Detection and Classification of Transmission Line Faults Using Convolutional Neural Networks

***Abstract*—Convolutional Neural Networks (CNNs) are among the best options for applications involving the detection and classification of transmission line faults. In this paper, a custom CNN architecture is developed, along with transfer learning using pre-trained CNN models, to enhance fault classification accuracy. Each model utilizes the voltage phasor A vs. time graph of all types of faults as input for the image classifier. A Simulink model of a 220 kV, 200 km transmission line is used to generate voltage signals under various fault conditions. Subsequently, these signals are encoded into two-dimensional Gramian Angular Field (GAF) images. The resulting image dataset is used to train different CNN models capable of extracting distinctive features from the images to detect and classify different fault conditions. In terms of detecting and classifying fault types, the proposed CNN model achieved an accuracy of 90-93%, outperforming pre-trained models such as ResNet and EfficientNet. A comparison between the proposed and other pre-trained models is also studied to show the relative strengths and weaknesses of different approaches.**

***Index Terms*—Electrical Fault Detection, Transmission Lines, Convolutional Neural Networks, Gramian Angular Field**

I. Introduction

Electrical power transmission deals with the transportation of electricity from a power generation source, such as a power plant, to an electrical substation. The interconnected system of power lines that enables this movement is called a transmission network. The network includes so many complex elements, which are subjected to unexpected and severe conditions. Sometimes the disturbances lead to electrical faults in the transmission line. It is evident that faults with overhead transmission lines contribute to roughly half of total faults. [1]

Failure to appropriately detect transmission line faults can lead to power outages, extensive power quality issues, and additional complications. The main goal is to detect fault and operate protection equipment in the shortest time possible so that the power system remains stable. It's important to accurately detect and classify faults on electrical power system transmission lines and clear them as quickly as possible with least disruption. [2] [3] The protective relaying of transmission lines has three main functions, which are to detect, classify, and locate faults on the transmission lines. [4] The fault diagnosis system needs to use the limited data available between the fault occurrence and the tripping of relays and circuit breakers. It must rapidly analyze this data to accurately diagnose the fault. [5]

Generally, transmission line faults are caused by short circuits within phases or between phases and ground. The main objective of fault detection is to determine the nature of a shorted connection from various possible combinations. In 3-phase systems, there are a total of ten possible states of transmission line faults, not including the healthy state. [6] These include single line-to-ground faults (AG, BG, CG), line-to-line faults (AB, BC, CA), double line-to-ground faults (ABG, BCG, CAG), and three-phase fault (ABCG). The process begins with fault detection (determining whether a fault is present or not) and advances to fault classification (identifying the type of fault if it exists). This involves the ability to recognize various types of faults.

The primary contributions of this research are as follows:

- Creating a dataset that is compatible with deep learning models, including CNNs and pre-trained models.
- Proposing a custom CNN architecture to enhance performance and increase evaluation accuracy.
- Implementing transfer learning with pre-trained CNN models for comparative analysis.

There are several machine learning and signal processing methods developed in previous studies for the fast and accurate identification of faults. Applying a pattern recognition technique could be useful in differentiating between faulty and healthy electrical power systems. It helps to differentiate among three phases, determining which phase of a three-phase power system is experiencing a fault. That's why the deep learning approach excels in detecting and classifying faults. Various deep learning techniques can be used for fault detection models for different components in power line systems. James et al. [7] proposed a joint detection technique using wind turbine (WT)-based Deep Neural Networks (DNNs) for islanding microgrids, with an average precision accuracy rate of 98.27%. Zhang et al. [8] investigated the use of LSTM and SVM algorithms for predicting line trips in power systems, reporting a prediction accuracy of 97.7%. Wang et al. [9] developed the FF-DNN-based architecture to supervise and detect gearbox faults of WTs using SCADA acquisition data. With the rapid advancement of computer vision in image classification, multiple approaches use the image representation of three-phase signals. The image representations allow the model to extract the features more accurately and efficiently. While time series data plays a crucial role in various aspects of power system monitoring, there are some limitations in detecting faults within transmission lines. Time series data is insufficient for detecting transmission line faults due to its focus on trends, limited information content, and susceptibility to noise. [10]

