

AUTOMATIC FEATURES EXTRACTION BY TRANSFER LEARNING FOR TRANSMISSION LINE PROTECTION

FEZAN RAFIQUE^{1,2}, LING FU¹, MUHAMMAD HASSAN UL HAQ², RUIKUN MAI¹

Keywords: Computational intelligence; Fault detection; Machine learning; Power systems; Transmission lines.

This work proposes a deep learning-based fault detection and classification model with relaxed dataset requirements. The most arduous part of any deep learning-based solution is the availability of large, labeled datasets. The proposed method uses a pre-trained deep learning model as a starting point, then retrains the adapted weight in transfer arrangement for fault classifier applications. This strategy expedites training and reduces the need for exhaustive labeled dataset requirements by leveraging an existing model. The proposed model automatically extracts features from input signals to decide the state of power transmission lines, eliminating the complex need to craft features for fault classification algorithms manually. The model is thoroughly tested for a wide range of performance tests. (The dataset used in this work is publicly available at this URL: <https://www.kaggle.com/datasets/fezanrafique/wscce9busfaultdataset>).

1. INTRODUCTION

Transmission lines carry bulk energy from the source to the consumption. Since transmission lines extend for several hundred kilometers, they may be continuously exposed to harsh environmental conditions such as storms, snow, moisture, etc. [1]. These conditions make transmission lines more prone to electrical faults, such as short circuits between two conductors. Hence, the demand for robust transmission line protection against electrical faults is critical. A good protection system isolates the faulty part correctly within the minimum possible time [2]. Therefore, researchers are always looking for improved and faster methods. This paper proposes a data-driven generic yet robust fault detection and classification (FDC) mechanism for power transmission lines using transfer learning (TL) with deep-convolutional neural networks.

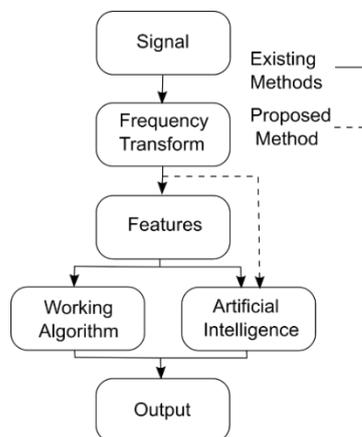


Fig. 1 – Hierarchy of existing and proposed fault detection methods.

Protection methods in contemporary literature are broadly classified into rule-based and data-driven algorithms. The hierarchy of these techniques is shown in Fig 1. Rule-based algorithms usually employ signal-processing techniques for feature extraction and decision-making. Signal-processing methods exploit the frequency content of signals for FDC tasks. Fault conditions exhibit frequency characteristics different from normal conditions. This property is utilized for developing FDC algorithms. The most common techniques for revealing frequency information include the Fourier transform (FT) and

the wavelet transform (WT) or discrete wavelet transform (DWT) [4]. These algorithms are usually based on two steps: the first is to acquire the frequency content of the signal through transform, which is usually called feature extraction; the second to develop a working algorithm by developing rules to utilize features for decision-making. Literature indicates WT (or DWT) gained more attention than FT due to its better capability in localizing time-frequency information. WT decomposes the signal into various frequency bands, which can be used as features for decision-making [5,6]. A literature survey indicates these techniques suffer weaknesses due to a lack of clarity and arbitrariness of feature selection criteria [7].

With new measurement devices in power systems, operational data increases the manifolds [8]. These massive data streams have attracted researchers to use artificial intelligence (AI) tools to solve power system operational challenges [9]. Most popular AI-based algorithms use support vector machines (SVM) and decision trees (DT) for designing fault classifiers [10–13]. However, both SVM and DT-based models are based on manually hand-crafted features using FT and WT. Then those features are put into the SVM or DT model for FDC, making them vulnerable like signal processing-based solutions. Deep-learning (DL) philosophy discourages using hand-crafted features. With the advent of DL techniques, the laborious need for feature mining can be eliminated.

