy are given less freedom than those with low accuracy. So that the neural network is capable of accurate predictions even in case of erroneous data, it needs to be trained with added noise as well. Hence, Gaussian noise layers are added to the network. Adding noise during training is a well-known tactic for improving a model's generalization capability, often termed "data augmentation". Therefore, a standard deviation $\sigma = 0.02$ for the Gaussian noise layer in this work's architecture was chosen. Results without and with measurement noise are presented in section 7.

Moreover, PMUs that shall accurately capture the system state need to be synchronized. In this project, a time signal is obtained through GPS. However, in case of unavailability of a signal synchronization errors might become an issue. Additionally, measurement data needs to be transferred from the measurement device to a centralized entity that combines all measurement data. Delays or communication errors on this path might introduce further errors that need to be considered in future work but are out of scope for this work.

5.3.6 Separability of Estimation

Especially for large grids, measures for reducing dimensionality of the estimation problem may be required due to constraints in memory and overall reduced performance. In [29] an interesting approach to dimensionality reduction is presented. The authors suggest

partitioning the grid into sections at measurement positions.

However, in the power grid evaluated in this work, shown in figure 12, the separation algorithm is not applicable if measurements points are set at the most distanced node to the transformer as in configuration a. If instead nodes at intersection points at the middle of the line are chosen as partition points, the grid can be separated into multiple parts. While measurement point configuration b in figure 12 is optimal to reduce the path length described in equation 18, the optimal configuration for partitioning is shown in figure 25. In total, the grid is partitioned into five sections at intersection points at

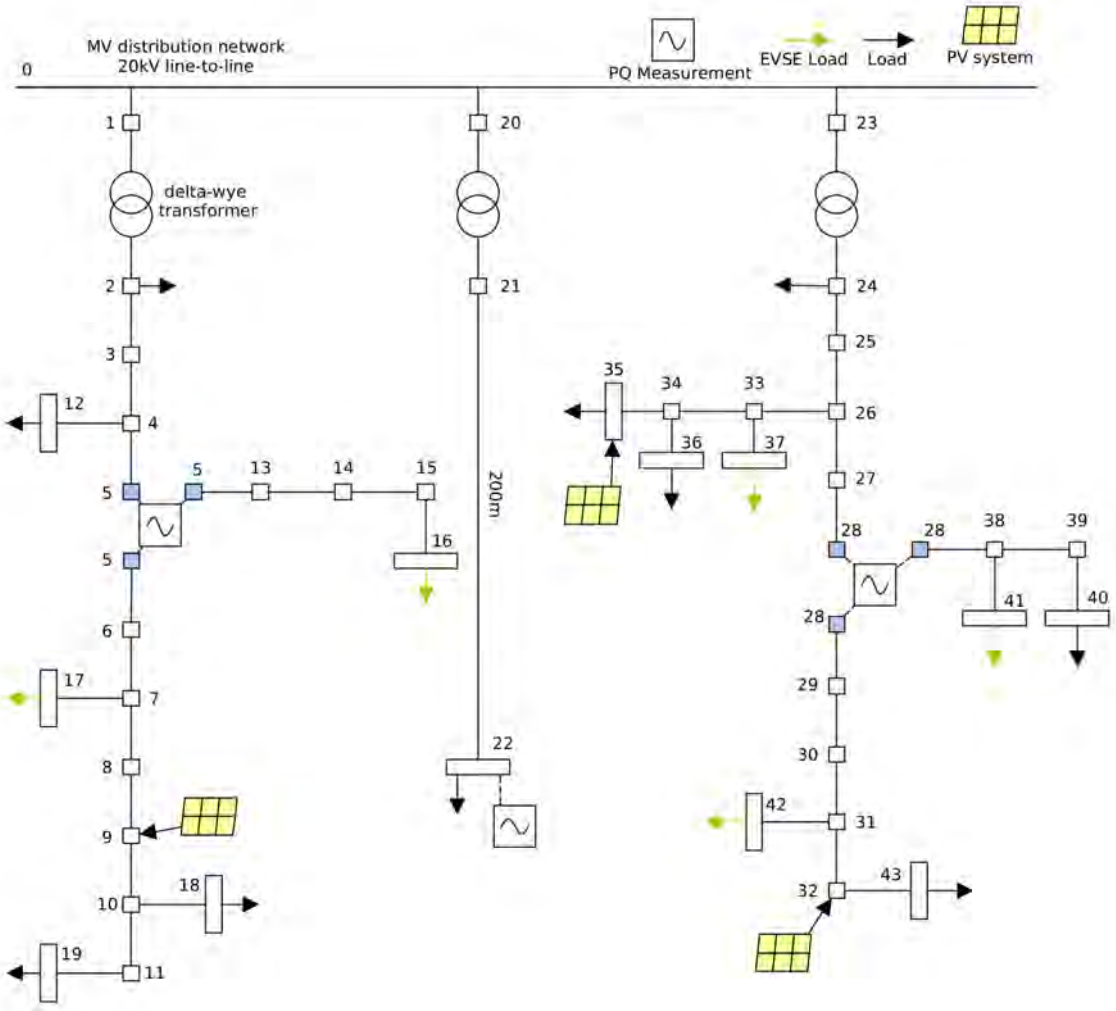


Figure 25 Grid partitioning at measurement nodes, marked in blue. Each section can be evaluated separately to reduce estimation dimensionality. Source: own illustration

nodes 5 and 28. Each section would be evaluated individually with the measurement point remaining in both sections. Then, for each section a neural network is trained. Logically, most information required to evaluate a single section should be contained in the adjacent measurement nodes. Nevertheless, further evaluation of this approach is required to ensure power quality issues spreading from one section to another would be accurately captured. Likewise, since during training every subnetwork only sees its respective nodes the ability to capture disturbance sources and the direction of their spreading may be hindered. By following this approach, the dimensionality of each section's neural network is greatly reduced. A similar approach with partitions at transformers could prove more useful for

power quality analysis since certain disturbances are dampened across transformers as described earlier.

5.3.7 Model Architecture

Taking all aforementioned considerations into account, a network architecture was derived that is presented in 26. Here, required layers are marked with full lines while optional

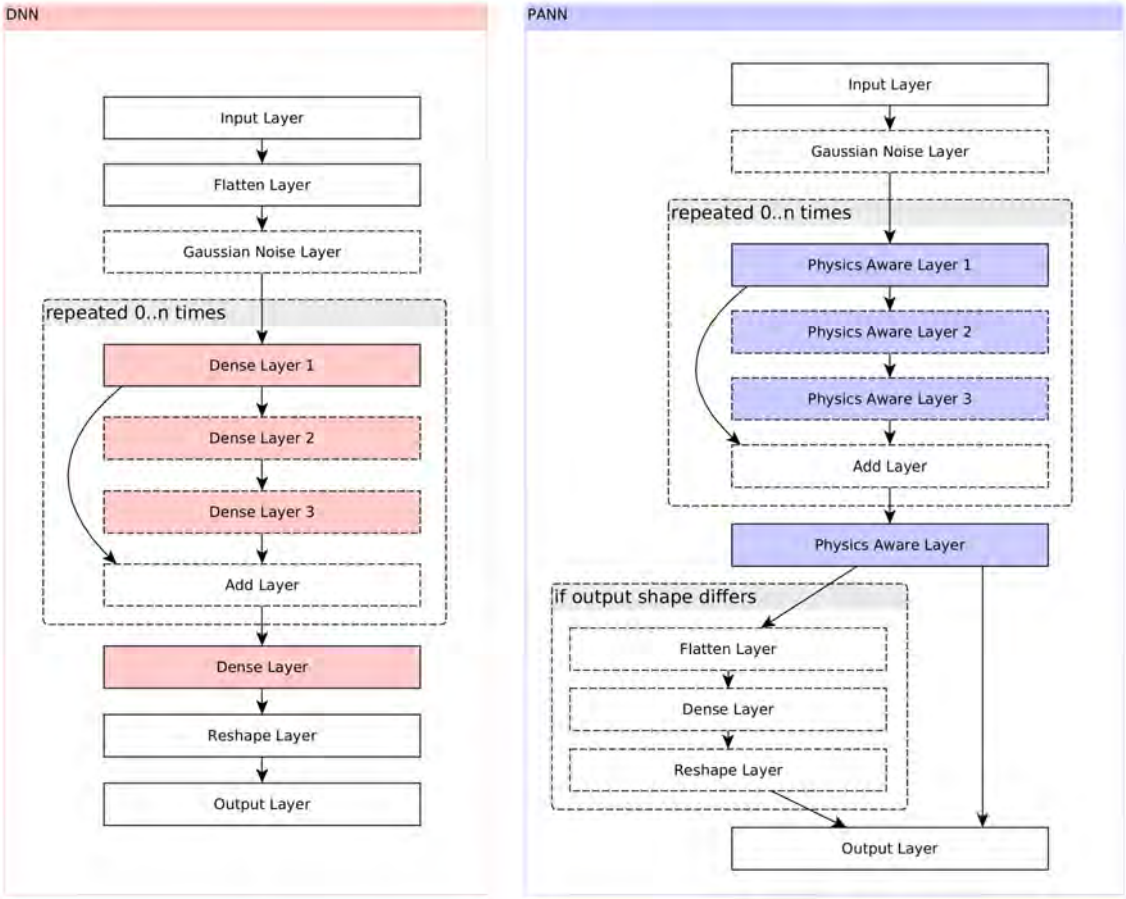


Figure 26 Chosen network architecture for dense neural networks (DNN) and physics-aware neural networks (PANN). Layers with dashed lines are optional. Source: own illustration

layers are marked with dashed lines. In case of DNN the input is immediately flattened and an optional Gaussian noise layer is added. Then, residual blocks can be repeated an arbitrary amount of times or be fully left out. Since the first dense layer transforms the input into output shape, any input shape is accepted. Finally, after a mandatory dense layer, the output is reshaped back to its original multidimensional form and passed to the output layer. PANN architecture is built similarly, however, flattening the input is undesired. Instead, the input is passed to the first physics-aware layer in its original multidimensional form since the spatial relationship needs to be preserved. Hence, after an arbitrary amount of residual blocks and a mandatory physics-aware layer, the output may need to be reshaped. This is only the case if the output shape differs from the input shape, for example under usage of hybrid complex representations, no precalculations or a neuron scaling factor other than 1.