Convolutional Neural Networks (CNNs) excel in detection and classification by using their deep architecture to ex-

tract high-level features. Commonly used in computer vision, CNNs preserve spatial relationships in image data, improving classification accuracy by considering pixel proximity rather than treating each pixel independently. Encoding 1D three-phase data into 2D image data can significantly improve the efficiency of fault detection. [11]

Pre-trained CNN models enhance transmission line fault detection by using their ability to extract complex features from large datasets, which reduces the need for extensive training data and time. Pre-trained CNNs, such as ResNet, Effiecintnet, ConvNext, and NasNet have shown higher precision, recall, and F1-scores in fault classification tasks, making them more accurate and reliable. [12] Additionally, these models are robust to noise, ensuring consistent performance even in real-world conditions.

II. System Modelling and Dataset Generation

A. Transmission line modelling and Time series data generation

In this study, a 220 kV, 50 Hz transmission line, spanning 200 km, is utilized to model a three-phase power system network. One end of the transmission line is linked to a swing-type generator, while the opposite end is connected to a delta-wye transformer, which further supplies a distribution load at 33 kV. The entire network is modeled using Simulink, and various scenarios are simulated in the MATLAB environment to gather data. Different faults yield distinct electrical signal patterns, influenced by power system parameters. The system is simulated for various fault conditions, as well as for different parameters such as source impedance, line impedances, fault impedance, and fault distances. The resulting raw voltage of A phasor is collected as time-series data. This time-series data is then transformed into images.

Figure 1 shows a simplified model of a three-phase power system designed to analyze transmission line behaviour under fault conditions. Various parameters of the components mentioned in the above section are altered to gather data for different fault scenarios. The voltage and current time-series data is obtained from the scope using the MATLAB workspace.

The system model is simulated for each type of fault for every scenario and therefore data is generated. The process is described in here:

- Preprocessing and Feature Extraction: After obtaining scope data (e.g., p.Voltage A.da), optional wavelet denoising is applied, and the data is normalized between 0 and 1. Phase angles and radius values are calculated for each data point.
- Time Series Construction: During simulation runs, segments of preprocessed voltage data are extracted for the phasor A and assigned to the time-series variable for further analysis outside Simulink.

B. Image Encoding of Time Series

Our proposed model uses only the Voltage-A data as input. Wang and Oates suggested that one-dimensional time series can be converted into two-dimensional images using the Gramian Angular Field (GAF) algorithm. [13] In this study, the Voltage-A data is represented using the GAF method. When time-series data is mapped in a polar coordinate system, it forms the GAF representation. Each element of the Gramian matrix is the cosine of the sum of the angles derived from the GAF, with each Voltage-A signal containing 3,400 data samples. This method uses the polar coordinate system to provide a view of time series data. It shifts the data from amplitude changes to angle variations over time. By calculating trigonometric relationships between points, it captures time correlations based on angles. This technique is effective in highlighting cyclical or repeating patterns within the series. The Gramian Angular Summation Field (GASF) formula is stated in the Equation 1:

$$GASF = \begin{pmatrix} \cos(\phi_1+\phi_1) & \cdot & \cdot & \cos(\phi_1+\phi_N) \\ \cos(\phi_2+\phi_1) & \cdot & \cdot & \cos(\phi_2+\phi_N) \\ \cdot & \cdot & \cdot & \cdot \\ \cos(\phi_N+\phi_1) & \cdot & \cdot & \cos(\phi_N+\phi_N) \end{pmatrix} \quad (1)$$
$$= \tilde{T}'_{va} \cdot \tilde{T}_{va} - \sqrt{I - \tilde{T}'^2_{va}} \cdot \sqrt{I - \tilde{T}^2_{va}}$$

The GAF maintains the sequence of the data. By analyzing the diagonal (from top-left to bottom-right of the matrix), changes in the relationships between data points over time can be observed. The matrix size is $n \times n$, where n represents the number of time series data points.
Flow chart in the Figure 2 outlines the steps for computing the Gramian Angular Field:

Both GASF and GADF images function in the same way. For the sake of simplicity, we have solely utilized GASF in this research. Table I gives the overview about the number of data generated varying different network parameters during each fault event.