Instead, in DL, features are learned by the model itself; this helps to achieve a more robust and generic solution [3]. However, such implementation requires enormous labeled datasets and computational resources for training the model [14]. Recently, some DL-based FDC models have been reported in the literature. Since FDC is essentially a pattern detection problem, this motivated researchers in power systems to apply DL tools for FDC tasks [15]. Such as K. Chen in [16] used sparse autoencoders and CNN for FDC in transmission lines. The implementation is computationally expensive and requires 250,000 training examples. Another study employing an RNN-based DL model for FDC is presented in [17]. This is also computationally expensive to train. Although these studies have claimed noteworthy accuracy results, these models are trained with random weight initialization, which requires exhaustive labeled datasets for acceptable performance. Acquiring enormous, labeled datasets from power system transmission lines is challenging. To tackle this stringent challenge of the availability of labeled datasets, the concept of TL or domain

¹ School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China, E-mail: lingfu@swjtu.cn

² Electrical Engineering Department, NED University, Pakistan, E-mail: fezan@neduet.edu.pk

transfer was introduced [19]. TL allows an existing DL model to be retrained on a different dataset with relatively less labeled data. Since the availability of fault data on transmission lines may be limited. Considering the literature review, this paper proposes a fully automatic solution by integrating feature extraction by employing a TL-based DL model for FDC in extra high voltage (EHV) transmission lines. Hence, the need for laborious feature engineering is eliminated while keeping the training resources limited.

2. TRANSFER LEARNING

The key element of success in DL models is the availability of extensive labeled datasets. However, the availability of labeled datasets may be difficult in some real-world scenarios. This problem might be overcome through TL (or domain transfer) [20,21]. In machine learning, the concept of TL exists, where knowledge acquired in one domain (source) may be utilized to enhance the learning performance of another domain (target), which might lack the volume of labeled data [20]. To understand TL, it is essential to define terms domain and task. Domain D consists of two parts: 1) feature space χ and 2.) probability distribution $P(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in \chi$, making $D = \{\chi, P(X)\}$. Given a domain D , task τ consists of two parts: 1) a label space y and objective predictive function $f(\cdot)$, making $\tau = \{y, f(\cdot)\}$. The predictive function is achieved by learning through training data $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in y$. $f(\cdot)$ can be used to determine $f(x)$, given any value of x . With these terms, TL can be defined; considering the source domain (*that has ample labeled data*) D_S , task τ_S , and the target domain (*which lacks labeled data*) D_T , task τ_T . TL improves the learning $f_T(\cdot)$ for D_T through the knowledge of D_S and τ_S while $D_S \neq D_T$ and $\tau_S \neq \tau_T$ [21]. TL has been used in various applications such as computer vision, medical diagnosis, natural language processing, *etc.*, Huatao Jiang in [22] used NVIDIA Dave-2 dataset and VGG16 model in TL arrangement to predict steering angles for self-driven cars. The designed model claims better performance than the NVIDIA model. Liang in [23] used TL to improve the diagnosis of pediatric pneumonia. The designed model was tested on real patient datasets and has reported an accuracy of 90.7%. Medical diagnosis applications using Electrocardiogram (ECG) scans and TL are reported in [24,25]. TL has also been used in text classification applications. Chongyu Pan in [26] proposed TL for designing a low-resource word embedding model for semantic text. The authors claimed better accuracy than other low-resource methods. Recently, TL has been used for real-time object identification at the edge-computing level, using customized GPU architecture and model compression techniques [27]. This has led to a myriad of applications that can benefit from data-driven methods and thus takes this research article closer to real-world implementation.

3. PROPOSED METHOD & DATA CURATION

This section provides implementation details of the proposed transfer learning architecture.

3.1. ALEXNET & TRANSFERRED ALEXNET

The proposed design uses AlexNet (AN) as the source domain of the algorithm. AN, proposed by Krizhevsky, is a landmark computer vision (CV) DL model designed using CNN [28]. AN model contains five convolutional layers and three fully connected layers. The convolutional layers are based on the convolution operation between the weight matrix and output from the previous. This can be modeled with the following equation:

$$h_j = \sigma \left(\sum_{k=1}^n \sum w_{ij} x_{ij,k} + b \right), \quad (1)$$

w_{ij} is the weight matrix of the convolution layer, which starts with random initialization and is later updated through backpropagation. This work, however, uses a pre-trained weight matrix of AN model. $x_{ij,k}$ corresponds to the previous layer's output, which can be multichannel; b is the bias. After performing the convolution, the activation function σ is computed. The activation function used in AN is called rectified linear unit (ReLU), mathematically expressed as:

$$\text{ReLU}(x) = \max(0, x). \quad (2)$$

The fully connected layers after the convolution layers also use ReLU activation. Forward pass for a fully connected layer can be expressed as:

$$z = w^T x + b, \quad (3)$$

w is the weight vector for the layer, x is input from the previous layer, and b is the bias. The final fully connected layer has softmax activation. Softmax activation maps the layer input to the multi-class outputs in terms of probability. For K classes, the input is mapped to a vector of length of K . The highest probability is assigned as the final classification. Softmax activation is mathematically given by:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad (4)$$