6 Architecture

This chapter describes system requirements and the resulting architecture design choices. The overall goal was to develop a framework that allows a comparison between different models, power grids and with various combinations of hyperparameters. The architecture was developed with considerations for functional and non-functional requirements listed in the following chapters.

6.1 Functional Requirements

Firstly, data generation shall be available for different power grids and with different load profiles and spectra attached. Secondly, in data preprocessing the framework shall clean, normalize and transform raw data into an optimal format for training and as well be as agnostic as possible to the underlying power grid. Next, the data shall be split into training, test and validation set with an option to either use portions of the same generated data for all sets or read in data generated with different parameters such as harmonic spectra. This requirement is based on the fact that the digital clone of the power grid will never perfectly reflect the grid conditions and parameters. Hence, the model shall be validated and tested on its generalization ability using real world data or data generated with different parameters. Consequently, the system shall be capable of supervised training including an easy way for tuning hyperparameters to optimize the model. Finally, the framework shall offer methods for comparable evaluation of different models.

6.2 Non-Functional Requirements

Several non-functional requirements stem from considerations for performance, software sustainability and limitations of large systems or systems that process big data. In terms of software sustainability, the framework shall be reusable for different use cases. This is accomplished through modularity with well-defined interfaces, code documentation and version control. Secondly, model performance is crucial in machine learning projects. Since time series data for long periods is created, creation must be optimized in terms of performance and storage efficiency. Memory management is critical due to the high dimensional input data and requirement for deep networks especially in case of PANN. Moreover, the system shall be scalable to accommodate for grids with more nodes or input dimensionality. Finally, the framework shall be capable of operating with existing power grid representations that are used by DSO.

6.3 Implemented Architecture

In the program this work is based on, all tasks were initially performed within the same module. While this simplifies usage in a singular use case, reusability for different projects or even for different input data is hindered. Thus, packages were split at those points, where a well-defined output is produced. In most cases, this output can be saved to disk and is human-readable. Figure 27 shows an overview over the developed packages. Inputs and outputs as text file are shown in green, as CSV file in orange and object-based in

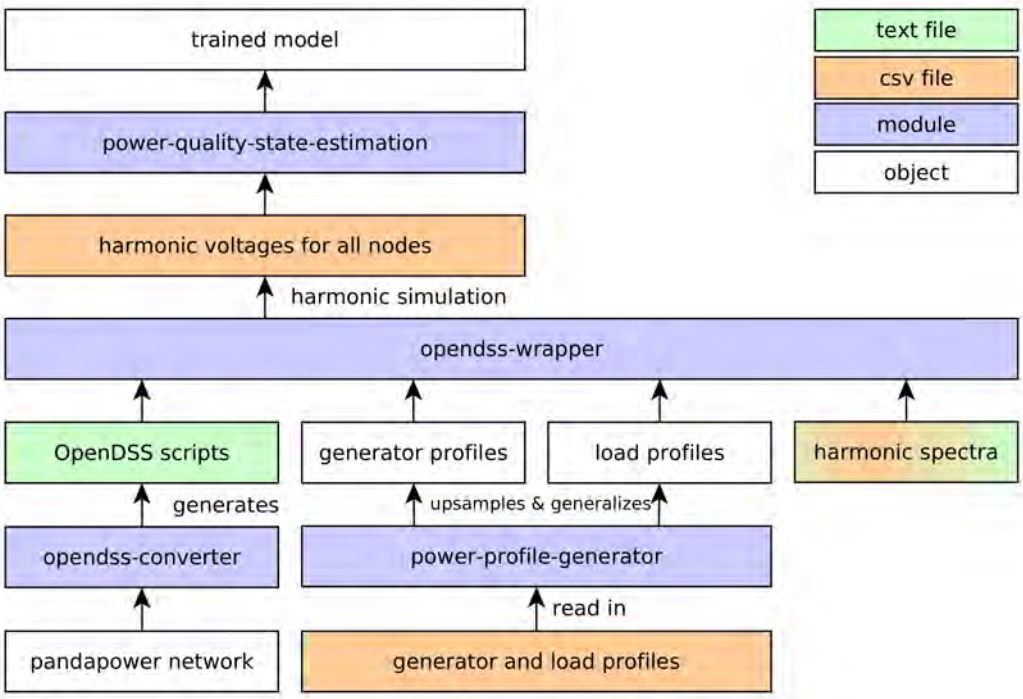


Figure 27 Overview of packages, file-based and object-based inputs and outputs. Source: own illustration

white. Harmonic spectra require both a CSV file with rows for each frequency and associated value and a text file to associate the spectrum with a specific component in OpenDSS. Since most inputs and outputs of the developed system are file-based, it is crucial to ensure all required files are available and in the correct format as specified in OpenDSS [17] or by the respective interface. Hence, input validation is performed whenever deemed necessary.

In the module *opendss-converter*, power grids are converted from *pandapower* to OpenDSS scripts. Furthermore, each load and each generator gets assigned with load profiles and harmonic spectra. The load profiles are converted from existing datasets into a common format in *power-profile-generator*. This package is designed to be expanded in the future so that load and generator profiles can be generated according to individual requirements. The *opendss-wrapper* module is an optimized and parallelized wrapper around OpenDSS power flow and harmonic flow solver that allows simulation of power grids under usage of the previously generated scripts, load profiles and harmonic spectra. Additionally, not depicted in the overview above, a toolbox package was developed for highly repeated tasks. This toolbox includes multidimensional scalers, logarithmic distribution methods and specific file operations including input validations. The class diagram for developed scalers is shown in figure 28. Those scalers, implement a common interface with three important methods. In *fit_ transform* the data is transformed along axes given as parameters and the normalization parameters are stored in the scaler object. Then, *transform* method allows scaling of different data of the same dimensions along the same axes using the previously saved normalization parameters. Finally, in a *revert* method, the normalization is reverted on the given data.

The last module is *power-quality-state-estimation* in which the simulated data is used as

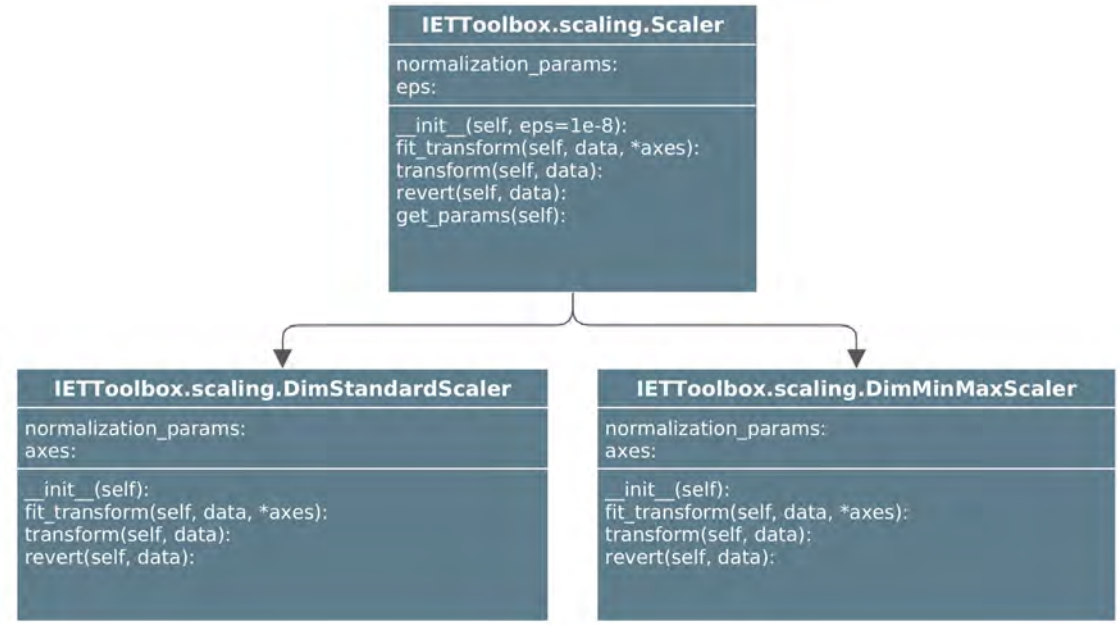


Figure 28 Class diagram of multidimensional scalers that scale data along specified axes. Source: own illustration

training data for different neural network models. Here, custom layers and models for power quality state estimation using PANN are implemented. A class diagram for custom models and layers is shown in figure 29 with some redundant attributes omitted. The specific model, that in all cases always inherits from the parent *DNNComplex*, is initialized with a dictionary of hyperparameters and network data. Both are then validated and saved within the parent class for later usage. In its training method, a callback is used to optionally save model weights after every selected amount of epochs. Moreover, saved weights can be re-used to save training time by providing a path to the weights and instructing the model to load saved weights. Easy model creation with different layer architectures is simplified by inheriting from the parent class. Because of that it is only required to implement those methods with custom changes. For example, PANN implements the *create_layers* method in which the custom physics-aware layers are called instead of simple dense layers. As described earlier, a dense layer is added at the end if the architecture requires it due to different output than input shapes. In contrast, the hybrid PANN adds at least two hidden layers after an arbitrary amount of physics-aware layers.

