A Novel Deep Learning Power Quality Disturbance Classification Method using Autoencoders

Callum O'Donovan, Cinzia Giannetti and Grazia Todeschini
College of Engineering, Swansea University, Fabian Way, Swansea, Wales, U.K.

Keywords: Classification, Feature Extraction, Power Quality Disturbance, Deep Learning, Convolutional Neural Network, LSTM, Recurrent Neural Network, Autoencoder.

Abstract: Automatic identification and classification of power quality disturbances (PQDs) is crucial for maintaining efficiency and safety of electrical systems and equipment condition. In recent years emerging deep learning techniques have shown potential in performing classification of PQDs. This paper proposes two novel deep learning models, called CNN(AE)-LSTM and CNN-LSTM(AE) that automatically distinguish between normal power system behaviour and three types of PQDs: voltage sags, voltage swells and interruptions. The CNN-LSTM(AE) model achieved the highest average classification accuracy with a 65:35 train-test split. The Adam optimiser and a learning rate of 0.001 were used for ten epochs with a batch size of 64. Both models are trained using real world data and outperform models found in literature. This work demonstrates the potential of deep learning in classifying PQDs and hence paves the way to effective implementation of AI-based automated quality monitoring to identify disturbances and reduce failures in real world power systems.

1 INTRODUCTION

In ideal power systems, voltage and current waveforms are sinusoids at fundamental frequency (i.e., 50 Hz or 60 Hz for Europe or the USA mains respectively) (Baggini, 2008). While amplitude of the voltage waveform is strictly regulated and maintained close to the rated value, the current waveform is more variable, as it depends on the rating of loads connected to the system and their power demand. Any deviation from the 'ideal' waveform is defined as a power quality disturbance (PQD) (Bollen, 2003). Numerous PQDs exist in practice, and power quality standards have been developed to provide a classification of each disturbance and to provide acceptable limits (IEEE Standards Association, 2019).

Because voltage waveforms are generally more stable and less subject to fluctuations of electricity demand, this research focused on the classification of voltage signals, and on the identification of three PQDs: voltage sags/dips, voltage swells and interruptions. Voltage sag is a reduction in voltage amplitude between 5-90% of the nominal (rated) voltage, voltage swell is an increase of the voltage amplitude above 105% of the nominal voltage, and an interruption is a reduction of the voltage amplitude below 10% of the nominal voltage.

Small deviations from the rated voltage value are acceptable and do not harm the electricity system or the equipment. With increasing levels of PQDs, some detrimental effects can be observed. For example, excessive fluctuations of the voltage waveform for extended periods of time may lead to damage of equipment connected to the power grid, such as motor failures (Wang & Chen, 2019).

In recent years, with the increase of power-electronics based devices connected to the power grid (such as renewable energy sources and electric vehicles), PQDs have become more common, thus leading to concerns for utilities and power system operators in terms of guaranteeing the quality of the electrical energy supplied to their customers. As a result, increasing numbers of power quality monitors are currently being installed, thus allowing the collection of large amounts of voltage and current data (Demirci et al., 2011). Analysis of these waveforms and identification of PQDs allows implementing mitigating solutions, thus improving system operating conditions and extending the life-time of the equipment (Wang & Chen, 2019). Various PQD classification methods exist, as described in (Demirci et al., 2011). Historically, PQDs have been classified using visual inspection of the voltage and current waveforms (Wang & Chen,

O'Donovan, C., Giannetti, C. and Todeschini, G.
A Novel Deep Learning Power Quality Disturbance Classification Method using Autoencoders.
DOI: 10.5220/0010347103730380
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 373-380
ISBN: 978-989-758-484-8

2019). Later on, techniques have been developed to automatically detect and classify PQDs, based on signal processing techniques (Bravo-Rodriquez, Torres and Borrás, 2020). In recent years, some methods have been proposed to provide automatic classification of PQDs using Big Data with machine learning (Wang & Chen, 2019).