z is input vector, the softmax activation is ratio of exponential of the given input and sum of all the exponentials. Originally AN has 1000 output classes, in this work the output classes are changed from $K = 1000$ to $K = 11$, because the fault classifier has total 11 possible outputs, such as no Fault, AG (fault between phase A and Ground), AB, *etc.* This modified AN is termed as Transferred AlexNet (TAN). Modification utilized the pre-learned weight matrix of AN. For training, the categorical cross entropy loss function is used as the cost function for the classifier. It is given by:

$$J(W) = - \sum_{i=1}^N \sum_{j=1}^K y_{ij} \log \left(\frac{e^{w_j^T x_i}}{\sum_{m=1}^K e^{w_m^T x_i}} \right), \quad (5)$$

y is the true label assigned with the dataset; exponential terms are feedforward vector value for each input. For training the model, the objective is to minimize the cost function. It is done by updating values of weight matrix w . Stochastic Gradient Descent with Momentum (SGDM) minimizes $J(W)$. This iterative process is given by:

$$v_t = \mu v_{t-1} - \alpha \frac{\partial J}{\partial w_{t-1}}, \quad (6)$$

$$w_t = w_{t-1} + v_t,$$

α is the learning rate, μ is the value of momentum. It may be noted that bias b is also updated using the same rules listed in (6). The complete architecture of the AN model is summarized in Table 1. The TAN model is shown in Fig 2.

3.2. WAVELET TRANSFORM

The TAN model requires an image as input. WT is used to obtain the image equivalent of the power system time series signal. Power transmission lines exhibit a non-stationary behavior during dynamic events such as faults, *etc.* This is quantified using WT. Essentially WT is a convolution operation between a signal $x(t)$ and set of functions generated by scaling and dilating the mother wavelet $\psi(t)$. It can be expressed as [29]:

$$C(m, \tau) = \frac{1}{\sqrt{m}} \int_0^{\infty} x(t) \psi^* \left(\frac{t-\tau}{m} \right) dt, \quad (7)$$

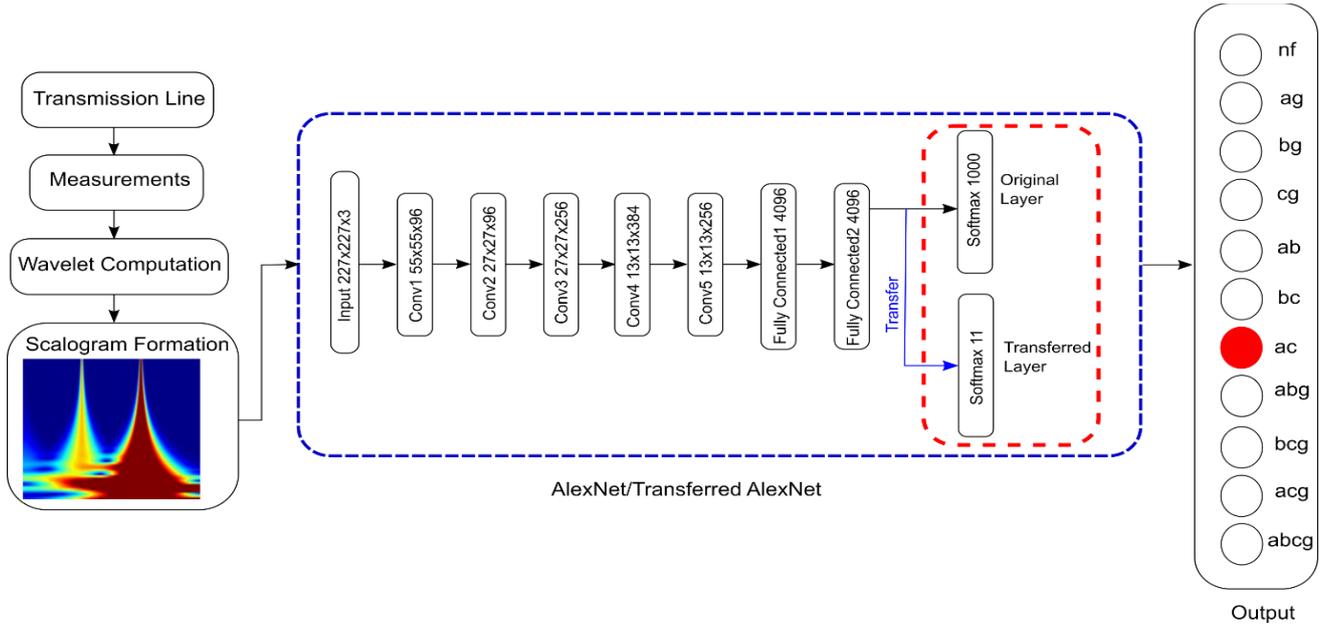


Fig. 2 – Transferred AlexNet architecture and complete workflow of the proposed model, redline before the softmax layer showing the transfer process.