In case of PANN, a custom layer needs to be provided. Here, the options are *AdjacencyPrunedLayer* or *AdmittanceWeightedLayer* which both inherit from *AdmittanceLayer* which itself inherits from Tensorflow Keras *Layer* class. These custom layers require additional parameters including the admittance matrices for each frequency, a threshold in case of adjacency pruning for cutting of admittance values close to zero and a redistribution factor in case of admittance weighting. The redistribution factor is equal to k in logarithmic transform equation 20. Further, the neuron scaling factor is passed to physics-aware layers so that the matrix multiplications can be adjusted according to the new dimensionality. Moreover, the known elements, i.e. measurement nodes, are passed to the custom layers. This allows for selecting the correct nodes from the admittance matrices in the first layer

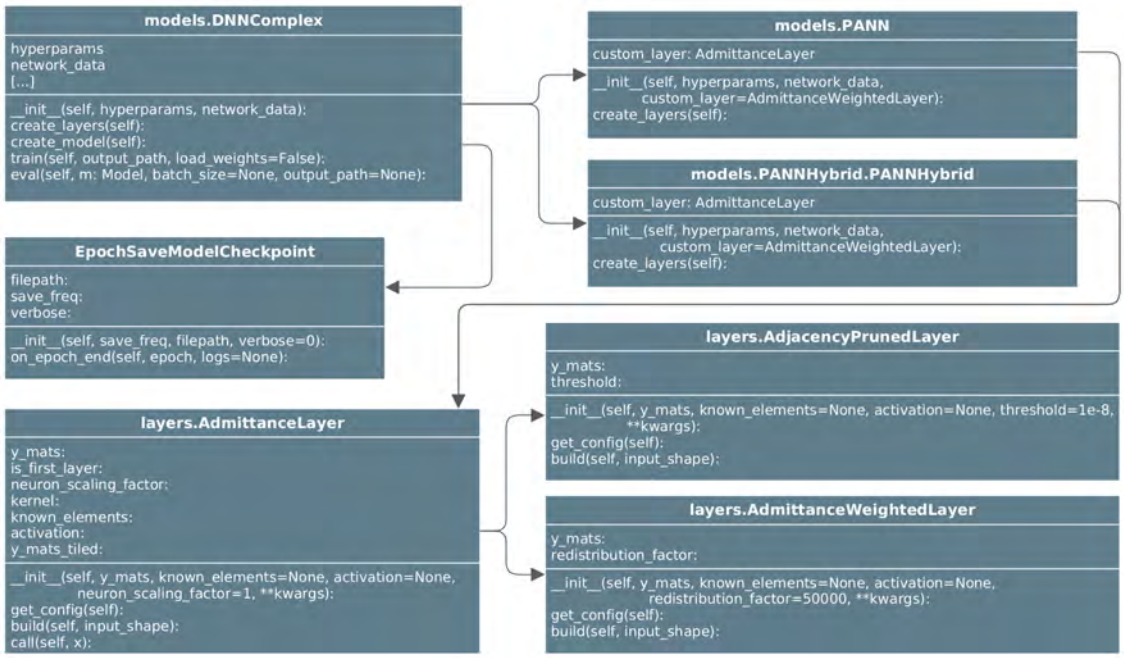


Figure 29 Simplified class diagram of custom models and layers. For better representation, some attributes were omitted. Source: own illustration

of the network.

In order to provide an easy way of hyperparameter tuning, hyperparameters are provided in a dictionary to the model and are automatically used at the appropriate spots. Table 6 gives an overview over the adjustable hyperparameters. Even though the amount of layers

Key	**Data type**	**Range**
batch_size	integer	[0..∞]
amount_of_layers	integer	[0..∞]
dropout	float	[0..1]
loss_function	string / function	
activation	string / function	
optimizer	string	'adam' / 'sgd'
learning_rate	float	[0..1]
neuron_scaling_factor	integer	[1..∞]
epochs	integer	[0..∞]
skip_connections	boolean	
gaussian_noise	float	[0..1]
batch_normalization	boolean	
callbacks	list	

Table 6 Hyperparameter data types and ranges

can be set here, additional layers may be added if required, for example due to different output than input shape. Dropout is only applied in case of DNN since PANN are already regulated through manual pruning. Further pruning poses the danger of worsening the vanishing gradient problem due to the sparseness of connections. The parameter *callbacks* allows specification of callback methods that are then used in the model training at each epoch. Here, early stopping and reducing learning rate on plateaus are the most common

examples.

Additionally, the input data can be customized in a preprocessing step. The possible parameters are shown in table 7. Here, *pre_ calculations* is an optional preprocessing step in

Key	**Data type**	**Description**
grid_name	string	folder name of simulated data
scaler	class	implementation of dimensional scaler
data_length_in_min	integer	[0..∞]
step_width_in_min	integer	< data_length_in_min
pre_calculations	boolean	
complex_repr	string	complex representation
test_size	float	[0..1]
validation_size	float	[0..1]

Table 7 Input data parameters and descriptions

which approximate values for all nodes are calculated from the available measured input nodes using a conventional algorithm. Even though these values are mostly inaccurate, they simplify network architecture designs by allowing the same input and output shape. Validation and test size only has an effect if the train-test split is done within the model. Optionally, a separate validation and test set can be provided by saving those sets in accordingly labeled folders at the same level of the training data.

7 Results

This section provides results of model performance with regard to different architectures, measurement point configurations and hyperparameters. After an initial grid search, conducted to gain an idea of which hyperparameters and architectures perform best, the optimal measurement point configuration is described in section 7.1. Then, section 7.2 shows results for different complex value representations. Subsequently, the model's resilience towards measurement noise is tested in section 7.3. Further parameters including skip connections and precalculations are evaluated in section 7.4 and the distribution of errors across features is shown. Finally, the application of predicted values to power quality is described in section 7.5 by providing absolute values in the original scale.

In an initial grid search, it was confirmed that DNN models with 2 layers perform better than those with 1, 3, 4 or 6. Moreover, a learning rate around 5×10^{-4} was identified as best-performing. Because of that, in the following experiments a triangular cyclic learning rate with a minimum value of 1×10^{-5} and a maximum value of 1×10^{-4} was chosen. By cyclically changing the learning rate, the model is capable of jumping out of local minima. Even though more experiments with different neuron scaling factors could prove useful, it was always set to either 1 or 2 dependent on complex representation so that the amount of parameters in the network remains roughly equal. Hereby, models are more comparable since increased amount of parameters typically allows capturing more complex relationships. Additionally, a fixed batch size of 8192 was chosen based on the maximum available GPU memory for deep PANN architectures.

7.1 Optimal Measurement Point Configuration

Table 8 shows a comparison of measurement point configurations as described in figure 12 on model losses for data in Cartesian form over 600 epochs. Even though equation

Config. a		**Config.** b	
DNN	**PANN**	**DNN**	**PANN**
1.3×10^{-5}	2.9×10^{-5}	1.4×10^{-5}	1.9×10^{-5}

Table 8 MSE validation losses for measurement points at the middle of each section (configuration b) and at the most distanced node from the transformer in each section (configuration a) for different models.

18 dictates 10 layers for PANN with measurement points in configuration a, i.e. at the most distanced node from the transformer in each section, a previous grid search showed a better performance with only 8 layers. Likely, the fully connected dense layer that is required due to the different output than input shape at the end, compensates for the missing layers and thus the model benefits from the reduced complexity. One can see that while DNN perform similarly in both cases, PANN with fewer layers due to optimized measurement placement in configuration b perform better with an MSE validation loss of 1.9×10^{-5} than their deep counterpart with a loss of 2.9×10^{-5}. PANN with measurement points at the middle of each section require 6 layers as shown in equation 18 and 2 layers were chosen for DNN. In another grid search, it was confirmed that 6 layers is

optimal for measurement point configuration b with 5.9×10^{-6} final training loss after 1000 epochs. However, 8 layer and 4 layer architectures performed only slightly worse with 6.1×10^{-6} and 6.2×10^{-6} final training loss respectively. Therefore, configuration b with measurement points at the middle of each section was chosen for the following experiments.

7.2 Ideal Complex Representation

DNN			PANN		
Cartesian	**Expon.**	**Hybrid**	**Cartesian**	**Expon.**	**Hybrid**
1.4×10^{-5}	2.13×10^{-4}	1.4×10^{-5}	1.9×10^{-5}	2.49×10^{-4}	1.5×10^{-5}

Table 9 MSE validation losses for DNN and PANN in different complex representations over fixed 600 epochs.

Table 9 shows the losses of DNN and PANN for different complex representations over a fixed amount of 600 epochs. If a hybrid complex representation is chosen in PANN the amount of neurons is doubled in each layer. So, to remain with comparable amounts of parameters, the other models' amount of neurons was doubled by setting the neuron scaling factor to 2. Exponential representation performed significantly worse than Cartesian and hybrid representation. However, combining exponential with Cartesian in a hybrid representation resulted in an improvement in PANN. PANN achieved comparable albeit not superior results to a simple DNN. However, previous model runs and the abovementioned grid search had shown that PANN tend to learn more slowly, possibly due to the increased model complexity. This was confirmed in two test runs over 1400 and 3000 epochs in hybrid complex representation, where the minimum MSE loss for PANN significantly improved to 7×10^{-6} and then 4×10^{-6} while the minimum MSE in case of a DNN only slightly improved from 1.4×10^{-5}, also at 1400 epochs, to finally 1.1×10^{-5} after 3000 epochs.

7.3 Effects of Input Noise and Gaussian Layers

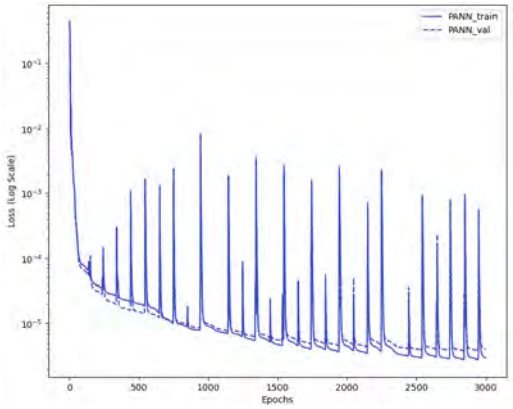
Next, the model's capability of dealing with measurement noise was tested by adding a noise with $\sigma = 0.01$ to the test data. Then, a Gaussian layer with $\sigma = 0.02$ was added to the model input. The results of two runs each performed with the same parameters are shown in table 10 for PANN trained over 3000 epochs. On accurate input, the loss without

	Accurate Input		Input Noise	
Trial run	**-**	**GaussianLayer**	**-**	**GaussianLayer**
1	5.45×10^{-6}	6.52×10^{-6}	8.58×10^{-6}	8.04×10^{-6}
2	5.75×10^{-6}	6.48×10^{-6}	8.93×10^{-6}	8.06×10^{-6}

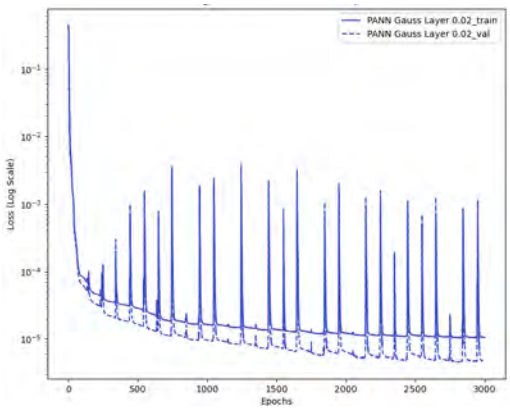
Table 10 Minimum MSE loss on the validation set with and without input noise and Gaussian layer with $\sigma = 0.02$ in two trial runs of physics-aware neural network models trained over 3000 epochs.