Machine learning is a broad term that refers to algorithms that can learn from large amount of data. In recent years, machine learning has gained much popularity due to development of more accurate algorithms, increased training data availability and increased computational resources worldwide (Jordan & Mitchell, 2015). Machine learning models can be used for a vast range of tasks such as credit-card fraud detection, speech recognition and medical diagnosis (Jordan & Mitchell, 2015). Deep Learning refers to the particular type of Machine Learning techniques used for learning high-level features from data in a hierarchical manner using stacked, layer-wise architectures (Goodfellow & Bengio, 2015). Among these are convolutional neural networks (CNNs), long short-term memory networks (LSTMs), convolutional autoencoders (CAEs), and LSTM autoencoders. Deep Learning models demonstrate excellent predictive capabilities in image and speech recognition, natural language processing (NLP), and intelligent gamification (Goodfellow & Bengio, 2015). We demonstrate the application of Deep Learning to automatically detect PQ events using real world datasets. The proposed deep learning models are capable of accurately classifying four different types of PQ events and outperform other models proposed in the literature.

Machine learning models explored in this paper are a combination of several techniques including convolutional neural networks (CNNs), long short-term memory networks (LSTMs), convolutional autoencoders (CAEs), and LSTM autoencoders.

The paper is organised as follows. Section 1 includes background information on machine learning techniques used for PQD classification. Section 2 describes the methodology; results are presented in section 3 and the paper is concluded in Section 4.

2 BACKGROUND

In this section a review of deep learning techniques in the context of PDQs is provided.

2.1 Convolutional Neural Networks (CNNs)

CNNs are a type of artificial neural network (ANN) used for feature extraction, that primarily take images as input but can also handle other data such as words and temporal signals (O'Shea & Nash, 2015), (Kalchbrenner, Grefenstette & Blunsom, 2014), (Palaz, Collobert & Magimai-Doss). In (Bagheri, Gu & Bollen, 2018), deep CNNs were utilised to perform automatic feature extraction and classify different types of voltage dips (sags) recorded by power quality monitors (specifically, the PQube meters). Pre-processed data was used rather than raw data. The model was trained and tested as case studies (C1, C2 and C3) which handled three voltage datasets in a different way. The three data set were from Sweden (D1), the world (D2) and the UK (D3) (Bagheri et al., 2018). This method is summarised in Table 1.

Table 1: Summary of the different ways data was used in (Bagheri, Gu & Bollen, 2018).

Case Study	Training Set	Testing Set
C1	0.75 (D1+D2+D3)	0.25 (D1+D2+D3)
C2	D2+D3	D1
C3	D1+D2	D3

Model performance was represented as loss, accuracy, classification rate and false alarm rate, but only accuracy will be discussed here to align with this project (Bagheri et al., 2018). For C1, C2 and C3, accuracy was 97.72%, 95.18% and 93.59%, respectively (Bagheri et al., 2018). Results suggest CNNs are effective for PQD classification, works in the literature also use data from http://map.pqube.com/. Architecture proposed in (Bagheri et al., 2018) is summarised in Table 2. Both batch size and epochs were set to 250, and the Adam optimiser was applied (Bagheri et al., 2018).

Table 2: Summary of architecture proposed in (Bagheri, Gu & Bollen, 2018).

Layers	Filter Size / No. of cells
2D Conv1+ReLU	(5, 5) x 16
2D Conv1+ReLU+Max-Pooling	(3, 3) x 32
2D Conv1+ReLU	(3, 3) x 64
2D Conv1+ReLU+Max-Pooling	(3, 3) x 128
FC1+ReLU	1024
FC2+ReLU	128
FC3+Softmax	7

Also supporting use of CNNs to classify PQDs is (Balouji & Salor, 2017), as it applies CNNs with real event images from four transmission substations and achieves 100% accuracy. The architecture proposed in (Balouji & Salor, 2017, pp219) is similar to (Bagheri et al., 2018, pp4); the main difference between the two papers is that (Bagheri et al., 2018) worked with pre-processed data whilst (Balouji & Salor, 2017) used images of voltage waveforms. This project will apply raw data, but these two alternatives should be considered in future. The study in (Balouji & Salor, 2017) also found that using 65 to 135 epochs was most suitable.

2.2 Long Short-term Memory Networks (LSTMs)

LSTMs are a type of recurrent neural network (RNN) that deal with data in a sequential format (Baccouche, Mamalet, Wolf, Garcia & Baskurt, 2011). This, combined with their gating system which gives them a 'memory' (as it will be explained later), means that LSTMs are able to put data into context (Baccouche et al., 2011).