Here, $C(m, \zeta)$ is coefficient matrix that contains time-scale version of the signal $x(t)$. ψ^* is the complex conjugate of the mother wavelet, m is the scaling factor, and ζ is the dilation factor. This study used an Analytic Morlet (Gabor) wavelet and ten scales. Assigning a colormap to magnitude intensity gives a colored scalogram in image format.

3.3. DATA CURATION

Western System Coordinating Council (WSCC) 9 bus framework is used for acquiring datasets. System parameters for the WSCC 9 bus system are adapted from [30]. A single-line representation of this benchmark system is shown in Fig 3. Simulations are performed in a MATLAB environment with a sampling frequency of 12 kHz. Signals (instantaneous rms values of current and voltage) in the time series format were acquired and subjected to added white Gaussian noise (AWGN) with a signal-to-noise ratio (SNR) value of 40 dB. Dataset details are reported in Table 2. After time-series data are generated, they must be converted into image format. The data-gathering process was further split into two stages: 1) for training and 2) for external testing and deployment. For training, time-series data of one cycle was taken as one example. Three-phase operational data was stacked into a single time series. For the deployment or external testing, it is proposed to split one cycle of time-series data into multiple time-series instances using a sliding window approach, as shown in Fig 4. The stride length (denoted with D henceforth) for the sliding window can be determined using the desired speed.

3.4. MODEL TRAINING & HYPERPARAMETERS

Hyperparameters used to train the TAN model are listed in Table 3. The optimizer chosen is stochastic gradient descent. It differs from the commonly cited gradient descent optimizer in the sense that the TAN model updates its weights after running a few examples – 100 in our case – instead of running the whole dataset before updating weights. The momentum feature speeds up the optimization by adaptively changing the base rate of updating weights. The number of epochs required to train TAN is the number

of times the whole training dataset is required to flow through the model to update weights. Unsurprisingly, TAN requires only one epoch to reach desired accuracy level due to the domain transfer from the original AN.

Table 1
Architecture details of AlexNet model

S. No	Layer Type	Filter Size	Feature Size	Activation
1	Input	N.A.	227×227×3	N.A.
2	Conv 1	11 × 11	55×55×96	ReLU
3	Max Pool 1	3 × 3	27×27×96	N.A.
4	Conv 2	5 × 5	27×27×256	ReLU
5	Max Pool 2	3 × 3	13×13×256	N.A.
6	Conv 3	3 × 3	13×13×384	ReLU
7	Conv 4	3 × 3	13×13×384	ReLU
8	Conv 5	3 × 3	13×13×256	ReLU
9	Max Pool 3	3 × 3	6×6×256	N.A.
10	Dropout 1	rate = 0.5	6×6×256	N.A.
11	Fully Connected 1	N.A.	4096	ReLU
12	Dropout 2	rate = 0.5	4096	N.A.
13	Fully Connected 2	N.A.	4096	ReLU
14	Fully Connected 3	N.A.	1000	Softmax

4. RESULTS

This section describes the performance results obtained using the TAN model for FDC. Results present both model performance metrics and operational performance of the TAN model under various operating conditions that may arise during power system operations.

4.1. ACCURACY OF MODEL

Key performance indicators for any deep learning model are its accuracy, precision, and recall. TAN model attained 99.1 % accuracy for FDC on the test dataset. The confusion matrix for fault classifications using the TAN model is shown in Fig 5.

Precision, recall, and F1 score indicators are listed in Table 4. Each class had 500 examples. The model reported 99.04 % accuracy on test data, which is at par with the contemporary transmission line protection algorithms standards.

Table 2
Dataset details

Parameter	Train Data	Test Data
Measuring Node	Bus 7	Bus 4, 7, 9
Lines Involved	7-8	4-5, 4-6(Measuring node 4) 5-7, 7-8(Measuring node 7) 6-9, 8-9(Measuring node 9)
Fault Resistance	1Ω	1Ω
Fault Distance	50km	5km, 25km, 50km, 75km
Fault Types	No fault (NF), ag, bg, cg, ab, bc, ac, abg, bcg, acg, abc	No fault (NF), ag, bg, cg, ab, bc, ac, abg, bcg, acg, abc
Inception Angle	0°	0° to 180°
Noise	40dB	20dB to 40dB
Examples Count	2500 for each class	500 for each class

Table 3
Hyperparameters for TAN model

Hyperparameter	Value
Optimizer	Stochastic gradient descent with momentum
Minibatch Size	100
Learning Rate	0.0001
Momentum	0.9
Epochs	1