Gaussian layers was lower which may be caused by the introduction of uncertainties to

the network. However, Gaussian layers improved the model's capability of dealing with Gaussian noise on the test data as can be seen by the lower loss in the rightmost column in table 10. Figure 30 shows the training and validation loss for PANN with and without



(a) Training and validation loss of PANN



(b) Training and validation loss of PANN with Gaussian Layer $\sigma = 0.02$

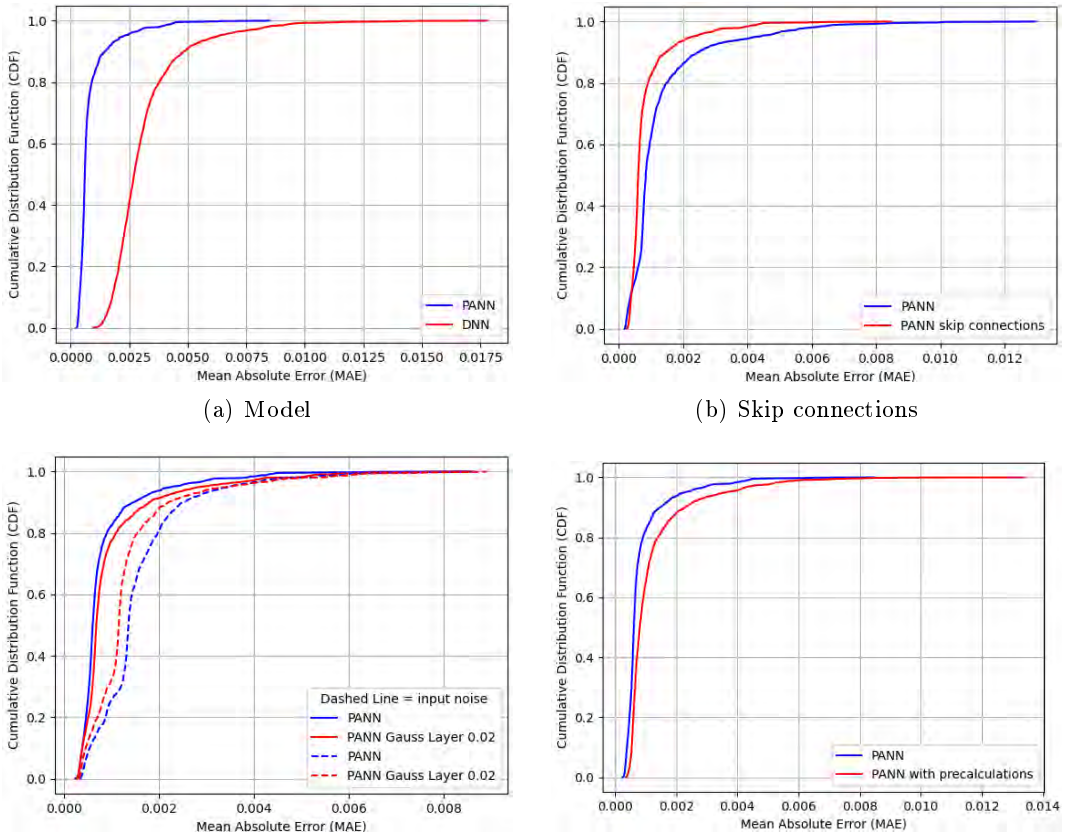
Figure 30 Comparison of training and validation loss of a physics-aware neural network (PANN) model with and without Gaussian layer. Source: own illustration

Gaussian Layers. Temporary spikes in the loss curve are caused by the triangular cyclic learning rate. One can see that for the model trained without Gaussian layers, the training loss decreases at the same rate as the validation loss, eventually even falling below. This might be a first indication of model overfitting. On the other hand, the training loss of the model trained with Gaussian layers decreases more slowly than the validation loss. This indicates the model's capability of generalizing from the training data.

7.4 Further Parameters and Error Distribution

The cumulative distribution function (CDF) of mean absolute errors, as shown in equation 9, for normalized data for different models is shown in figure 31. In the plot, the CDF at a given error x quantifies the fraction of data points with an error value less than or equal to x. A CDF curve that rises steeply towards 1 is desirable, as it indicates that the majority of errors are of small magnitude. If the curve extends significantly to the right on the x-axis, it suggests the existence of outliers with larger errors, which is typically not desirable.

In figure 31(a) a comparison between a DNN and a PANN model was conducted. Here, the DNN model was configured with a dropout value of 0.02 in each layer, which proved as a successful regularization mechanism to prevent the frequent occurrence of exploding gradients. Throughout multiple test runs, exploding gradients only occurred for DNN models, especially those with deep architectures, and for PANN models with precalculations. Potentially, the sparseness of input data and the pruning of connections functions as an effective regularization technique. In future work, additional regularization mechanisms such as gradient clipping, L2 or L1 regularization can be tested and compared to the existing models. Nevertheless, DNN exhibits a higher probability of bigger errors on



(a) Model (b) Skip connections

(c) Gaussian layers and input noise (d) Precalculations

Figure 31 Cumulative distribution function (CDF) of MAE of normalized data for different models. Source: own illustration

the test set than the PANN model.

On the other hand, figure 31(b) shows a more desirable curve if skip connections are applied to the model. Likewise, the minimum validation loss of PANN with skip connections was at 4×10^{-6} while without skip connections a minimum validation loss of only 8×10^{-6} was achieved.

In 31(c) different models were evaluated with and without input noise and with and without Gaussian layers. Expectedly, for input data without noise the probability of small errors is higher than for input data with noise. The probability of higher errors increases when Gaussian layers are added to the model without input noise on the test set. Conversely, it decreases when Gaussian layers are incorporated along with noise on the data.

Finally, figure 31(d) shows the evaluation of the effect of precalculations, described at the end of section 6.3, on the input data. One can see that the model performs worse with precalculations of the input data. Even though in that case inputs are available even for nodes without measurements, those inputs are inaccurate. A minimum validation loss of 6×10^{-6} was achieved compared to 4×10^{-6} for a model without precalculations.

Overall, one can see that across the whole test set the probability of MAEs below 0.002 is greater than 80% for the best-performing models.

Figure 32 shows a heatmap of the MAE over frequencies and nodes in the grid for normal-

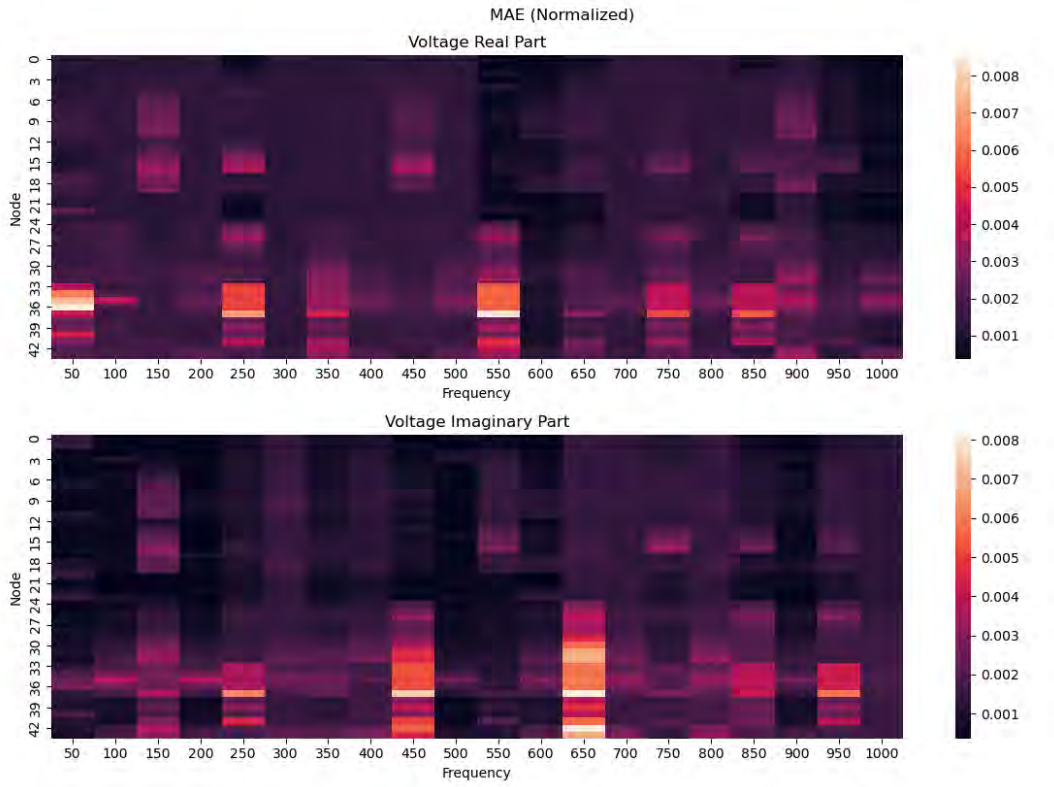


Figure 32 Heatmap of MAE for normalized data in a physics-aware neural network. The y-axis shows nodes in the power grid and the x-axis shows frequencies in Hertz. Source: own illustration

ized data. The upper plot shows the real part of complex values while the lower plot shows

the imaginary part. Due to the separate min-max normalization for each input feature, the network is incentivized to minimize the error for each frequency and node independently of the value ranges. Nevertheless, those nodes that are distant to measurement points remain more difficult to estimate. This is particularly visible for nodes 33 to 37 that lie in the commercial subnetwork.