TABLE I: Number of data generated varying different network parameters

Parameters that were varied	**No. of Data Generated**
Source impedance	350
Fault impedance	130
	100
Transmission Line (prior fault location)	250
Transmission Line (after fault location)	250
	Total = 980

C. Effect on Voltage-A data for Different Fault in a Specific Scenario

There is a significant impact of fault distance, source impedance, fault resistance, and transmission line impedance on voltage signals in a transmission line. Fault distance, ranging from 1 km to 199 km, influences voltage, with a decrease observed near the fault point. Higher source impedance results in a reduced voltage drop at the fault location, while fault resistance, varying from 0.1 Ω to 100 Ω, affects the voltage magnitude at the fault. Transmission line impedance, including

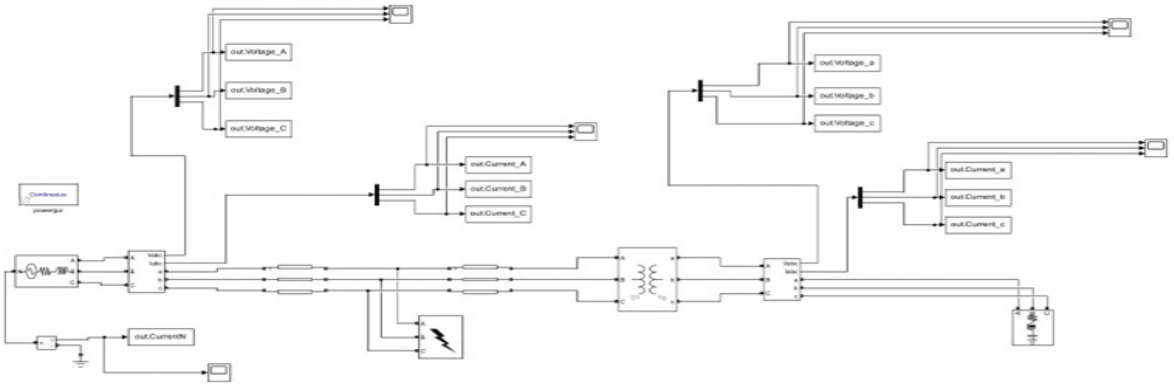


Fig. 1: Transmission line electrical network modeling

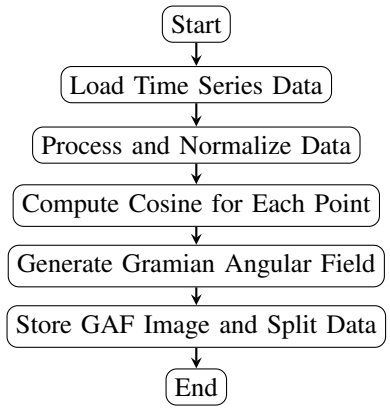


Fig. 2: Flowchart for the Proposed Framework

both positive and zero sequence impedances, plays a critical role in determining the fault voltage, aiding in fault detection and system protection.

III. Implementation of CNN Model

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data. The research focused on building a suitable architecture that can classify the GASF image by using a combination of sequential operation of convolution, pooling, and others.