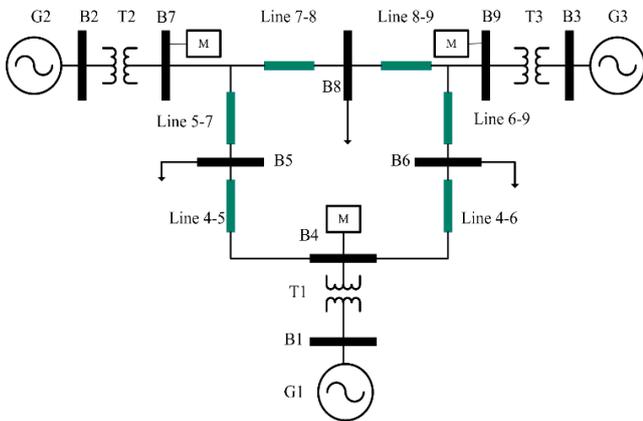


Fig. 3 – WSCC 9 bus system used to create the dataset.

Table 4
Performance indicators of TAN model

Class	Classified	Accuracy	Precision	Recall	F1
AG	500	100 %	1	1	1
BG	500	100 %	1	1	1
CG	500	100 %	1	1	1
AB	500	100 %	1	1	1
BC	500	100 %	1	1	1
AC	500	100 %	1	1	1
ABG	500	100 %	1	1	1
BCG	500	100 %	1	1	1
ACG	500	100 %	1	1	1
ABCG	447	99.04 %	1	0.89	0.94
NF	553	99.04 %	0.9	1	0.95

4.2. TIME PERFORMANCE OF MODEL

Time performance illustrates the minimum signal duration required to classify its type after fault inception. The time performance of the TAN model depends on the stride length of the moving window used in computing WT. Keeping the overlap stride smaller will decrease the classification time. However, it will increase the processing requirements to compute WT. The test is performed for all kinds of faults occurring at a distance of 25 km from the measuring node on lines 7-8. For $D = 1$, test results for line-ground fault are reported in Fig 7. It shows the output state of the TAN model

as data is fed to it in real-time. All faults were started at time instant 74.7ms.

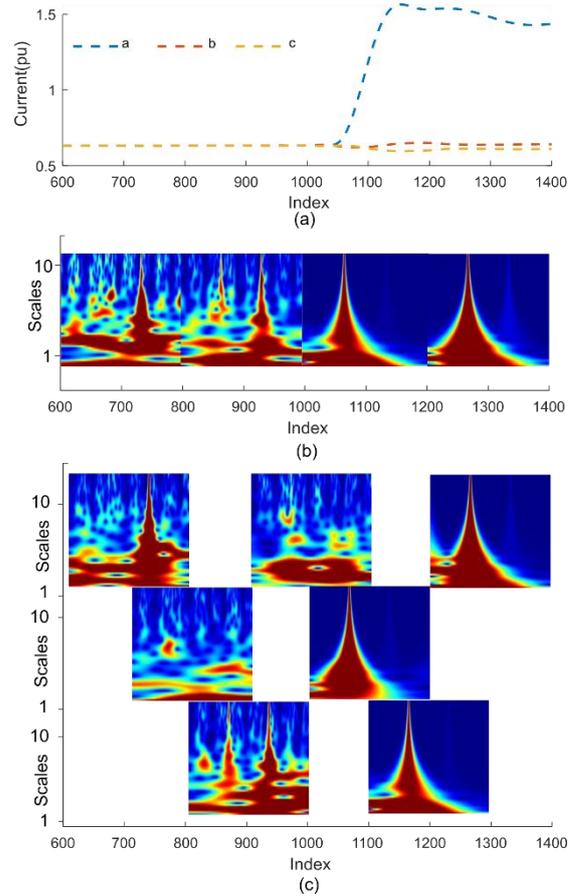


Fig. 4 – Data instance example: a) three-phase rms time series with two cycles of normal and one cycle of ag fault; b) WT of rms signal with a stride length of 200; c) WT of rms signal in moving window fashion with a stride length of 100.

		Confusion Matrix											
		AB	ABCG	ABG	AC	ACG	AG	BC	BCG	BG	CG	NF	
Output Class	AB	500	0	0	0	0	0	0	0	0	0	0	100%
	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	ABCG	0	447	0	0	0	0	0	0	0	0	0	100%
	0.0%	8.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	ABG	0	0	500	0	0	0	0	0	0	0	0	100%
	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	AC	0	0	0	500	0	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	ACG	0	0	0	0	500	0	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	AG	0	0	0	0	0	500	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	BC	0	0	0	0	0	0	500	0	0	0	0	100%
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
BCG	0	0	0	0	0	0	0	500	0	0	0	100%	
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	
BG	0	0	0	0	0	0	0	0	500	0	0	100%	
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	
CG	0	0	0	0	0	0	0	0	0	500	0	100%	
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	
NF	0	53	0	0	0	0	0	0	0	0	500	90.4%	
0.0%	1.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	9.6%	
100%	89.4%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	89.0%	
0.0%	10.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.0%	

Fig. 5 – Confusion matrix showing model performance for each classification target.