Moreover, those frequencies where different harmonics are injected in the train and validation set especially challenge the model's generalization ability. As shown in section 5.1.2, in this work, only uneven harmonic spectra at those nodes with EVSE were altered and thus the highest errors are also visible there. This issue becomes even more visible when looking at the maximum absolute error in the normalized data for each frequency and node as shown in figure 33. Here, the highest error occurs at the 13-th harmonic at node 37 that

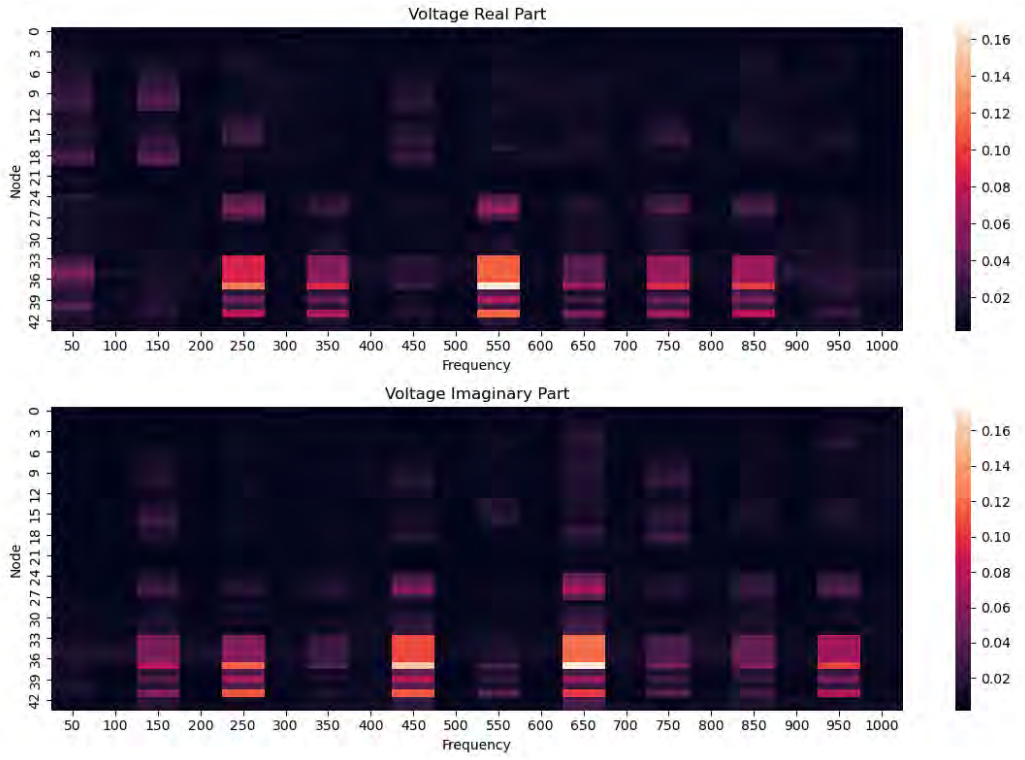


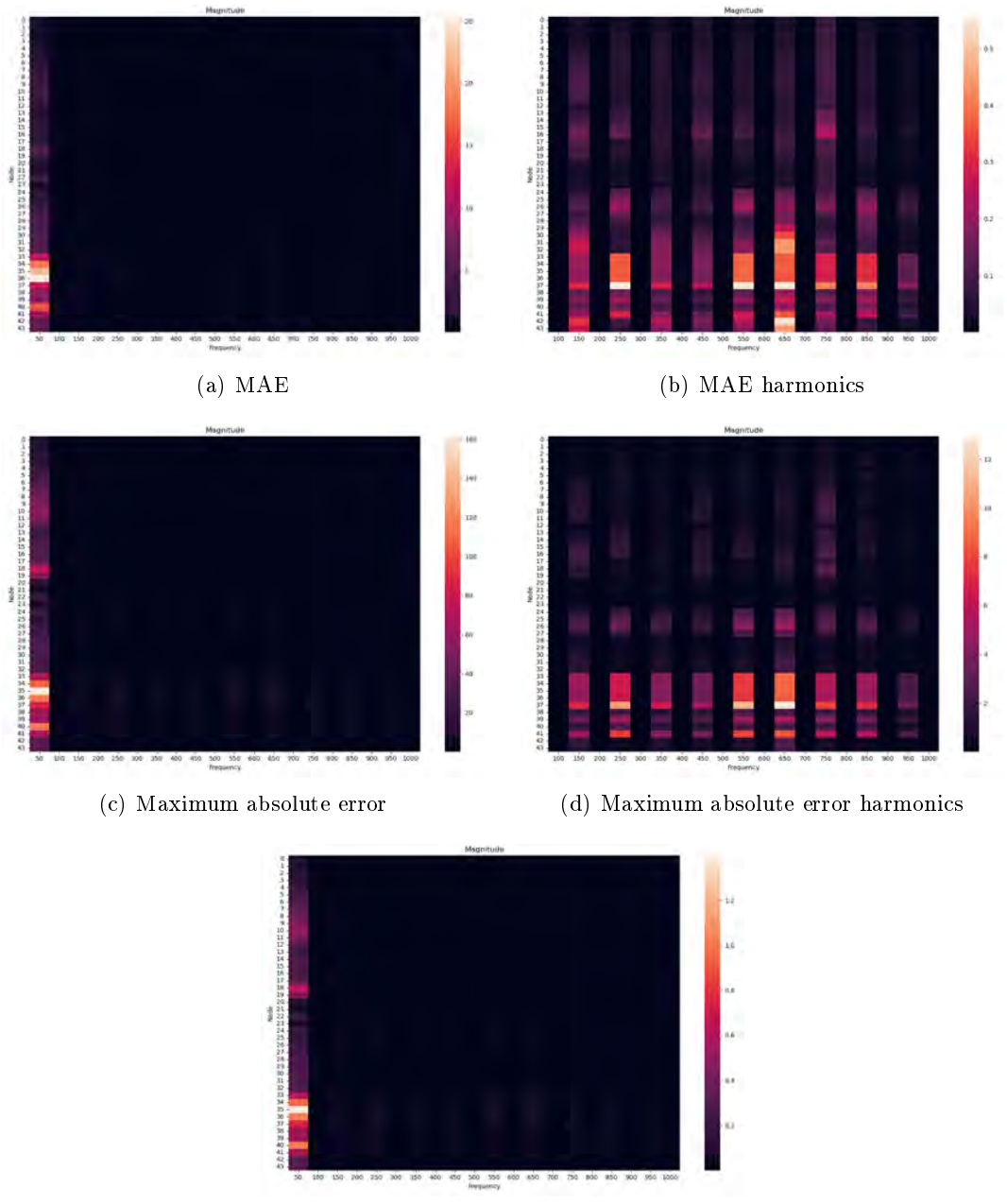
Figure 33 Heatmap of maximum absolute error of each node and frequency (in Hertz). Normalized data for physics-aware neural network model. Source: own illustration

is equipped with an EVSE load. Looking at the harmonic spectrum table 5 reveals that the difference between those values used in the respective sets is particularly high for the 13-th harmonic with a magnitude of 0.4 on the test and validation set and magnitude 1.9 or 4.6 on the training set. This underscores the necessity for usage of harmonic spectra in the simulation that accurately represent the real world conditions.

7.5 Application of Predicted Values to Power Quality

Plots utilizing normalized data such as figures 32 and 33 show the error relative to the respective minimum and maximum value for each feature. Hereby, one gains insight into

the approximation and generalization ability of the neural network. However, it is also important to take a look at values in their original scale. In table 2 the maximum allowed voltage distortions relative to the RMS value of the fundamental frequency were shown for each harmonic. Power quality state estimation should be able to estimate values precisely enough that limit violations are recognized and false positives avoided. To visualize values



(a) MAE

(b) MAE harmonics

(c) Maximum absolute error

(d) Maximum absolute error harmonics

(e) Maximum absolute error relative to RMS voltage at 50 Hz (in%)

Figure 34 Mean absolute error (MAE) and maximum absolute error of magnitudes after reverting normalization. Source: own illustration

on their original scale, min-max normalization is reverted and the magnitude errors are plotted in figure 34. Notably, all input data is still scaled to the nominal voltage level of the medium voltage section, 20 kV. This was necessary to simplify the admittance matrix of a network that possesses nodes in different voltage levels. Hence, line-to-neutral

voltage magnitudes at the fundamental frequency reside around $20\,\mathrm{kV} \times \sqrt{3} \approx 11.547\,\mathrm{kV}$ RMS value. In the appendix, a comparison of actual and predicted values at node 37 for a randomly selected time step is shown in table 11 allowing a first insight into the scale of values.

Figure 34(a) shows the magnitude MAE over all frequencies and nodes. Expectedly, the error is largest for the fundamental frequency with the highest MAE at 25.352 V at node 36. Due to the vastly different scales, a plot excluding the fundamental frequency and instead only showing harmonics is depicted in figure 34(b). Here, the highest MAE occurs at node 37, a node with an EVSE load, at frequencies 250 Hz, 550 Hz and 650 Hz with values 0.5563 V, 0.5536 V and 0.556 V, respectively.

In figure 34(c) the overall maximum absolute error on the test set is shown for each feature. Here, the maximum error at the fundamental frequency amounts to 161.5 V or 1.4% of the RMS value and can be found at node 35, that is equipped with a PV system. For harmonics, shown in figure 34(d), the maximum absolute error once again occurs at node 37 at 650 Hz with a magnitude of 12.94 V or 0.11% of the RMS value of the fundamental frequency. Table 2 had shown that for the 13-th harmonic a relative amplitude distortion of 3% is within the permitted limits. Finally, figure 34(e) shows the maximum error as percentage of the RMS voltage for each feature. However, this should also be evaluated relative to the absolute maximum values in the test set. For the 13-th harmonic this amounts to 75.35 V or 0.65% of the RMS voltage at the fundamental frequency. Nevertheless, one can see that the state estimation algorithm developed in this work consistently achieves estimation accuracies in different orders than the allowed deviations.

8 Summary

The purpose of this study was to explore the capabilities and limitations of physics-aware neural networks (PANN) in the context of power quality state estimation. The results highlight the advantages of PANN over traditional dense neural networks (DNN) and showcase solutions to challenges introduced by increased model complexity. PANN offer advantages by harnessing additional information contained in the physical structure of the power grid and incorporating it into the network architecture. While models with fully connected layers can pass messages to any node, PANN only pass messages to nodes adjacent in the underlying power grid which serves as an effective regularization mechanism.