A. The Convulutional Layer

The primary purpose of convolutional layers is to act as feature extractors. They convolve the input GASF image with kernels (filters) to capture key features from subspaces of the image. The processed image is transformed from 128×128 pixels to feature-extracted output of $2 \times 2 \times 1024$ pixels. For this layer, stride = 1, padding = 'same', and kernel size = 3×3. The number of filters changes with each forward propagation. The output of the convolution operation of each layer can be

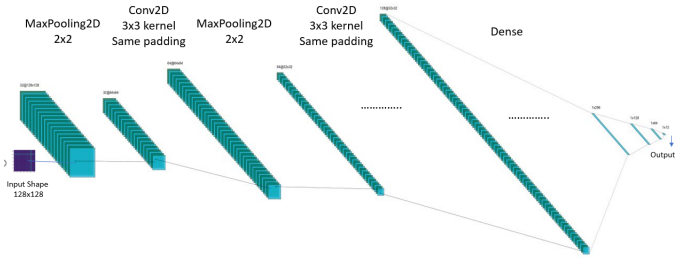


Fig. 3: Proposed CNN architecture as feature extractor

mathematically calculated as follows:

$$x_l^j = f\left(\sum_{i=1,2,\ldots,M} x_i^{l1} \times k_{ij}^l + b_j^l\right), j = 1, \ldots, N \quad (2)$$

where x_l^j denotes the j-th output map of the convolution layer for the filter, k, and the number of inputs, M. Further, x_j^{l1} represents the i-th input feature map of (l 1) layer, b_j^l implies the bias value j-th filter, and f denotes the activation function. [14]

B. Pooling Layer

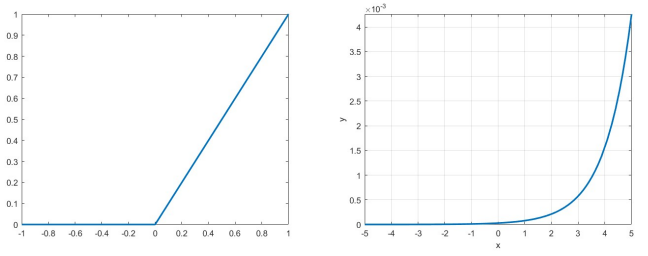
Pooling layers downsample feature maps by summarizing features within patches. Two common methods are average pooling and max pooling, which summarize the average and most activated presence of features, respectively. In this research, the pooling layer used is MaxPooling2D with a stride of 2 and a pool size of 2×2, as shown in Table II. The corresponding equation for this layer is provided.

$$[h]x_j^l = f\left(\beta_j^l.down(x_j^{l1}) + b_j^l\right), j = 1, \ldots, M \quad (3)$$

Here, x_j^l and x_j^{l-1} are the j-th output and input map.The f and down(.) represent the activation function and sub-sampling function, respectively. Two bias operations, multiplicative bias and additive bias for the j-th filter, are denoted by β_j^l and b_j^l. [14]

TABLE II: CNN Layer Structure

Layer Name	Output Shape	Parameters
Conv2D	128,128,32	896
BatchNormalization	128,128,32	128
MaxPooling2D	64,64,32	0
Conv2D	64,64,64	18496
BatchNormalization	64,64,64	256
MaxPooling2D	32,32,64	0
Conv2D	32,32,128	73856
BatchNormalization	64,64,128	512
MaxPooling2D	16,16,128	0
Conv2D	16,16,256	295168
BatchNormalization	32,32,256	1024
MaxPooling2D	8,8,256	0
Conv2D	8,8,512	1180160
BatchNormalization	8,8,512	2048
MaxPooling2D	4,4,512	0
Flatten	8192	0
Dense	512	4194816
Dropout	512	0
BatchNormalization	512	2048
Dense	256	131328
Dropout	256	0
BatchNormalization	256	1024
Dense	128	32896
Dropout	128	0
BatchNormalization	128	512
Dense	64	8256
Dropout	64	0
BatchNormalization	64	256
Dense	10	650



(a) Relu activation function (b) Softmax activation function

Fig. 4: Activation functions in dense layers

C. Fully Connected Layer

Fully connected layers in a CNN learn high-level representations and make predictions from extracted features. In this research, the $2 \times 2 \times 1024$ output tensor is converted into a 1D matrix of 4096 elements, connecting dense layers. Each dense layer uses the ReLU activation function to introduce non-linearity and solve the vanishing gradient issue. The final output is a probability for each class, computed using the SoftMax activation function.