After inception, the model takes some time to give the classification output. Time is calculated by counting the number of steps elapsed between fault inception and classification. A time performance test was also performed for $D = 50$ and $D = 100$. Comprehensive time-impedance results are shown with a scatter plot in Fig 7.

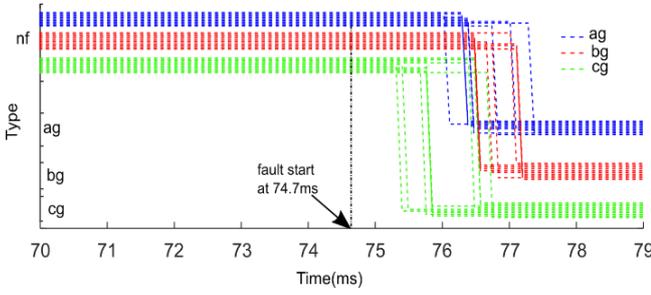


Fig. 6 – Time performance of TAN model with $D = 1$, ten examples shown for single line to ground faults.

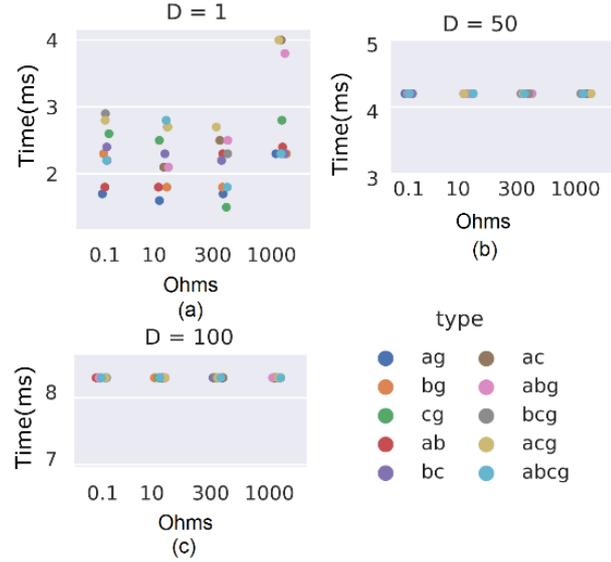


Fig. 7 – Time performance of TAN model with different window strides and fault impedance: a) time-impedance matrix for $D = 1$; b) time-impedance matrix for $D = 50$; c) time-impedance matrix for $D = 100$.

4.3. FAULT DISTANCE & INCEPTION ANGLE TEST

Distance between the fault point and measurement node can impact the performance of the protection algorithm. It is usually referred to as the reach of the relay. The proposed TAN model is tested for variations in the fault distance. The model is also verified for fault inception angles. Tests were performed on lines 7–8 by creating ab faults of 1Ω . For distance test, distances of 5 km, 25 km, 60 km, & 75 km were used. For inception angle testing, inception angles from 0° to 180° with a difference of 30° were considered. Results are shown in Fig. 8 (for $D = 50$). Distance and inception angle do not impact the performance of the TAN model.

4.4. MEASURING NODES & FAULT OBSERVABILITY

A comparison test was conducted to evaluate model performance based on measurement nodes. Firstly, the measurement node was fixed (using Bus 7 for measurements), and a fault was applied on lines 4-6, 4-5, 8-9, 6-9, 5-7. TAN model successfully classified all faults within 4.2 ms of the fault signal (using $D = 50$ for WT). This is illustrated in Fig. 9a. Secondly, the measurement node was changed to the closest bus near the faulty line. The model successfully classified all faults in 4.2 ms despite the difference in absolute values of measured signals. This is shown in Fig. 9b. These results indicate the potential of the proposed TAN model to be used for wide-area protection applications or backup protection.