For predefined frequencies, the neural network models were trained to estimate complex voltages of all nodes in a grid based on a sparse input with only three nodes as measurement points. Figure 35 shows estimations at 650 Hz of a DNN and a PANN model for a

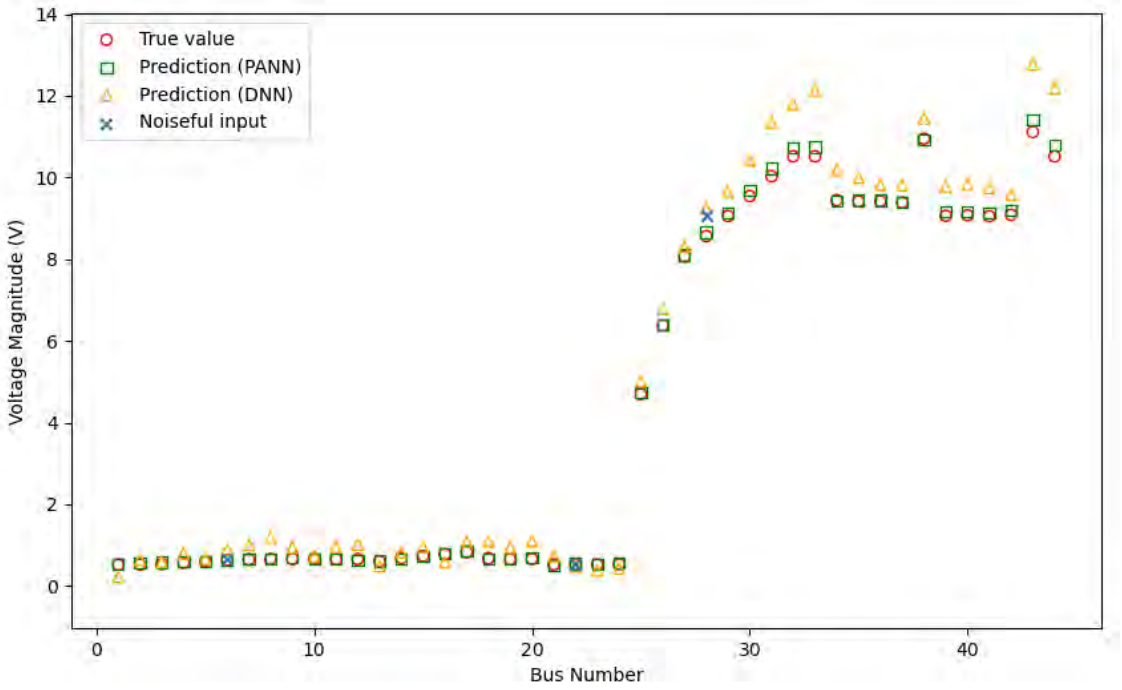


Figure 35 Harmonic predictions at 650 Hz for a randomly chosen iteration. Harmonics are predicted in the same fashion as the fundamental frequency with physics-aware neural networks (PANN) showing a higher accuracy than dense neural networks (DNN). Source: own illustration

randomly chosen time step.

For complex values, various representations were evaluated with a hybrid form slightly outperforming a standard Cartesian form with an MSE validation loss of 1.5×10^{-5} compared to 1.9×10^{-5} after fixed 600 epochs.

Even though PANN suffer from increased model complexity, effective ways of mitigating issues caused by deep architectures and sparse connections were shown. Partly, the disadvantages can be compensated for by sophisticated algorithms for measurement point selection and by intelligent design of neural network models, especially through skip connections. Here, it was shown that measurement points are optimally positioned in the middle of a grid section. Hereby, fewer layers are required in PANN resulting in a smaller

MSE loss of 1.9×10^{-5} over 600 epochs compared to 2.9×10^{-5} for suboptimal placement.

Further, it was shown that the model is fairly resilient towards input noise caused by measurement inaccuracies with an average minimum MSE loss of 8.8×10^{-6} compared to 5.6×10^{-6} with no input noise over two runs with 3000 epochs. Figure 36 shows the predictions of PANN and DNN at the fundamental frequency for a randomly chosen time step. Even with significant noise added to input values, the PANN model is able to predict actual values accurately. Adding Gaussian layers not only improved the model's capability

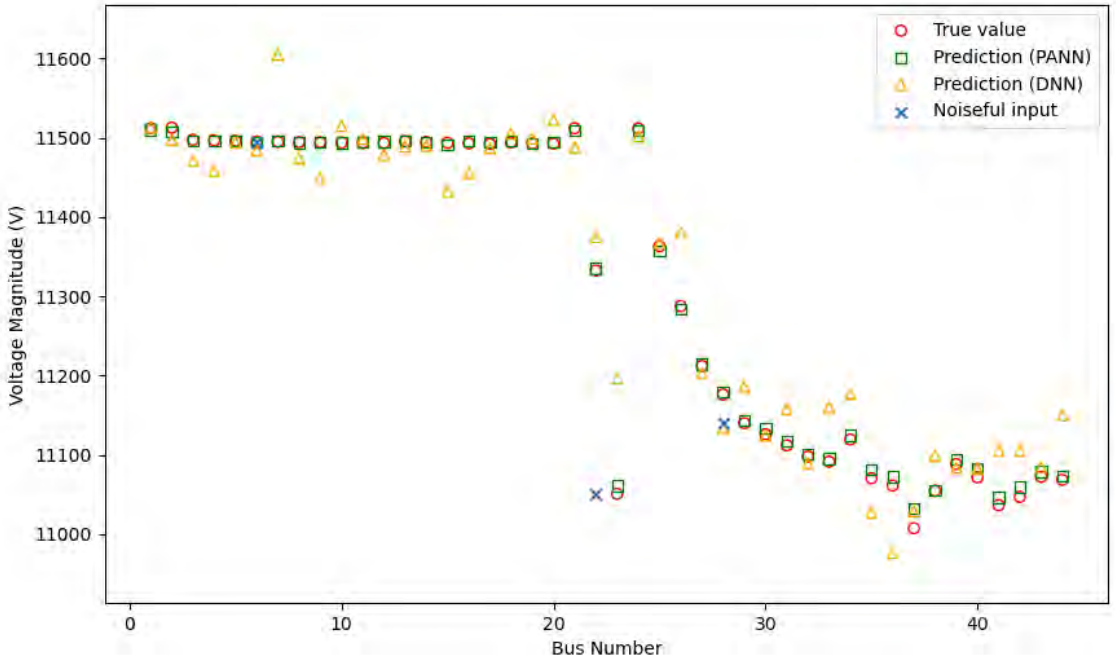


Figure 36 Predictions for a single randomly chosen iteration of the test set at 50 Hz. Physics-aware neural networks (PANN) and dense neural networks (DNN) predict values for all nodes based on inputs at three nodes with added measurement noise. Source: own illustration

to deal with input noise, resulting in an average minimum MSE loss of 8.05×10^{-6} over two runs, but also enhanced its generalization ability.

Even though computational complexity of PANN is higher, their overall performance exceeds that of DNN by a significant margin with a minimum validation MSE loss of 4×10^{-6} compared to 1.1×10^{-5} over 3000 epochs.

8.1 Outlook

This work opens multiple avenues for future research in power quality analysis using neural networks. Firstly, the simulation used to generate training data can be improved by increasing variance of the generated data so that more possible system states are covered. Here, time- and load-dependent spectra and more diversified load profiles with different loads for each phase are especially promising. Secondly, adding more features to the neural network can be considered. Demographic data of occupants, such as age and occupation give an insight into their power consumption patterns. Weather information can be used to better estimate power generation based on solar energy and energy consumption for

heating purposes.

Subsequently, time series data or data associated with time stamps may further help show intricate relationships that are time-dependent. This could be supported by implementing Long Short-Term Memory (LSTM) networks that are designed to capture long-range dependencies in input data. Additionally, complex valued neural networks that deal with the high correlation of values in split complex representations provide a promising avenue.

Aside from implementing additional models, the existing models can be further improved. Here, implementing physical laws in physics-informed neural networks offers further constraints on the neural network that help guide its output towards physically possible states. The increased complexity in PANN can be tackled in various ways. First, pruning connections instead of multiplying weights helps reduce amount of trainable parameters, however, is a highly complex task. Furthermore, estimating only those frequencies and nodes where breaches of limits are to be expected helps reduce the dimensionality and input size. Here, critical nodes and frequencies can be identified individually for each grid based on installed loads and generators. Moreover, principal component analysis helps break down highly dimensional input but is not applicable to PANN since those require preserving the dimensionality for connection pruning. Partitioning the power grid into individual sections as proposed in section 5.3.6 also helps reduce dimensionality albeit requiring development of neural networks for each section.

Due to time and model complexity constraints this work focussed on harmonics in the range of 50 Hz to 1 kHz. To fully capture all power quality disturbances a broader range also including inter-, super- and subharmonics should be considered.

Finally, whereas during this work simulation data was used for model evaluation, the model performance must be evaluated in a real-life scenario using measured values, once available.

8.2 Conclusion

This work has successfully demonstrated the efficacy of a physics-aware neural network prototype for power quality state estimation. Notably, PANN models have been shown to outperform their conventional fully connected neural network counterparts with a minimum validation MSE loss of 4×10^{-6} compared to 1.1×10^{-5}. Detailed analyses were conducted to examine the impacts of measurement point selection, input preprocessing, architectural considerations, and hyperparameter selection on the model performance, particularly considering the complex nature of power quality data. Moreover, methods for the integration of physical information into neural network models were critically assessed for their applicability in power quality state estimation. The results affirm that neural networks incorporating physical information constitute an effective approach for state estimation in power quality analysis.

References

[1] S. Halpin and A. Card, "Power quality," in Dec. 31, 2007, pp. 1053–1066, ISBN: 978-0-12-088479-7. DOI: 10.1016/B978-012088479-7/50056-0.

[2] E. C. for Standardization, *EN 50160 voltage characteristics of electricity supplied by public electricity networks*, Nov. 2020.

