

2023

Recurrent Neural Networks RNNs and Decision Tree DT Machine Learning-Based Approaches For Transmission System Faults Diagnosis

Sayed Abuanwar, Mohammed Saeed, Hanan Mosalem, Hatem Khater

Follow this and additional works at: <https://digitalcommons.aaru.edu.jo/erjeng>

Recommended Citation

Abuanwar, Mohammed Saeed, Hanan Mosalem, Hatem Khater, Sayed (2023) "Recurrent Neural Networks RNNs and Decision Tree DT Machine Learning-Based Approaches For Transmission System Faults Diagnosis," *Journal of Engineering Research*: Vol. 7: Iss. 5, Article 10.
Available at: <https://digitalcommons.aaru.edu.jo/erjeng/vol7/iss5/10>

This Article is brought to you for free and open access by Arab Journals Platform. It has been accepted for inclusion in Journal of Engineering Research by an authorized editor. The journal is hosted on [Digital Commons](#), an Elsevier platform. For more information, please contact rakan@aar.edu.jo, marah@aar.edu.jo, u.murad@aar.edu.jo.

Recurrent Neural Networks RNNs and Decision Tree DT Machine Learning-Based Approaches For Transmission System Faults Diagnosis

Sayed Abulanwar^{1,2}, Mohammed Saeed^{2,3}, Hanan Mosalem³, Hatem Khater¹

¹ Faculty of Engineering, Horus University-Egypt, New Damietta 34518, Egypt

² Electrical Engineering Department, Faculty of Engineering, Mansoura University, 35516, Mansoura, Egypt

³ Communications Engineering Department, Mansoura Higher Institute of Engineering and Technology, El-Mansoura, Egypt

Abstract- Accurate and prompt detection of system faults are crucial to maintain sufficient protection of system equipment, avoid false tripping, and cascaded failures. This paper presents a comprehensive study on the effectiveness of machine learning techniques for electrical fault detection and classification. Specifically, a comparative analysis is conducted between two prominent algorithms: Recurrent Neural Networks (RNNs) and Decision Tree (DT). The study employs a dataset comprising real-world electrical fault scenarios to evaluate the performance of RNNs and DT in identifying and categorizing faults. While DT algorithm showed slightly better accuracy in some cases, the RNN exhibited better generalization capabilities and a lower risk of overfitting. The analysis involves various performance metrics such as accuracy, precision, recall, and confusion matrices to comprehensively assess the algorithms' capabilities. The findings provide valuable insights into the strengths and limitations of each approach in the context of electrical fault management. This paper contributes to the selection of suitable techniques based on specific application requirements, advancing the field of predictive maintenance and fault mitigation in electrical systems.
Keywords- Decision Tree, Electrical Faults, Fault Classification, Fault Detection, Machine Learning, Recurrent Neural Networks.

I. Introduction

Security of power systems is currently threatened owing to the steady growth of renewable energy resources (RERs) which aggravates their complexity and causes voltage instabilities raised by their intermittencies [1].

As transmission systems represent a pivotal role within the power system, nowadays, their protection is gaining increased popularity owing to the high-power demand and their complex structures caused by suboptimal extensions. In this context, the advent of high-capacity electrical generating plants and the establishment of synchronized grids spanning disparate geographical regions necessitate rapid fault detection and the seamless operation of protection equipment. This expeditious response is crucial to maintaining the power system's stability [1].

During emergencies, conventional protection systems are liable to maloperation caused by inaccurate discrimination between recurrent RERs disturbances and incepted faults [2-4] which may cause series blackouts [5]. Enhancement of conventional protection systems' identification of faults will avoid false triggering incurred by RERs and/or loads operational changes thus improving system reliability. The malfunctioning of protection systems can be attributed to the waveforms resemblance of both normal disturbances and system faults [6]. Therefore, it is crucial to prioritize efficacious data processing and feature extraction techniques

for analyzing the operational behavior of such systems. Various studies exist on waveforms characterization with the central premise of obtaining pronounced markers for accurate fault identification [7-14]. An enhanced Fast Fourier Transform-based method is devoted for extracting distinguished voltage features for identifying voltage dips is introduced in [7-9].