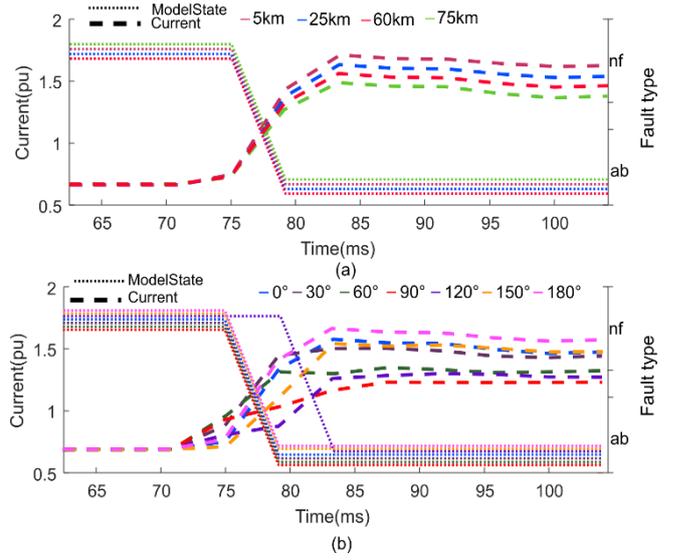


Fig. 8 – Performance of TAN model during variation in: a) fault distances; b) inception angle.

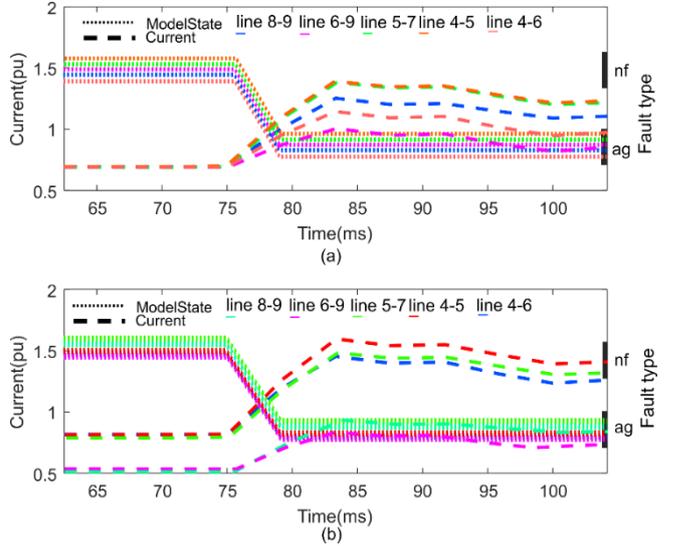


Fig. 9 – Variation in performance of TAN model based on measurement nodes: a) fault classification performance on lines with measurements taken from bus 7; b) fault classification performance on line with measurements taken from the closest bus.

5. CONCLUSIONS

An efficient, simple, and fully autonomous fault classifier for fault detection and classification in the transmission network is proposed in this paper. The classifier uses TL on a pre-trained and well-proven computer vision model, AlexNet. This implementation benefitted from the advantages of DL for a feature-free generic solution, and TL reduced the exhaustive need for large datasets and training mechanisms. The training process required only one epoch to complete. The designed model automatically exploits the signal's frequency content and decides the type of fault occurring on the transmission line within the quarter cycle (4.2 ms for $D = 50$) of the fundamental period. The method is rigorously tested for operational conditions such as fault resistance (up to $1\text{ k}\Omega$), distance, inception angle, etc. Test results indicate the capabilities of the proposed method for deployment due to faster performance, resilience towards operational conditions and lesser requirements for data pre-processing.

ACKNOWLEDGEMENTS

This work is sponsored by the National Natural Science Foundation of China under grant 51777173.