LSTMs were utilised for automatic feature extraction and classification of three-phase voltage dips collected from various countries (Balouji, Gu, Bollen, Bagheri & Nazari, 2018). Model performance was evaluated using classification rate and false alarm metrics on the test set. Seven different classes were defined, and the median average classification and false alarm rates of dip types were 93.4% and 7.78% respectively (Balouji et al., 2018). Additionally, four LSTM layers were implemented with a hyperbolic tangent activation for feature extraction, so that different layers could extract the multiscale features (Balouji et al., 2018). Each LSTM layer is accompanied by a batch normalisation layer, and a fully-connected layer (also known as a dense layer) with softmax activation for classification is used as the final layer (Balouji et al., 2018). Before feature extraction and classification, the method pre-processed the voltage sequence data into root mean square (RMS) sequence data and divided it into segments for computational efficiency (Balouji et al., 2018).

In (Katić & Stanisavljević, 2018) a LSTM-based network was proposed to automatically detect and classify voltage dips in real, simulated and laboratory-produced data. High model performance was shown through overall classification accuracy exceeding 97% (Katić & Stanisavljević, 2018). Together, (Balouji et al., 2018) and (Katić & Stanisavljević, 2018) suggest application of LSTM layers to classify PQDs could be effective.

2.3 Convolutional Autoencoders (CAEs)

CAEs take CNNs' ability to extract spatially local features, but also employ autoencoders' (AEs) ability to learn features from unlabelled data (unsupervised learning), which allows distinction of more subtle features than a CNN could identify alone (Seyfioğlu, Özbayoğlu & Gürbüz, 2018).

A comparison of performances of a convolutional autoencoder to a multiclass support vector machine (SVM), an autoencoder and a CNN, when classifying different types of human activity based on radar measurements, is shown in (Seyfioğlu et al., 2018). Results showed that accuracies of a multiclass SVM, an autoencoder, a CNN and a CAE were 76.9%, 84.1%, 90.1% and 94.2% respectively. This suggests that using a deep CAE (DCAE) during model development could result in a higher performance model than using a traditional CNN or AE.

This approach is supported by another research work that compared performance of a DCAE to SVM, sparse representation classifier (SRC) and stacked autoencoder (SAE) models when classifying high-resolution synthetic aperture radar (SAR) images (Geng, Fan, Wang, Ma & Chen, 2015). Results showed overall accuracy of the SVM, SRC, SAE and DCAE models were approximately 76.92%, 81.08%, 82.45% and 88.11% respectively (Geng et al., 2015). Results presented in this work further support application of DCAE models for classification because DCAE accuracy significantly exceeds accuracy of other models. Furthermore, results showed that the DCAE was most accurate at classifying four out of five individual classes (Geng et al., 2015).

Even though both (Seyfioğlu et al., 2018) and (Geng et al., 2015) propose 'deep convolutional autoencoders', their interpretation of this nomenclature is different. In (Seyfioğlu et al., 2018), three convolutional layers are used on each side, with max-pooling and unpooling layers located between them. The number of filters applied decreases across each layer for the first three convolutional layers (encoding) and increases across each layer for the last three convolutional layers (decoding) (Seyfioğlu et al., 2018). However, (Geng et al., 2015) proposes a convolutional layer in the same network as a traditional sparse autoencoder (encoder-decoder made from fully-connected layers rather than convolutional layers).

As results from (Seyfioğlu et al., 2018) were encouraging, application of six convolutional layers for a CAE will be tested.

2.4 LSTM Autoencoders

LSTM autoencoders are also known as sequence-to-sequence autoencoders and have been shown to be successful in tasks such as machine translation, natural language generation and reconstruction and image captioning (Mehdiyev, Lahann, Emrich, Enke, Fettke & Loos, 2017).

A GRU-based autoencoder presented in (Amiriparian, Freitag, Cummins & Schuller, 2017) successfully classified labelled acoustic scene audio data with accuracy of 88%. LSTM units were adopted instead of GRUs during model design, but did not show a performance improvement, suggesting value in experimenting with GRU and LSTM autoencoders.