Post-fault, the harmonic content is analyzed for accurate identification of voltage dips in distribution systems in [10]. Kalman filter-based algorithms are also involved in [11-12] for fault identification in microgrids (MGs). A discrete wavelet transform (DWT) integrated with support vector machine (SVM) in [13] for fault extraction. Likewise, [14] presented an optimized DWT to discriminate transmission system faults. Owing to their excellent features extraction capability, deep learning (DL) techniques are extensively employed in fault detection techniques [15-24].

Being a data-driven approach, the parameters of the DL model undergo optimization through a loss function, enabling automatic feature extraction without manual intervention. In [15-19], DL-based approach is employed in fault occurrence determination in modular multi-level converter (MMC) via processing voltage data. Various voltage dips and other disturbance features are being classified through convolution neural network (CNN) in [20-21]. A novel intermediate distribution alignment (DA) algorithm is introduced in [22] for fault transfer diagnosis. Though the aforesaid studies showed interesting results regarding fault identification via DL, several concerns still exist. For instance, during the fault diagnosis phase, the influence of suppression on the judgement output of the disturbance state during fault is not considered in several works. Besides, DL is adversely influenced by confusing samples which aggravates classifier performance. On the other hand, increasing model complexity for obtaining precise results will result in overfitting and worsening training model. Detecting and accurately categorizing faults in transmission lines is of paramount importance, demanding swift resolution [23]. Effective fault detection systems enable dependable, secure, and rapid relay operations, playing a pivotal role in maintaining system integrity. Employing pattern recognition techniques is advantageous in distinguishing between healthy and faulty power systems and identifying fault phases within a three-phase power system.

This paper presents a comparative analysis of the application of two distinct machine learning algorithms, RNNs and DT, for the detection and classification of electrical faults. With the

increasing importance of maintaining the reliability and safety of electrical systems, the timely identification of faults is of paramount significance. This paper demonstrates the effectiveness of artificial intelligence in detecting and classifying electrical faults, particularly in a three-phase system.

The rest of the paper is organized as follows. Section 2 discusses the problem statement, material and methods used in the paper is explained in detail in section 3. The proposed methodology is discussed in section 4, while the simulation results and discussion are discussed in section 5. Finally, the paper is concluded in section 6.

II. Problem Statement

Conventional methods for detecting faults often rely on manual observation and periodic checks, which can lead to delays in uncovering issues and recognizing potential hazards. As electrical systems become increasingly complex, there is a growing need to adopt more robust and automated techniques for fault detection. This study aims to address the challenge of online electrical fault detection and categorization by employing recurrent neural networks. In this proposed approach, the input data consists of the three-phase currents and voltages from one end of the electrical system. The combination of a feedforward neural network and the backpropagation algorithm is utilized to detect and classify faults, analyzing each of the three phases involved. By utilizing real-time data and the sequential characteristics of electrical signals, the intended outcome of this method is to enhance the accuracy and efficiency of problem identification and classification. Such progress leads to optimized maintenance schedules and a reduction in costly periods of system downtime. To validate the adoption of the recurrent neural network, a comprehensive analysis has been conducted. This problem statement serves as a foundation for conducting a comparative analysis of electrical fault detection and classification, involving both the recurrent neural network and the decision tree algorithm.

III. Material and Methods

This section presents the methods utilized for the discovery and classification of electrical defects through the application of RNN. In order to establish reliable and secure power networks, the suggested methodology seeks to enable the timely identification of various problem types using real-time fault detection. To assure the inclusion of a diverse range of fault types, a thorough dataset is developed, spanning multiple electrical failure scenarios. The methodology comprises of two main stages: data preparation and training. The combination of a recurrent neural network with a multi-layered feedforward neural network leverages the inherent ability of RNNs to retain sequential information, which is then utilized by the feedforward network to extract relevant features for making predictions. The training process for both components uses the backpropagation technique, which involves iteratively modifying the weights and biases in order to decrease the errors in predictions.

A. System Description and Modelling

Figure 1 shows the test system [24]. The power system configuration comprises four generators 50 MVA operating at 11 kV. Transformers 11/132 kV are strategically placed within the system to facilitate the study of various fault scenarios occurring at the midpoint of the transmission line. The data is sourced from a MATLAB Simulink model, specifically designed to simulate fault analysis in a power system. To gather the data, a basic power grid model is created, incorporating a transmission line between two power systems. The authors utilized the three-phase fault block in MATLAB Simulink to generate faults over the transmission line, simulating all six types of fault conditions. The line voltages and currents at the output side of the power system were measured and recorded under both normal operating conditions and different fault scenarios. Figure 2 shows the SIMULINK model for the test system.