[3] I. E. Commission, *IEC 61000-3-2 electromagnetic compatibility (EMC) - part 3-2: Limits - limits for harmonic current emissions (equipment input current <=16 a per phase)*, version 5.0, Jan. 26, 2018.

[4] I. E. Commission, *IEC 61000-3-12 electromagnetic compatibility (EMC) - part 3-12: Limits - limits for harmonic currents produced by equipment connected to public low-voltage systems with input current >16 a and <= 75 a per phase*, Jun. 4, 2021.

[5] S. Perera and S. Elphick, "Chapter 1 - introduction to power quality in modern power systems," in *Applied Power Quality*, S. Perera and S. Elphick, Eds., Elsevier, Jan. 1, 2023, pp. 1–17, ISBN: 978-0-323-85467-2. DOI: 10.1016/B978-0-323-85467-2.00006-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323854672000068 (visited on 04/04/2023).

[6] L. v. d. Sluis, *Transients in power systems*. Chichester ; New York: Wiley, 2001, 207 pp., ISBN: 978-0-471-48639-8.

[7] P. S. Wright, A. E. Christensen, P. N. Davis, B. Larson, T. Lippert, and P. Patel, "Power quality propagation measurements in smart grids," in *2016 Conference on Precision Electromagnetic Measurements (CPEM 2016)*, ISSN: 2160-0171, Jul. 2016, pp. 1–2. DOI: 10.1109/CPEM.2016.7540458. [Online]. Available: https://ieeexplore.ieee.org/document/7540458 (visited on 10/20/2023).

[8] J. Blanco, R. D. Leal, J. Jacome, J. F. Petit, G. Ordoñez, and V. Barrera, "The influence of transformers, induction motors and fault resistance regarding propagation voltage sags," *Ingeniería e Investigación*, vol. 31, no. 2, pp. 139–147, Jun. 1, 2011, ISSN: 2248-8723, 0120-5609. DOI: 10.15446/ing.investig.v31n2SUP.25225. [Online]. Available: https://revistas.unal.edu.co/index.php/ingeinv/article/view/25225 (visited on 10/20/2023).

[9] Gargoyle888, *English: Delta wye transformer*, Jan. 1, 2009. [Online]. Available: https://commons.wikimedia.org/wiki/File:DeltaWyeTransformer.svg (visited on 10/28/2023).

[10] M. Fotopoulou, S. Petridis, I. Karachalios, and D. Rakopoulos, "A review on distribution system state estimation algorithms," *Applied Sciences*, vol. 12, no. 21, p. 11073, Nov. 1, 2022, ISSN: 2076-3417. DOI: 10.3390/app122111073. [Online]. Available: https://www.mdpi.com/2076-3417/12/21/11073 (visited on 01/03/2023).

[11] C. Fantin, M. Castillo, B. de Carvalho, and J. London, "Using pseudo and virtual measurements in distribution system state estimation," in *2014 IEEE PES Transmission & Distribution Conference and Exposition - Latin America (PES T&D-LA)*, Medellin, Colombia: IEEE, Sep. 2014, pp. 1–6, ISBN: 978-1-4799-6251-8 978-

1-4799-6250-1. DOI: 10.1109/TDC-LA.2014.6955287. [Online]. Available: http://ieeexplore.ieee.org/document/6955287/ (visited on 01/08/2023).

[12] E. Waffenschmidt, M. De Koster, C. Hotz, S. Baum, and I. Stadler, "Decentralized grid control using power grid state estimation," in *27th International Conference on Electricity Distribution (CIRED 2023)*, Rome, Italy: Institution of Engineering and Technology, 2023, pp. 3869–3873, ISBN: 978-1-83953-855-1. DOI: 10.1049/icp.2023.0551. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/icp.2023.0551 (visited on 10/10/2023).

[13] S. M. M. Niyaragh, A. J. Irani, and H. Shayeghi, "Modeling of zero injection buses based to optimal placement of PMUs for full observability of power systems," *Journal of Electrical Engineering & Technology*, vol. 15, no. 6, pp. 2509–2518, Nov. 2020, ISSN: 1975-0102, 2093-7423. DOI: 10.1007/s42835-020-00536-0. [Online]. Available: https://link.springer.com/10.1007/s42835-020-00536-0 (visited on 01/13/2023).

[14] Ali Farzanehrafat, "Power quality state estimation," Ph.D. dissertation, University of Canterbury, Christchurch, New Zealand, Oct. 2014, 108 pp. [Online]. Available: https://core.ac.uk/download/pdf/35471741.pdf (visited on 03/22/2023).

[15] N. R. Watson, "Power quality state estimation," *European Transactions on Electrical Power*, vol. 20, no. 1, pp. 19–33, 2010, ISSN: 1546-3109. DOI: 10.1002/etep.357. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/etep.357 (visited on 09/04/2023).

[16] A. Gudiño-Ochoa, J. Jalomo-Cuevas, J. E. Molinar-Solís, and R. Ochoa-Ornelas, "Analysis of interharmonics generation in induction motors driven by variable frequency drives and AC choppers," *Energies*, vol. 16, no. 14, p. 5538, Jan. 2023, Number: 14 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1996-1073. DOI: 10.3390/en16145538. [Online]. Available: https://www.mdpi.com/1996-1073/16/14/5538 (visited on 10/26/2023).

[17] R. C. Dugan, D. Montenegro, and A. Ballanti, "The open distribution system simulator (OpenDSS)," 2020.

[18] L. Thurner, A. Scheidler, F. Schäfer, *et al.*, "Pandapower - an open source python tool for convenient modeling, analysis and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2018.2829021. arXiv: 1709.06743[cs]. [Online]. Available: http://arxiv.org/abs/1709.06743 (visited on 10/17/2023).

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, Number: 6088 Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0 (visited on 10/26/2023).

[20] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7, pp. 753–768, Aug. 1, 2008, ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2007.11.004.

[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152407001655 (visited on 03/13/2023).

[21] N. Amor, G. Rasool, and N. C. Bouaynaya, *Constrained state estimation – a review*, Mar. 11, 2022. arXiv: 1807.03463[eess]. [Online]. Available: http://arxiv.org/abs/1807.03463 (visited on 01/15/2023).

[22] V. H. Quintana, B. W. Scott, and A. Y. Chikhani, "Constrained power system state estimation," *IFAC Proceedings Volumes*, 10th Triennial IFAC Congress on Automatic Control - 1987 Volume II, Munich, Germany, 27-31 July, vol. 20, no. 5, pp. 7–12, Jul. 1, 1987, ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)55409-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017554099 (visited on 10/10/2023).

[23] G. Korres, "A robust method for equality constrained state estimation," *Power Engineering Review, IEEE*, vol. 21, pp. 69–69, Jan. 1, 2002. DOI: 10.1109/MPER.2001.4311235.

[24] B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022, ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2022.3167722. [Online]. Available: https://ieeexplore.ieee.org/document/9758816/ (visited on 03/09/2023).

[25] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019, ISSN: 00219991. DOI: 10.1016/j.jcp.2018.10.045. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125 (visited on 09/11/2023).

[26] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4796–4805, Nov. 2019, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2019.2909150.

[27] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep neural networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 15, pp. 4069–4077, Aug. 1, 2019, ISSN: 1053-587X, 1941-0476. DOI: 10.1109/TSP.2019.2926023. arXiv: 1811.06146[cs,stat]. [Online]. Available: http://arxiv.org/abs/1811.06146 (visited on 10/11/2023).

[28] L. Zhang, G. Wang, and G. B. Giannakis, "Distribution system state estimation via data-driven and physics-aware deep neural networks," in *2019 IEEE Data Science Workshop (DSW)*, Jun. 2019, pp. 258–262. DOI: 10.1109/DSW.2019.8755581. [Online]. Available: https://ieeexplore.ieee.org/document/8755581 (visited on 10/10/2023).

[29] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4347–4356, Nov. 2020, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2020.2988352.

[30] M.-Q. Tran, A. S. Zamzam, P. H. Nguyen, and G. Pemen, "Multi-area distribution system state estimation using decentralized physics-aware neural networks," *Energies*, vol. 14, no. 11, p. 3025, Jan. 2021, Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1996-1073. DOI: 10.3390/en14113025. [Online]. Available: https://www.mdpi.com/1996-1073/14/11/3025 (visited on 03/13/2023).

[31] M.-Q. Tran, A. S. Zamzam, and P. H. Nguyen, "Enhancement of distribution system state estimation using pruned physics-aware neural networks," in *2021 IEEE Madrid PowerTech*, Jun. 2021, pp. 1–5. DOI: 10.1109/PowerTech46648.2021.9494950.

[32] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems - a review," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 572–588, Jan. 2023, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2022.3162473. [Online]. Available: https://ieeexplore.ieee.org/document/9743327 (visited on 10/10/2023).

[33] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, *Physics-informed neural networks for power systems*, Jan. 29, 2020. DOI: 10.48550/arXiv.1911.03737. arXiv: 1911.03737[cs,eess]. [Online]. Available: http://arxiv.org/abs/1911.03737 (visited on 09/04/2023).

[34] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2082–2092, May 2021, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2020.3029557. [Online]. Available: https://ieeexplore.ieee.org/document/9216092 (visited on 10/10/2023).

[35] M. Arpogaus, J. Montalbano, M. Linke, and G. Schubert, "Probabilistic real-time grid operation management of future distribution grids with high penetration of renewable generators and electrical vehicles based on artificial intelligence," in *CIRED Porto Workshop 2022: E-mobility and power distribution systems*, vol. 2022, Jun. 2022, pp. 147–151. DOI: 10.1049/icp.2022.0681.

[36] V. Bolz, J. Rueß, and A. Zell, "Power flow approximation based on graph convolutional networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 1679–1686. DOI: 10.1109/ICMLA.2019.00274. [Online]. Available: https://ieeexplore.ieee.org/document/8999165 (visited on 10/20/2023).

[37] B. Donon, B. Donnot, I. Guyon, and A. Marot, "Graph neural solver for power systems," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8, ISBN: 978-1-72811-985-4. DOI: 10.1109/IJCNN.2019.8851855. [Online]. Available: https://ieeexplore.ieee.org/document/8851855/ (visited on 10/10/2023).