Specific parameter values for the architecture proposed in (Amiriparian et al., 2017) are not given, but the general idea is that RNN layers define the encoder, followed by a fully-connected layer with a hyperbolic tangent activation function. The final layers consist of RNN layers for the decoder, followed by a linear projection layer with a hyperbolic tangent activation function are also applied to the RNN layers' inputs and outputs (Amiriparian et al., 2017). This type of architecture will be tested when developing models, as the results from (Cho et al., 2014), (Bengio et al., 2015), (Amiriparian et al. 2017) and (Patilkulkarni & Lakshmi, 2013) have shown to improve performance.

2.5 CNN-LSTM

CNN-LSTM networks are neural networks that combine elements of CNNs (mainly CNN and pooling layers) with elements of LSTM networks (mainly LSTM and flatten layers). CNN-LSTM networks are used in classification problems as they provide advantages of both CNNs and LSTMs, namely the spatial feature extraction ability of CNNs and the temporal sequential learning ability of LSTMs (Mohan, Soman & Vinayakumar, 2017).

A comparison of the performance of several models that were CNN, RNN, identity recurrent neural network (I-RNN), LSTM, GRU and CNN-LSTM based, when classifying synthetic and real-time PQDs, can be found in (Mohan et al., 2017). The synthetic data contained eleven different classes which were both single and combined disturbance types, whereas real-time data contained only three classes (Mohan et al., 2017). Results showed that for synthetic data, the CNN-LSTM model have the highest overall accuracy of 98.4% (Mohan et al., 2017). Only the CNN-LSTM model was tested on real-time data and it achieved an accuracy of 91.9% (Mohan et al., 2017). These results suggest a CNN-LSTM model can perform accurate classification of synthetic and real-time PQDs. A batch size of 32 and 1000 epochs were proposed (Mohan et al., 2017).

The performance of a CNN-LSTM model when classifying electrocardiogram (ECG) signals into five different classes for automatic arrythmia diagnosis was studied in (Oh, Ng, Tan & Acharya, 2018). Results showed that the hybrid model performed with 98.1% accuracy (Oh et al., 2018). This result supports the claim that adoption of a CNN-LSTM model can result in accurate classification performance. Although ECG signals differ to PQ signals, ECG signals are still voltage measurements but taken in the heart, and both data types are periodical. Architecture of the CNN-LSTM proposed in (Oh et al., 2018) is described with good detail. A batch size of ten was chosen and the model was trained for 150 epochs.

In (Garcia et al., 2020), a CNN-LSTM model was used to classify five different PQDs using voltage waveforms as training and testing data and achieved a maximum accuracy of 84.76%.

Based on the literature review, the following networks have been identified as successful for the identification of PQDs: CNN autoencoder with LSTM and CNN with LSTM autoencoder.

Therefore, the networks above were adopted for testing with PQD signals. In the following sections, tests were carried out using CNN and LSTM. Additional networks are at the moment under development and will be presented in future work.

3 METHODOLOGY

The proposed approach is comprised of two steps. Step 1 is the collection and pre-processing of data. Step 2 refers to the development of two models using Design of Experiments (DOE) and suggestions from the literature.

3.1 Step 1: Data Collection & Pre-processing

This work uses real data recorded by PQube power quality monitors. The data can be accessed online and is openly available (Power Standards Lab, 2019). Each sample contains three-phase voltage data (L1-N, L2-N and L3-N) for varying numbers of time-steps, accompanied by the label for the type of PQD present. The source website contains PQD data from numerous PQube meters located around the world.

Different meters had different software versions and were recording data for different types of electrical systems, which meant each meter had a different range of disturbances. In addition, some meters recorded voltages in L-N format, some in L-L format, and some only recorded voltages in one or two phases rather than three. The dataset involved with this work was retrieved from the PQube map website (Power Standards Lab, 2019) and is summarised in Table 3.

Table 3: Number of samples for each class.

Class	Number of Samples
Snapshot (Normal)	976
Voltage Sag/Dip	315
Voltage Swell	275
Interruption	123
Total	1689

After data was imported, it was split into training and testing data using the Python 'train_test_split' function. A 65:35 train-test split was adopted as ratios of 80:20, 75:25 and 70:30 were experimented with and gave poorer results. This was likely due to 65% of the data being enough for the model to learn about it and predict classes, whereas any higher percentage resulted in the model learning the training data too well and therefore performing poorly on test data (overfitting).