IV. Proposed Pre-trained CNN Architectures & Algorithms

A. Pre-trained CNN Architectures

This research utilizes five different pretrained CNN architectures. The used architecture are being displayed. Each architecture possesses distinct features and unique characteristics for evaluating the GASF dataset. This paper focuses on leveraging these unique features to enhance the overall accuracy.

B. Proposed Transfer Learning Algorithm

Transfer learning decreases the requisite computational costs to build models for new problems. By re-purposing pre-trained models or pre-trained networks to tackle a different task. [15] There are two algorithm to support transfer learning.

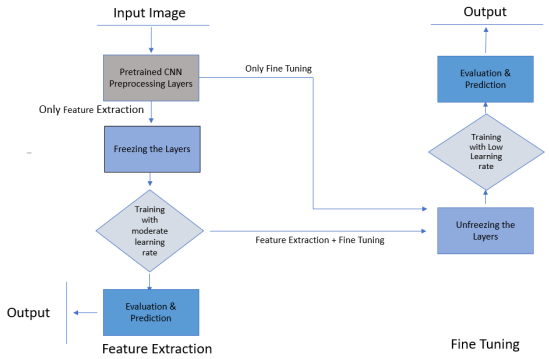


Fig. 5: Transfer learning flowchart

- Feature Extraction: Feature extraction is an efficient method for evaluating feedforward networks. In Transfer Learning, it reduces the number of epochs and training time by utilizing already extracted features. After preprocessing the dataset, all layers of the pre-trained model are frozen, and training on the custom dataset is conducted with a moderate learning rate. [15]
- Fine-Tuning: Fine-tuning a pre-trained model is faster, more cost-effective, and more efficient than training from scratch. In fine-tuning, the pre-trained model's layers are unfrozen and made trainable, but a low learning rate is essential to avoid overfitting, task mismatch, and performance issues like catastrophic forgetting. [15]

V. Result and Analysis

The CNN and pre-trained CNN models have offered distinct insights in evaluating the GASF dataset. These models effectively detect faults by leveraging their unique characteristics.

From Table III, the performance level of different pre-trained models are kept around 84-90%. Among the pretrained models, ConvNext achieved the highest accuracy on our GASF dataset. The consistent accuracy across all folds during cross-validation indicates the model's stability, effectively minimizing both over-fitting and under-fitting issues.

The proposed CNN model outperforms all pretrained models in accuracy, contributing significantly to the research.

TABLE III: Model Performance Analysis

	Cross Validation (1 of 5)	Cross Validation (1 of 5)	Cross Validation (2 of 5)	Cross Validation (2 of 5)	Cross Validation (3 of 5)	Cross Validation (3 of 5)	Cross Validation (4 of 5)	Cross Validation (4 of 5)	Cross Validation (5 of 5)	Cross Validation (5 of 5)	Test Accuracy (%)
Model Name	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	
CNN	88.024	0.4178	89.252	0.9399	90.673255	0.2943	87.1525	1.1125	90.5497	0.29223	93
ConvNext	87.105	0.3261	87.1667	0.348	86.2368	0.3469	87.291	0.33496	87.158	0.379	89
EfficientV2 Large	81.852	0.44284	81.655	0.498	83.076	0.466	83.446	0.3913	82.026	0.508	85
NesNet Large	81.173	0.5656	81.037	0.5719	80.173	0.622	79.678	0.575	81.47	0.554	86
ResNet 152V2	79.259	0.5979	79.555	0.6894	80.853	0.5104	78.073	0.6698	79.4935	0.5606	86
VGG19	84.753	0.4973	83.262	1.376	84.187	0.5406	83.879	1.69189	82.8906	0.593	86
Inception Resnet	78.518	0.5337	79.987	0.517	78.999	0.5097	78.9376	0.5621	79.246	0.531	84