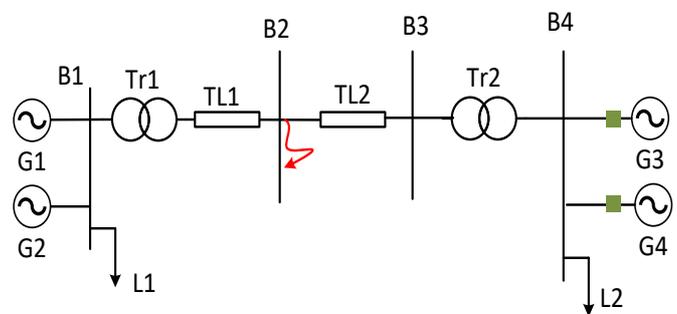


Figure 1. Test System Schematic Diagram.

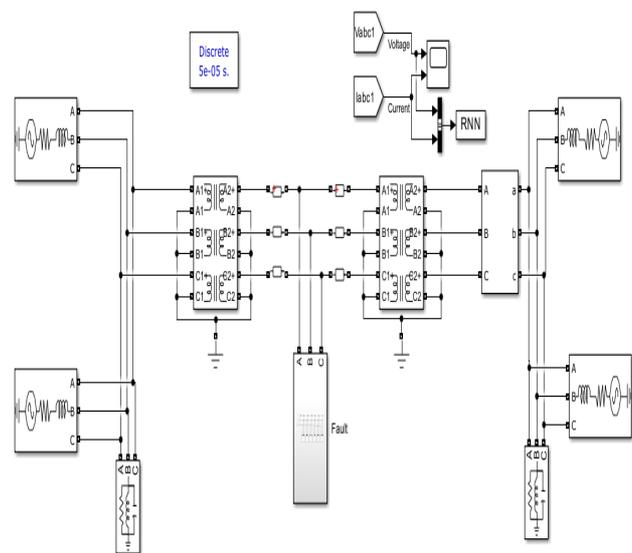


Figure 2. MATLAB Simulink Model for the Test System.

B. Datasets

The system is simulated under both normal operating circumstances and various fault scenarios. Following data collection, each data point is meticulously labeled to indicate the specific fault condition present during the simulation.

These fault labels are assigned based on the type and severity of the simulated fault, allowing for accurate training and evaluation of machine learning models for fault detection and classification. The resulting dataset consists of nearly 12,000 data points, providing a substantial and diverse collection of fault scenarios. The dataset's size and diversity are instrumental in developing robust machine learning models capable of effectively detecting and classifying various fault types.

C. RNN

The Recurrent Neural Network (RNN) is a distinct type of artificial neural network that has been specifically designed to effectively process sequential input. RNNs possess the capability to effectively handle sequential inputs through the utilization of a "hidden state" mechanism, which enables the retention of pertinent information from preceding inputs. This stands in opposition to conventional feedforward neural networks, which operate on inputs in isolation and lack the ability to retain information from previous inputs. RNNs possess the capability to perform this task due to their inherent ability to handle inputs in a non-independent manner. The hidden state of a RNN undergoes modification at each time step through a weighted combination of the current input and the preceding hidden state, with the weights for this combination being acquired through prior learning. This functionality allows the network to generate predictions by leveraging the past input data and effectively capturing the temporal relationships within the dataset. RNNs provide a diverse range of methods that can be effectively employed for addressing time series problems.

Several types of RNNs exist, with the Long Short-Term Memory (LSTM) networks and the Gated Recurrent Unit (GRU) networks being the most prominent examples. Both of these types of RNNs have been shown to be highly effective in detecting and capturing long-term dependencies in sequential data.

The fundamental RNN model, when given an input sequence x , aims to forecast a state s_t at time t . It achieves this by including the previous state s_{t-1} through the utilization of a differentiable function f . The preceding state encompasses not only the data from the preceding time step but can be understood as a compressed representation of all preceding states. The weight parameters in the architecture of RNNs, denoted as U , V , and W , are shared across all layers, as shown in Figure. 3 [26].

$$s_t = f(s_{t-1} * W + x_t * U) \tag{1}$$

$$y_t = f(s_t * V) \tag{2}$$

Where, y_t is the output at time step t , V is the weight matrix that governs the mapping between the hidden state and the output.