As shown in Table 3, there is a significant imbalance between the number of samples of each class, with the snapshot class being a large majority class: of the 1689 samples, 976 (approx. 57.8%) are snapshots. If the data was trained and tested on with this imbalance, it could potentially reduce model performance, because any model could learn that it can achieve this as an accuracy just by classing every sample as a snapshot, which is undesirable (Towards Data Science, 2019). Therefore, to solve this problem, oversampling was performed to balance the number of samples of each class. Several oversampling options existed, and multiple methods were attempted. The RandomOverSampler function was chosen and works by choosing random samples of the minority class or classes and duplicating them until classes are balanced (Towards Data Science, 2019).

3.2 Step 2: Model Training & DOE Optimisation

Initially, elements from the models proposed in (Mohan et al., 2017), (Seyfioğlu et al., 2018) and (Balouji & Salor, 2017) were combined to produce a convolutional autoencoder with LSTM model named CNN(AE)-LSTM. This achieved an average accuracy of 93.8% over ten runs. As suggested by (Mehdiyev et al., 2017), (Cho et al., 2014) and (Bengio et al., 2015), the next model developed replaced the convolutional autoencoder element of the CNN(AE)-LSTM model with a normal CNN element, and replaced the LSTM element with a LSTM autoencoder. This model was named CNN-LSTM(AE). This achieved an average accuracy of 96.6% over ten runs.

The second aim of Step 2 was to find the optimal model parameters which was achieved using orthogonal arrays. Five factors were chosen for optimisation, namely: number of convolutional filters, convolutional and max-pooling strides, dropout rate, number of LSTM memory blocks and max-pooling filter size.

Each factor had three different levels, taken mostly from the literature. The exceptions were a CNN filter combination of 16, 8, 4, 8, 16, stride of two, a dropout rate of 0.7 and a max-pooling filter size of four. These were chosen experimentally for convenience and are not informed from the literature. All settings are summarised in Table 4 and Table 5.

Note that the convolutional filter sizes were not changed as previous work (Balouji & Salor, 2017), (Mohan et al., 2017) and (Seyfioğlu et al., 2018) agreed three was the best. Orthogonal arrays applied were $L_{27}(3^5)$. For the two optimised models, time per epoch was compared.

Table 4: Parameters and levels chosen for the orthogonal array for the CNN(AE)-LSTM model.

Parameter	Setting 1	Setting 2	Setting 3
No. of Conv Filters	8, 4, 2, 4, 8	16, 8, 4, 8, 16	32, 16, 8, 16, 32
Conv & Pooling Strides	1	2	3
Dropout Rate	0.3	0.5	0.7
LSTM Memory Blocks	20	50	128
Max-Pooling Filter Size	2	3	4

Table 5: Parameters and levels chosen for the orthogonal array for the CNN-LSTM(AE) model.

Parameter	Setting 1	Setting 2	Setting 3
No. of Conv Filters	8	16	32
Conv & Pooling Strides	1	2	3
Dropout Rate	0.3	0.5	0.7
LSTM Memory Blocks	27,15, 8, 15, 27	62, 32, 8, 32,62	128, 64, 32, 64, 128
Max-Pooling Filter Size	2	3	4

4 RESULTS & DISCUSSION

All models were run ten times and overall accuracies are reported in Table 6 which shows slightly better accuracies overall for the CNN-LSTM(AE) model than the CNN(AE)-LSTM model.

Also, no significant drops in accuracy exist, likely because feature extraction of the LSTM autoencoder has been more effective than the plain LSTM.

Table 6: Testing accuracies achieved for every run of every model that initially had high performance during Step 2.

		Accuracy (%)	
	Exp. No.	CNN(AE)-LSTM	CNN-LSTM(AE)
Test No.	1	95.6	96.7
	2	93.5	96.1
	3	96.5	98.2
	4	95.8	96.8
	5	97.2	97.1
	6	83.2	97.4
	7	96.0	94.8
	8	93.7	96.9
	9	92.3	95.2
	10	94.8	96.9
	Avg.	93.9	96.6
	Std. Dev.	4.0	1.0

The CNN-LSTM(AE) model in Table 7 shows that using less convolutional and pooling strides resulted in better performance. At first glance, this is not supported by the CNN(AE)-LSTM model. However, the CAE element of the CNN(AE)-LSTM model uses four convolutional layers and two max-pooling layers, meaning that using a lower stride was much more computationally demanding causing memory failures at one stride. Using two strides worked but this did not appear as the optimal stride number because three strides was tested with other parameters at better settings.