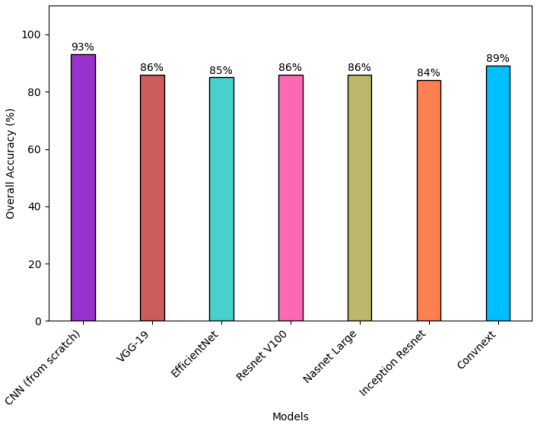


Fig. 6: Accuracy comparison of different CNN models

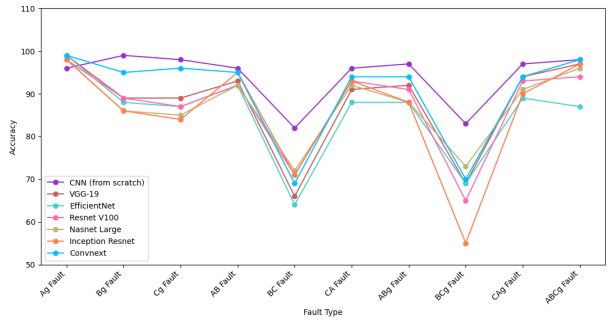
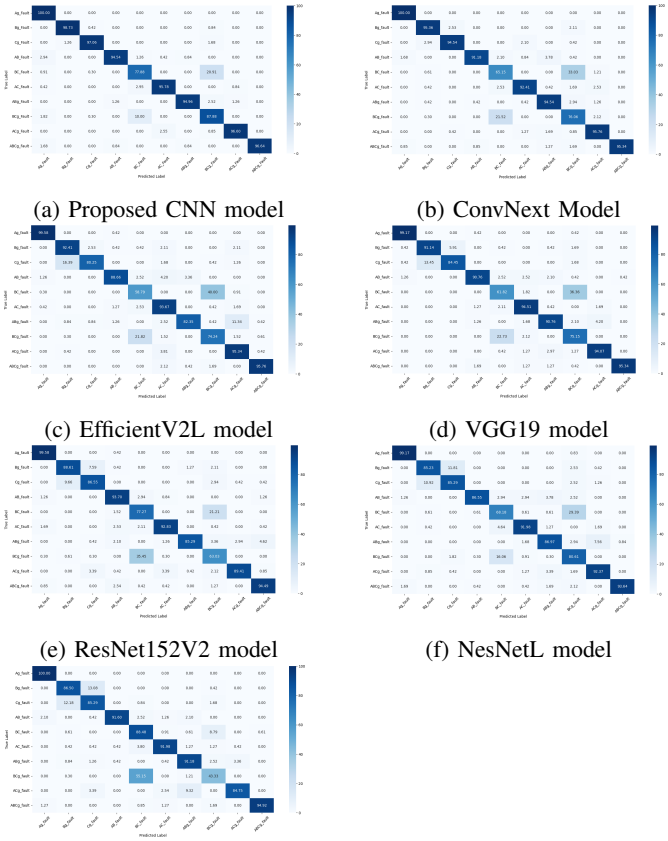