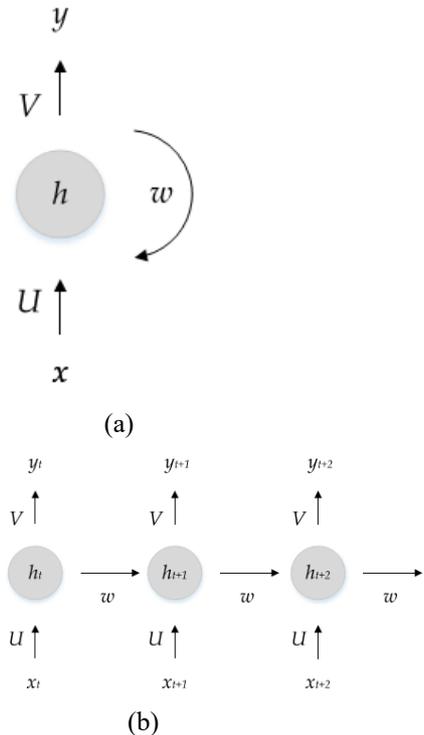


Figure 3. (a) RNN Unit (b) Unfolded RNN Unit with Different Time-Steps State.

RNN models are trained with the objective of minimizing a loss function, which quantifies the discrepancy between the predicted values and the actual observed values. During the training process, the RNN model is initially decomposed into its individual recurrent steps, as depicted in Figure 3(b). Subsequently, the gradient of the loss function with respect to the output state at any given moment is determined. The calculated gradient is propagated in a retrograde manner across the network across multiple time steps to facilitate the completion of the procedure. The subsequent equations denote, in sequence, the recurrent associations and the cumulative gradients of parameters [26].

$$\frac{\partial J}{\partial s_{t-1}} = \frac{\partial J}{\partial s_t} * W \tag{3}$$

Where: $\frac{\partial J}{\partial s_t}$ is represents the gradient of the loss function with respect to the current hidden state at time step t . Similarly, $\frac{\partial J}{\partial s_{t-1}}$ denotes the gradient of the loss function with respect to the previous hidden state at time step $t-1$. This gradient is obtained by backpropagating the gradient through the current hidden state at time step t , utilizing the chain rule of differentiation.

$$\frac{\partial J}{\partial U} = \sum_{t=0}^n \frac{\partial J}{\partial s_t} * x_t \tag{4}$$

Where: x_t represents the current input at time step t , n denotes the length of the input vector. $\frac{\partial J}{\partial U}$ represents the gradient of the loss function with respect to the weight matrix U . This weight

matrix determines the impact of the current input on the current hidden state. The computation of the current hidden state involves summing the product of the gradient with respect to the current hidden state and the current input over all time steps.

$$\frac{\partial J}{\partial W} = \sum_{t=0}^n \frac{\partial J}{\partial s_t} * s_{t-1} \quad (5)$$

Where: $\frac{\partial J}{\partial W}$ represents the partial derivative of the loss function with respect to the weight matrix W. This weight matrix determines the impact of the previous hidden state on the current hidden state. The computation involves summing the product of the gradient with respect to the current hidden state and the previous hidden state across all time steps.

D. Decision Tree (DT)

Decision trees are a common and adaptable machine learning method that may be used for both classification and regression. They are well-known for their openness and the ease with which they may be interpreted, qualities that lend them utility in fields such as data mining and decision support systems. The nodes, branches, and leaves that make up a decision tree are known as "decisions," "conditions," and "outcomes," respectively, while the "leaves" can reflect final decisions or predictions [27]. In order to design a decision tree, one must first recursively partition the data based on informative characteristics. These features can be chosen based on factors such as entropy, Gini impurity, and classification error.

The Gini impurity metric quantifies the likelihood of misclassifying a randomly selected element. Entropy is a metric used to quantify the degree of impurity or disorder within a given dataset. The classification error metric is determined by evaluating the mistake rate, which takes into account the proportion of the majority class.

$$\text{Gini Impurity } (I) = 1 - (p^0 + p^1) \quad (6)$$

$$\text{Entropy } (H) = -(p^0 * \log^2(p^0) + p^1 * \log^2(p^1)) \quad (7)$$

$$\text{Classification Error} = 1 - \max(p^0, p^1) \quad (8)$$

Where p^0 is the probability of class 0, and p^1 is the probability of class 1.