Received on 17 January 2023

REFERENCES

1. A. Mukherjee, P.K. Kundu, A. Das, *Transmission line faults in power system and the different algorithms for identification, classification and localization: a brief review of methods*, J. Inst. Eng. Ser. B, **102**, 4, pp. 855–877 (2021).
2. S. Belagoune, N. Bali, A. Bakdi, B. Baadji, K. Atif, *Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems*, Meas. J. Int. Meas. Confed., **177**, p. 109330 (2021).
3. Y. Lecun, Y. Bengio, G. Hinton, *Deep learning*, Nature, **521**, 7553, pp. 436–444 (2015).
4. K. Chen, C. Huang, and J. He, *Fault detection, classification and location for transmission lines and distribution systems: a review on the methods*, High Volt., **1**, 1, pp. 25–33 (2016).
5. Z. He, L. Fu, S. Lin, Z. Bo, *Fault detection and classification in EHV transmission line based on wavelet singular entropy*, IEEE Trans. Power Deliv., **25**, 4, pp. 2156–2163 (2010).
6. C. Pang, M. Kezunovic, *Fast distance relay scheme for detecting symmetrical fault during power swing*, IEEE Trans. Power Deliv., **25**, 4, pp. 2205–2212 (2010).
7. S.R. Fahim, Y. Sarker, S.K. Sarker, M.R.I. Sheikh, S.K. Das, *Self attention convolutional neural network with time series imaging based feature extraction for transmission line fault detection and classification*, Electr. Power Syst. Res., **187**, p. 106437 (2020).
8. H. Yang, X. Liu, D. Zhang, T. Chen, C. Li, W. Huang, *Machine learning for power system protection and control*, Electr. J., **34**, 1, p. 106881 (2021).
9. E. Hossain, I. Khan, F. Un-Noor, S.S. Sikander, M.S.H. Sunny, *Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review*, IEEE Access, **7**, pp. 13960–13988 (2019).
10. M. Jaya Bharata Reddy, P. Gopakumar, D.K. Mohanta, *A novel transmission line protection using DOST and SVM*, Eng. Sci. Technol. an Int. J., **19**, 2, pp. 1027–1039 (2016).
11. H.R. Baghaee, D. Mlakic, S. Nikolovski, T. Dragicevic, *Support Vector Machine-Based Islanding and Grid Fault Detection in Active Distribution Networks*, IEEE J. Emerg. Sel. Top. Power Electron., **8**, 3, pp. 2385–2403 (2020).
12. S. Kar, S.R. Samantaray, M.D. Zadeh, *Data-Mining Model Based Intelligent Differential Microgrid Protection Scheme*, IEEE Syst. J., **11**, 2, pp. 1161–1169 (2017).
13. D.P. Mishra, S.R. Samantaray, G. Joos, *A combined wavelet and data-mining based intelligent protection scheme for microgrid*, IEEE Trans. Smart Grid, **7**, 5, pp. 2295–2304 (2016).
14. H. Lee et al., *An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets*, Nat. Biomed. Eng., **3**, 3, pp. 173–182 (2019).
15. Z. He, S. Lin, Y. Deng, X. Li, Q. Qian, *A rough membership neural network approach for fault classification in transmission lines*, Int. J. Electr. Power Energy Syst., **61**, pp. 429–439 (2014).
16. K. Chen, J. Hu, J. He, *Detection and Classification of Transmission Line Faults Based on Unsupervised Feature Learning and Convolutional Sparse Autoencoder*, IEEE Trans. Smart Grid, **9**, 3, pp. 1748–1758 (2018).
17. F. Rafique, L. Fu, R. Mai, *End to end machine learning for fault detection and classification in power transmission lines*, Electr. Power Syst. Res., **199**, June, p. 107430 (2021).
18. K. Weiss, T.M. Khoshgoftaar, D.D. Wang, *A survey of transfer learning*, J Big Data., **3**, 1, Springer International Publishing, 2016.
19. S. Bozinovski, *Reminder of the First Paper on Transfer Learning in Neural Networks, 1976*, Informatica, **44**, 3 (2020).
20. F. Zhuang et al., *A comprehensive survey on transfer learning*, Proc. IEEE, **109**, 1, pp. 43–76 (2021).
21. S.J. Pan, Q. Yang, *A Survey on Transfer Learning*, IEEE Trans. Knowl. Data Eng., **22**, 10, pp. 1345–1359 (Oct. 2010).
22. H. Jiang, L. Chang, Q. Li, D. Chen, *Deep transfer learning enable end-to-end steering angles prediction for self-driving car*, 2020 IEEE Intelligent Vehicles Symposium (IV), **IV**, pp. 405–412 (Oct. 2020).
23. G. Liang, L. Zheng, *A transfer learning method with deep residual network for pediatric pneumonia diagnosis*, Comput. Methods Programs Biomed., **187**, p. 104964 (2020).
24. Mohebbanaaz, L.V.R. Kumar, Y.P. Sai, *A new transfer learning approach to detect cardiac arrhythmia from ECG signals*, Signal, Image Video Process., **16**, 7, pp. 1945–1953 (2022).
25. K. Weimann, T.O.F. Conrad, *Transfer learning for ECG classification*, Sci. Rep., **11**, 1, pp. 1–12 (2021).
26. C. Pan, J. Huang, J. Gong, X. Yuan, *Few-Shot Transfer Learning for Text Classification with Lightweight Word Embedding Based Models*, IEEE Access, **7**, pp. 53296–53304 (2019).
27. ****Nota uses AI to make roadways safer and more efficient*. <https://resources.nvidia.com/en-us-metropolis-software-success-stories/embedded-nota-soluti> (accessed Apr. 10, 2022).
28. A. Krizhevsky, I. Sutskever, G.E. Hinton, *ImageNet classification with deep convolutional neural networks*, Commun. ACM, **60**, 6, pp. 84–90 (2017).
29. N.E. Huang et al., *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis*, Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci., **454**, 1971, pp. 903–995 (Mar. 1998).
30. A.S. Al-Hinai, *Voltage collapse prediction for interconnected power systems*, West Virginia University, 2000.