No trends were found regarding the number of LSTM memory blocks used, which appeared to have less impact on testing accuracy than CNN-based parameters. Setting max-pooling and up-sampling filter sizes to two was shown to consistently be the best option for this application. This conclusion is aligned with the suggestions in the literature (Mohan et al., 2017), (Seyfioğlu et al., 2018), (Liu et al., 2017). In addition, lower max-pooling filter sizes mean less smoothing, so more data is preserved. Up-sampling size is the number of times each sample is magnified (Medium, 2018).

The LSTM autoencoder model surpassed the model using a normal LSTM layer. LSTM autoencoders were more effective than normal LSTM elements as LSTM autoencoders learn about data more thoroughly.

Table 7: Summary of the settings leading to the best experimental results for each model, with average accuracies achieved and standard deviation values.

Model	CNN(AE)-LSTM	CNN-LSTM(AE)
Trial 1	8.003	35.014
Trial 2	5.002	32.013
Trial 3	5.002	32.012
Average	6.002	33.013

Table 8 shows time taken for each model to run one epoch. Three epoch trials were run for each model and then an average was taken to minimise error, as there was some variation between each execution. Results show employing LSTM autoencoders required four to five times the time per epoch as the non-LSTM autoencoder model. Also, the CNN(AE)-LSTM was fast but less accurate than the CNN-LSTM(AE) model which was quite slow and gave moderately good accuracy.

Table 8: Average times achieved by each model's best experimental set-up in seconds.

Model	CNN(AE)-LSTM	CNN-LSTM(AE)
No. of Conv Filters	16, 8, 4, 8, 16	8
Conv & Pooling Strides	3	1
Dropout Rate	0.3	0.3
LSTM Memory Blocks	50	27, 15, 8, 15, 27
Max-Pooling & Up-Sampling Filter Size	2	2
Average Accuracy	0.952	0.979
Standard Deviation	0.012	0.006

5 CONCLUSIONS

In this paper two deep learning models for predicting PQDs have been proposed and tested, namely CNN(AE)-LSTM and CNN-LSTM(AE) These models achieved accuracies of 95.153%±0.012 and 97.894%±0.006%, respectively.

The CNN-LSTM(AE) achieved great accuracy but it was relatively slow, whilst the CNN(AE)-LSTM achieved poorer accuracy but was much quicker per epoch.

For the model optimisation step it was found that one stride was more accurate but more computationally demanding, affecting memory usage the most. Larger filter sizes and strides caused lower accuracy due to lower resolution of data captured by filters, whilst more convolutional filters resulted in higher accuracies.

Generally, a dropout rate of 0.3 was the best.

CNN layers appeared to be more computationally demanding and more effective than LSTM layers, possibly because CNN layers are generally used with images and filters pixel values in a matrix (similar to the PQube data), unlike LSTM layers that are generally used with sequences. Accuracy shared no relationship with LSTM memory blocks or decomposition level.

The CNN-LSTM(AE) exceeded performance of models in the literature (Bagheri et al., 2018), (Balouji et al., 2018), (Garcia et al., 2020), (Uyar et al., 2008), (Abdel-Galil et al., 2004), of which some worked with synthetic data and others worked with real data, whilst the CNN(AE)-LSTM exceeded some of these (Balouji et al., 2018), (Garcia et al., 2020), (Uyar et al., 2008), (Abdel-Galil et al., 2004) The next steps of this research will consist of further development of the proposed model and in testing its accuracy in detecting other PQDs.

6 COPYRIGHT FORM

ACKNOWLEDGEMENTS

The authors would like to acknowledge the M2A funding from the European Social Fund via the Welsh Government (c80816) and the Engineering and Physical Sciences Research Council (G. Todeschini: Project EP/T013206/1; Dr Giannetti: Project EP/S001387/1).

REFERENCES

Abdel-Galil, T.K., Kamel, M., Youssef, A.M. & El-Saadany, E.F. & Salama, M.M.A. (2004). Power Quality Disturbance Classification Using the Inductive Inference Approach. IEEE.