Decision trees employ several approaches to choose which characteristic to split on at each internal node. The concept of information gain is used to quantify the decrease in entropy or impurity that results from dividing a dataset based on a certain attribute. The gain ratio is a metric that takes into account the inherent information of each feature when calculating the information gain. The Gini index is a metric that quantifies the level of impurity within a dataset. It is commonly employed to assess the impurity of subsets resulting from a data partition. Predictions in decision trees are generated through a process of traversing the tree structure, starting from the root node and progressing towards a leaf node. This traversal is guided by the decision path, which is determined by the results of feature tests. The decision-making process involves:

1. Starting at the root node.
2. Evaluating the feature associated with the current node.
3. Moving to the child node corresponding to the outcome of the feature test.

4. Repeating steps 2-3 until a leaf node is reached.
5. Assigning the class label or numeric value associated with the leaf node as the prediction.

Decision trees handle missing values by considering multiple routes when encountering a missing value during the traversal. For categorical features, the tree can handle them directly by splitting based on each category.

Decision trees come in different flavors to accommodate various types of machine learning tasks. In this section, we'll explore two primary types of decision trees: Classification Trees and Regression Trees.

Classification trees are used when the target variable is a categorical variable, and the goal is to assign input instances to one of several predefined classes. At each internal node, the algorithm selects the best feature to split the data based on criteria like information gain, Gini impurity, or entropy. The selected feature's values are used to create child nodes corresponding to each possible outcome. The process continues recursively until leaf nodes are reached.

Each leaf node is associated with a class label, representing the predicted class for instances that follow that decision path [28]. Regression trees are used when the target variable is continuous, and the goal is to predict a numeric value. At each internal node, the algorithm selects the best feature to split the data to minimize variance or another suitable measure. Child nodes are created based on the selected feature's values.

Although there are commonalities between both types of trees, such as the utilization of a recursive splitting process, they exhibit differences in terms of their output types and assessment metrics. Classification trees employ metrics such as information gain and Gini impurity to assess the effectiveness of splits, whereas regression trees utilize measurements such as variance reduction. The evaluation metrics are chosen based on the inherent characteristics of the target variable, with categorical variables being assessed using classification metrics and continuous variables being evaluated using regression metrics.

IV Proposed Methodology

This section presents a comprehensive methodology employed in the manuscript. The primary objective of this study is to develop and evaluate machine learning models for accurate detection and classification of electrical faults. The research follows a rigorous and systematic approach, encompassing data collection, preprocessing, algorithm selection, model training, evaluation, result analysis, and interpretation. Each step of the methodology is carefully outlined, highlighting its significance in achieving robust and effective fault detection and classification outcomes.

A. Data Collection

The foundation of this research lies in the acquisition of a comprehensive and representative dataset. Real-world electrical system parameters, including voltage, current, and corresponding fault labels, were collected using online measurements for the SIMULNK model of the system under study. The dataset's size and diversity were carefully

considered to ensure the models' generalizability and applicability to various fault scenarios.

B. Exploratory Data Analysis (EDA)

EDA involves a comprehensive examination and visualization of the dataset to gain valuable insights and identify patterns that aid in the subsequent stages of the research. To perform EDA, various data visualization techniques were employed, including histograms, pie charts, and correlation matrices. Each visualization technique contributed to a deeper understanding of the dataset's characteristics and facilitated informed decision-making throughout the research.

- Histograms: Histograms provide a clear representation of the distribution of different electrical system parameters, such as voltage and current. By visualizing the data distribution, we could identify potential outliers, assess data symmetry, and gain insights into the overall data structure.
- Pie Charts: Pie charts were utilized to visualize the class distribution of fault labels. This allowed us to assess the balance of classes and identify potential class imbalance issues. A balanced class distribution is essential to ensure that the machine learning models effectively capture patterns from all fault types.
- Correlation Matrices: Correlation matrices provided a comprehensive view of the relationships between different features in the dataset. Understanding feature correlations was critical in identifying multicollinearity and redundant features, guiding the feature selection and engineering process [26]. Pearson's correlation formula is given as

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (9)$$

Where, r is the correlation coefficient, n is the total values number, x_i, y_i are the individual data points in the sample, and \bar{x}, \bar{y} are the sample means of the individual values.