[38] O. Kundacina, M. Cosovic, D. Miskovic, and D. Vukobratovic, *Distributed nonlinear state estimation in electric power systems using graph neural networks*, Sep. 8, 2022. DOI: 10.48550/arXiv.2207.11465. arXiv: 2207.11465[cs,eess]. [Online]. Available: http://arxiv.org/abs/2207.11465 (visited on 04/06/2023).

[39] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Jun. 12, 2022, pp. 1–6. DOI: 10.1109/PMAPS53380.2022.9810559. arXiv: 2201.04056[cs,eess,math]. [Online]. Available: http://arxiv.org/abs/2201.04056 (visited on 04/06/2023).

[40] O. Kundacina, G. Gojic, M. Cosovic, D. Miskovic, and D. Vukobratovic, *Scalability and sample efficiency analysis of graph neural networks for power system state estimation*, Mar. 2, 2023. DOI: 10.48550/arXiv.2303.00105. arXiv: 2303.00105[cs]. [Online]. Available: http://arxiv.org/abs/2303.00105 (visited on 04/06/2023).

[41] B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022, Conference Name: IEEE Transactions on Instrumentation and Measurement, ISSN: 1557-9662. DOI: 10.1109/TIM.2022.3167722.

[42] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 541–551, Apr. 2012, Conference Name: IEEE Transactions on Neural Networks and Learning Systems, ISSN: 2162-2388. DOI: 10.1109/TNNLS.2012.2183613. [Online]. Available: https://ieeexplore.ieee.org/document/6138313 (visited on 10/13/2023).

[43] J. Bassey, L. Qian, and X. Li, *A survey of complex-valued neural networks*, Jan. 28, 2021. arXiv: 2101.12249[cs,stat]. [Online]. Available: http://arxiv.org/abs/2101.12249 (visited on 04/27/2023).

[44] T. Wu, A. Scaglione, and D. Arnold, *Complex-value spatio-temporal graph convolutional neural networks and its applications to electric power systems AI*, Sep. 22, 2022. arXiv: 2208.08485[cs,eess]. [Online]. Available: http://arxiv.org/abs/2208.08485 (visited on 04/27/2023).

[45] K. Strunz, E. Abbasi, R. Fletcher, N. Hatziargyriou, R. Iravani, and G. Joos, *TF C6.04.02 : TB 575 – Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources*. Apr. 1, 2014, ISBN: 978-2-85873-270-8.

[46] N. Pflugradt, P. Stenzel, L. Kotzur, and D. Stolten, "LoadProfileGenerator: An agent-based behavior simulation for generating residential load profiles," *Journal of Open Source Software*, vol. 7, no. 71, p. 3574, Mar. 25, 2022, ISSN: 2475-9066. DOI: 10.21105/joss.03574. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.03574 (visited on 10/21/2023).

[47] T. Tjaden, J. Bergner, J. Weniger, and V. Quaschning, *Repräsentative elektrische Lastprofile für Einfamilien- häuser in Deutschland auf 1-sekündiger Datenbasis*, Hochschule für Technik und Wirtschaft (HTW) Berlin. (visited on 10/21/2023).

[48] E. Waffenschmidt. "Gesammelte messdaten der SMA-beispiel-PV-anlage "SMA flexible storage system"." (), [Online]. Available: https://www.100pro-erneuerbare.com/messdaten/messdaten.htm (visited on 10/21/2023).

[49] R. F. Arritt and R. C. Dugan, "Validation of harmonic models for PV inverters: PVMOD milestone 2.8.2," Electric Power Research Institute (EPRI), Washington, D.C. (United States), 2.8.2, Sep. 26, 2022. DOI: 10.2172/1894588. [Online]. Available: https://www.osti.gov/biblio/1894588 (visited on 10/21/2023).

[50] I. N. Laboratory. "SteadyStateLoadCharacterization2014bmwi3.pdf." (), [Online]. Available: https://avt.inl.gov/sites/default/files/pdf/fsev/SteadyStateLoadCharacterization2014BMWi3.pdf (visited on 10/21/2023).

[51] I. N. Laboratory. "SteadyStateLoadCharacterization2015leaf.pdf." (), [Online]. Available: https://avt.inl.gov/sites/default/files/pdf/fsev/SteadyStateLoadCharacterization2015Leaf.pdf (visited on 10/21/2023).

[52] I. N. Laboratory. "SteadyStateVehicleChargingFactSheet_2015mercedesbclass.pdf." (), [Online]. Available: https://avt.inl.gov/sites/default/files/pdf/fsev/SteadyStateVehicleChargingFactSheet_2015MercedesBClass.pdf (visited on 10/21/2023).

[53] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, *Visualizing the loss landscape of neural nets*, Nov. 7, 2018. DOI: 10.48550/arXiv.1712.09913. arXiv: 1712.09913[cs,stat]. [Online]. Available: http://arxiv.org/abs/1712.09913 (visited on 09/14/2023).

[54] K. Bogner, F. Pappenberger, and H. L. Cloke, "Technical note: The normal quantile transformation and its application in a flood forecasting system," *Hydrology and Earth System Sciences*, vol. 16, no. 4, pp. 1085–1094, Apr. 2, 2012, ISSN: 1607-7938. DOI: 10.5194/hess-16-1085-2012. [Online]. Available: https://hess.copernicus.org/articles/16/1085/2012/ (visited on 10/29/2023).

[55] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv.org. (Feb. 11, 2015), [Online]. Available: https://arxiv.org/abs/1502.03167v3 (visited on 10/09/2023).

[56] M. D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks*, Nov. 28, 2013. DOI: 10.48550/arXiv.1311.2901. arXiv: 1311.2901[cs]. [Online]. Available: http://arxiv.org/abs/1311.2901 (visited on 09/21/2023).

[57] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, Dec. 10, 2015. DOI: 10.48550/arXiv.1512.03385. arXiv: 1512.03385[cs]. [Online]. Available: http://arxiv.org/abs/1512.03385 (visited on 10/13/2023).

[58] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, USA: IEEE, Jul. 2016, pp. 1–5, ISBN: 978-1-5090-4168-8. DOI: 10.1109/PESGM.2016.7741972. [Online]. Available: http://ieeexplore.ieee.org/document/7741972/ (visited on 10/16/2023).

[59] D. Salls, J. R. Torres, A. C. Varghese, J. Patterson, and A. Pal, *Statistical characterization of random errors present in synchrophasor measurements*, Mar. 16, 2021. DOI: 10.48550/arXiv.2103.09285. arXiv: 2103.09285[cs,eess]. [Online]. Available: http://arxiv.org/abs/2103.09285 (visited on 10/16/2023).

[60] S. Wang, J. Zhao, Z. Huang, and R. Diao, "Assessing gaussian assumption of PMU measurement error using field data," *IEEE Transactions on Power Delivery*, vol. 33, no. 6, pp. 3233–3236, Dec. 2018, Conference Name: IEEE Transactions on Power Delivery, ISSN: 1937-4208. DOI: 10.1109/TPWRD.2017.2762927. [Online]. Available: https://ieeexplore.ieee.org/document/8067439 (visited on 10/16/2023).

[61] P. Castello, C. Muscas, and P. A. Pegoraro, "Statistical behavior of PMU measurement errors: An experimental characterization," *IEEE Open Journal of Instrumentation and Measurement*, vol. 1, pp. 1–9, 2022, Conference Name: IEEE Open Journal of Instrumentation and Measurement, ISSN: 2768-7236. DOI: 10.1109/OJIM.2022.3205678. [Online]. Available: https://ieeexplore.ieee.org/document/9887624 (visited on 10/16/2023).

Declaration of Originality

I certify that I have independently written the thesis I have submitted. All passages taken verbatim or in essence from published or unpublished works of others or of the author themselves have been identified as such. All sources and aids used for the work are cited. The work has not been submitted in the same form or in essential parts to any other examining authority.

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum Unterschrift

Appendix

Code Snippet 1 Preparation of admittance matrices for PANN

```
import numpy as np
def prepare_admittance_matrices(admittance_matrices, known_bus_indices):
    """
    Modify the given admittance matrices based on known bus indices. Extend dimensions and compute the absolute value of elements.
    :param admittance_matrices: Array of admittance matrices per frequency and complex component.
    :param known_bus_indices: Indices of buses with known elements (e.g., measurement buses).
                None if inputs for all nodes are available.
    :return Array of modified admittance matrices.
    """
    # Extract elements corresponding to known bus indices from the original admittance matrices.
    if known_bus_indices is not None:
        pruned_admittance_matrices = admittance_matrices[:, :, known_bus_indices, :]
    else:
        pruned_admittance_matrices = admittance_matrices

    # Compute the absolute value of the pruned admittance matrices.
    abs_admittance_matrices = np.abs(pruned_admittance_matrices)

    # Extend the dimensions of the absolute value matrices to fit kernel shape
    extended_dimensions = abs_admittance_matrices[:, :, :, :, np.newaxis, np.newaxis]

    # Repeat the extended matrix along the newly added dimensions.
    tiled_matrices = np.tile(extended_dimensions,
                             (1, 1, 1, 1,
                             abs_admittance_matrices.shape[0],
                             abs_admittance_matrices.shape[1]))

    # Reorder the dimensions to obtain the final result.
    reordered_matrices = np.transpose(tiled_matrices, (0, 1, 2, 4, 5, 3))

    return reordered_matrices
```

Harmonic	Actual	Predicted
1	11054.8	11055.8820125668
2	0.413326	0.410686680708806
3	28.45515	29.2180298464032
4	0.498088	0.500611968886191
5	38.3808	39.6027188250123
6	0.131465	0.131788541629096
7	29.0194	29.7383663731357
8	1.054265	1.06041234297184
9	21.34695	21.919773254299
10	0.894305	0.899097252778349
11	46.34915	47.6234288221109
12	0.0689875	0.0682768737938953
13	52.2995	53.8263010557241
14	0.3159585	0.316605248770766
15	32.5914	33.5480388193729
16	0.2891325	0.29052842086897
17	29.05805	29.8672324524794
18	0.0186696	0.0186430476394765
19	13.9893	14.4323291840178
20	0.113021	0.112392585744883

Table 11 Actual and predicted values for node 37 of a randomly selected single iteration of the test set, predicted using a PANN model.