Fig. 7: Proposed CNN and pretrained CNN models performance analysis



(a) Proposed CNN model (b) ConvNext Model

(c) EfficientV2L model (d) VGG19 model

(e) ResNet152V2 model (f) NesNetL model

(g) Inception_ResNet model

Fig. 8: Confusion matrices of proposed CNN and pre-trained CNN models

Figure 7 shows the performance of various models across fault classes, with the proposed CNN achieving an accuracy of 90-93 %. Approximately 2,560 GASF images were tested, effectively detecting fault classes. However, since the dataset focuses on Voltage phase A, detecting faults in phases B and C, particularly BC and BCg faults, remains challenging. Expanding the dataset to include phases B and C could improve accuracy in these areas. Figure 8 presents the confusion matrices of the proposed and pretrained CNN models. Despite some limitations and errors, the models have successfully evaluated the dataset and provided efficient results in detecting transmission line faults.

VI. Conclusion

This paper presents a new approach to detect and classify transmission line faults, using a custom CNN architecture followed by transfer learning. By converting voltage phasor time-series data into GAF images, the method increases fault classification accuracy. Studies show that the model performs better than the traditional pre-trained models, achieving an accuracy rate of 90-93% for different fault types. Comparative analyses underline the reliability of the proposed CNN model, even in the presence of noise and varying conditions.

REFERENCES

[1] B. Ram and D. N. Vishwakarma, *Power System Protection and Switchgear*. McGraw-Hill Education, 2001.

[2] M. Jamil, S. K. Sharma, and R. Singh, "Fault detection and classification in electrical power transmission system using artificial neural network," *SpringerPlus*, vol. 4, no. 1, 7 2015. [Online]. Available: https://doi.org/10.1186/s40064-015-1080-x

[3] S. Singh and D. N. Vishwakarma, "Intelligent techniques for fault diagnosis in transmission lines — an overview," in *2015 International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, 2015, pp. 280–285.

[4] K. Krishnanand, P. Dash, and M. Naeem, "Detection, classification, and location of faults in power transmission lines," *International journal of electrical power energy systems*, vol. 67, pp. 76–86, 5 2015. [Online]. Available: https://doi.org/10.1016/j.ijepes.2014.11.012

[5] F. M. Shakiba, M. Shojaee, S. M. Azizi, and M. Zhou, "Real-Time sensing and fault diagnosis for transmission lines," *International journal of network dynamics and intelligence*, pp. 36–47, 12 2022. [Online]. Available: https://doi.org/10.53941/ijndi0101004

[6] M. Singh, B. Panigrahi, and R. P. Maheshwari, "Transmission line fault detection and classification," in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, 2011, pp. 15–22.

[7] J. Q. James, Y. Hou, A. Y. S. Lam, and V. O. K. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1694–1703, 2017.

[8] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using lstm networks and svm," *IEEE Access*, vol. 6, pp. 7675–7686, 2017.

[9] L. Wang, Z. Zhang, H. Long, J. Xu, and R. Liu, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1360–1368, 2016.

[10] D. Łuczak, "Machine fault diagnosis through vibration analysis: Time series conversion to grayscale and rgb images for recognition via convolutional neural networks," *Energies*, vol. 17, no. 9, 2024. [Online]. Available: https://www.mdpi.com/1996-1073/17/9/1998

[11] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, 3 2024. [Online]. Available: https://doi.org/10.1007/s10462-024-10721-6

[12] K. K, B. D, and V. K. D, "Transmission power line fault detection using convolutional neural networks." EAI, 6 2021.

[13] Z. Wang and T. Oates, "Imaging Time-Series to improve classification and imputation," 6 2015. [Online]. Available: https://arxiv.org/abs/1506.00327

[14] R. N. Toma, F. Piltan, K. Im, D. Shon, T. H. Yoon, D.-S. Yoo, and J.-M. Kim, "A bearing fault classification framework based on image encoding techniques and a convolutional neural network under different operating conditions," *Sensors*, vol. 22, no. 13, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/13/4881

[15] Y. Gao and K. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, 04 2018.