Through the application of these visualization techniques, we gained important insights into the dataset's distribution, relationships between variables, presence of outliers, and potential data quality issues. These insights influenced subsequent data preprocessing decisions, such as handling missing values, outlier treatment, feature selection, and addressing class imbalance.

C. Data Preprocessing

Data preprocessing is crucial for enhancing data quality and model performance. The collected dataset underwent rigorous data cleaning to handle missing values, outliers, and inconsistencies. Feature engineering techniques were employed to select relevant features and engineer new ones to capture essential fault patterns. Furthermore, normalization and scaling were applied to standardize the features and facilitate the convergence of machine learning algorithms.

D. Algorithm Selection

Selecting appropriate algorithms significantly impacts the efficacy of fault detection and classification. For this manuscript, two main algorithms were chosen: the DT and the RNN. The Decision Tree algorithm provides interpretability

and is suitable for binary classification tasks, making it suitable for fault detection. The RNN, with its ability to capture complex patterns, is well-suited for more intricate fault classification scenarios.

E. Model Training

Model training involved feeding the preprocessed data to the selected algorithms. For the RNN, the architecture was carefully tailored, including the number of layers, neurons per layer, and activation functions. During the training process, models adjusted their parameters through backpropagation and optimization algorithms to minimize the loss function. Careful consideration was given to avoid overfitting, and regularization techniques were applied when necessary.

F. Model Evaluation

The performance evaluation of the trained models was a crucial step in gauging their effectiveness. The dataset was split into training and testing sets to ensure unbiased evaluation. Performance metrics such as accuracy, precision, recall, and confusion matrices were employed to quantify the models' fault detection and classification capabilities accurately.

V. Simulation Results

In this study, the effectiveness and robustness of the proposed electrical fault detection and classification methods, utilizing RNNs and DT, are thoroughly assessed through comprehensive simulation experiments. As modern industries rely heavily on the seamless operation of electrical systems, the early detection and accurate categorization of faults play a pivotal role in preventing potential disruptions and ensuring safety.

A. The Electric Fault Detection Dataset

The electric fault detection dataset comprises 12001 records. The attributes are $I_a, I_b, I_c, V_a, V_b, V_c$, and Outputs. The variables I_a, I_b, I_c, V_a, V_b , and V_c are numeric variables. I_a, I_b, I_c , represent line currents of phases A, B, C whereas, V_a, V_b, V_c stand for phase voltages of phases A,B,C. The variable outputs are a categorical variable. It consists of two values, 0 and 1. The value 0 means there is no fault detected, while the value 1 means a fault has been detected.

B. Pre-processing the Electric Fault Detection Data

Before training the model, the data has to be pre-processed. The features are I_a, I_b, I_c, V_a, V_b , and V_c . The target variable is Outputs (s). The dataset has been splitted into the training set and the test set in the ratio of 80:20.

We used a training set to train the machine learning models and the test set to evaluate the performance of the models. There are 9600 samples in the training set and 2401 samples in the test set. We applied standardization to the dataset to keep all of our data in the same range and scale. It also helps our machine learning model train faster.

C. The Electric Fault Classification Dataset

The electric fault classification dataset comprises 7861 records. The attributes are $I_a, I_b, I_c, V_a, V_b, V_c, G, C, B$, and A. The variables I_a, I_b, I_c, V_a, V_b , and V_c are the same as the ones in the electric fault detection dataset. The variables G, C, B, and



A are categorical variables. They consist of two values, 0 and 1. The value 0 means there is no fault detected, while the value 1 means a fault has been detected. A value of 1 in the G variable means there is a fault in the ground. A value of 1 in the C variable means there is a fault in Phase C. A value of 1 in the B variable means there is a fault in Phase B. A value of 1 in the A variable means there is a fault in Phase A. Table I shows how the type of fault is determined according to new values of variables A, B, C, G.

Table I. Fault cases classifications according to A-B-C-G variables.

Fault Case	G	C	B	A
No Fault	0	0	0	0
A-G Fault	1	0	0	1
A-B Fault	0	0	1	1
A-B-G Fault	1	0	1	1
A-B-C Fault	0	1	1	1
A-B-C-G Fault	1	1	1	1

The pie chart in figure 4 shows that there is more data for the 'No Fault' class. The 'Phase B & C' class has less data compared to the other classes. The 'Phase A & Ground', 'Phase A, B & Ground', 'Phase A, B & C', and 'Phase A, B, C & Ground' have similar proportions of data. The dataset has imbalanced classes. Table II shows the total number of records in each class of the electric fault classification dataset.