Amiriparian, S., Freitag, M., Cummins, N. & Schuller, B. (2017). Sequence to Sequence Autoencoders for Unsupervised Representation Learning from Audio. ResearchGate.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. & Baskurt, A. (2011). Sequential Deep Learning for Human Action Recognition. Springer.

Baggini, A. (2008). Handbook of Power Quality. Wiley.

Bagheri, A., Gu, I.Y.H. & Bollen, M.H.J. (2018). A Robust Transform-Domain Deep Convolutional Network for Voltage Dip Classification. IEEE.

Balouji, E. & Salor, O. (2017). Classification of Power Quality Events Using Deep Learning on Event Images. IEEE.

Balouji, E., Gu, I.Y.H., Bollen, M.H.J., Bagheri, A., Nazari, M. (2018). A LSTM-based Deep Learning Method with Application to Voltage Dip Classification. IEEE.

Bengio, S., Vinyals, O., Jaitly, N. & Shazeer, N. (2015). Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. arXiv.

Bollen, M.H.J. (2003). What is power quality? Elsevier.

Bravo-Rodriquez, J.C., Torres, F.J. & Borrás, M.D. (2020). Hybrid Machine Learning Models for Classifying Power Quality Disturbances: A Comparative Study. MDPI.

Cho, K., Merrienboer, B.V., Gulcehre, C. & Bougares, F. (2014). Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation. arXiv.

D2L. (Date Unknown) Padding and Stride.

Demirci, T., Kalaycioglu, A., Kucuk, D., Salor, O., Guder, M., Pakhuylu, S. et al. (2011). Nationwide real-time monitoring system for electrical quantities and power quality of the electricity transmission system. IEEE.

Garcia, C.I., Grasso, F., Luchetta, A., Piccirilli, M.C., Paolucci, L. & Talluri, G. (2020). A Comparison of Power Quality Disturbance Detection and Classification Methods Using CNN, LSTM and CNN-LSTM. MDPI.

Geng, J., Fan, J., Wang, H., Ma, X., Li, B. & Chen, F. (2015). High-Resolution SAR Image Classification via Deep Convolutional Autoencoders. IEEE.

Goodfellow, I. & Bengio, Y. (2015). Deep learning. MIT Press.

IEEE Standards Association. (2019). IEEE 1159-2019 - IEEE Recommended Practice for Monitoring Electric Power Quality. IEEE.

Jordan, M.I. & Mitchell, T.M. (2015). Machine learning: Trends, perspectives, and prospects. ScienceMag.

Kalchbrenner, N., Grefenstette, E. & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. arXiv.

Katić, V.A. & Stanisavljević, A.M. (2018). Smart Detection of Voltage Dips Using Voltage Harmonics Footprint. IEEE.

Liu, Q., Zhou, F., Hang, R. & Yuan, X. Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyperspectral Image Classification. (2017). arXiv.

Medium. (2018). Basic Overview of Convolutional Neural Network (CNN).

Mehdiyev, N., Lahann, J., Emrich, A., Enke, D., Fettke, P. & Loos, P. (2017). Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry. ScienceDirect.

Mohan, N., Soman, K.P. & Vinayakumar, R. (2017). Deep Power: Deep Learning Architectures for Power Quality Disturbances Classification. IEEE.

O'Shea, K. & Nash, R. (2015). An Introduction to Convolutional Neural Networks. arXiv.

Oh, SL., Ng, E.Y.K., Tan, R.S. & Acharya, U.R. (2018). Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. ScienceDirect.

Palaz, D., Collobert, R. & Magimai-Doss M. (2013). Estimating Phoneme Class Conditional Probabilities from Raw Speech Signal using Convolutional Neural Networks. arXiv.

Patilkulkarni, S. & Lakshmi H.C.V. (2013). Vanishing Moments of a Wavelet System and Feature Set in Face Detection Problem for Color Images. ResearchGate.

Power Standards Lab. (2019). PQube - Live World Map of Power Quality.

Seyfioğlu, M.S., Özbayoğlu, A.M. & Gürbüz S.Z. (2018). Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. IEEE.

Towards Data Science. (2019). A Deep Dive Into Imbalanced Data: Over-Sampling.

Uyar, M., Yildirim, S. & Gencoglu, M.T. (2008). An effective wavelet-based feature extraction method for classification of power quality disturbance signals. ScienceDirect.

Wang, S. & Chen, H. (2019). A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network. Elsevier.