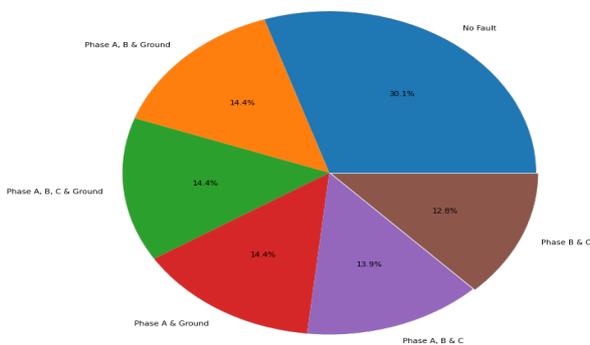


Figure 4. Proportion of Dataset Classification Pie Chart.

Table II: Total number of records in each class of the electric fault classification dataset.

Category	No of samples
No Fault	2365
A-B-G	1134
A-B-C-G	1133
A-G	1129
A-B-C	1096
B-C	1004

D. Exploratory Data Analysis

The histogram depicted in Figure. 5 illustrates that the majority of values for the I_a variable are concentrated within the interval of -100 to 100. The cumulative count does not exceed 200 for each number falling outside of the range. Figure. 8 shows the histogram of the V_a variable, that most of the values fall within the range of -0.1 to 0.1. There are also about 800 counts of

voltages with a value of 0.6. There are about 700 counts of voltages with a value of -0.6. The rest of the values are below the count of 500.

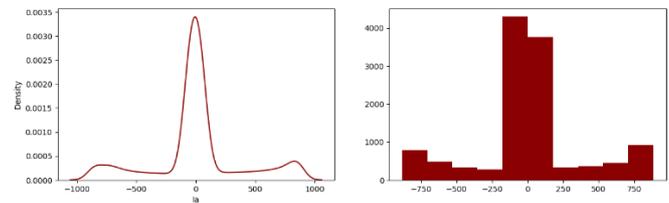


Figure 5. Histogram plot of the I_a variable.

The histogram of the I_b variable in figure 6 shows that most of the values fall within the range of -100 to 100. Each of the values outside that range is not up to 150 in the total count.

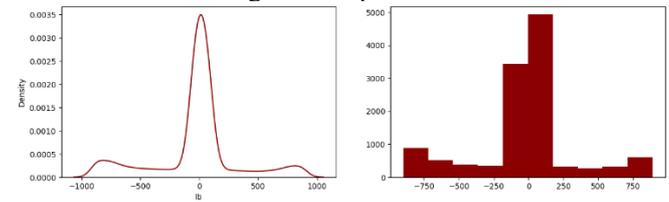


Figure 6. Histogram plot of the I_b variable.

The histogram of the I_c variable in figure 7 shows that most of the values fall within the range of -100 to 100. Each of the values outside that range is not up to 100 in the total count.

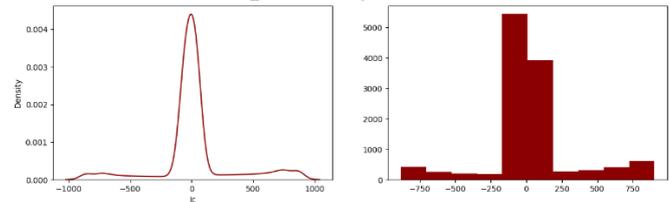


Figure 7. Histogram plot of the I_c variable.

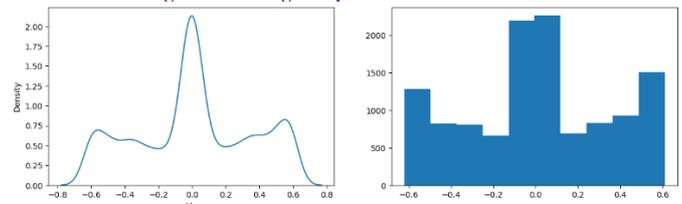


Figure 8: Histogram plot of the V_a variable.

The histogram of the V_b variable in Figure. 9 shows that most of the values fall within the range of -0.05 to 0.05. There are also about 800 counts of voltages with a value of 0.6. There are over 800 counts of voltages with a value of -0.6. The rest of the values are below the count of 600.

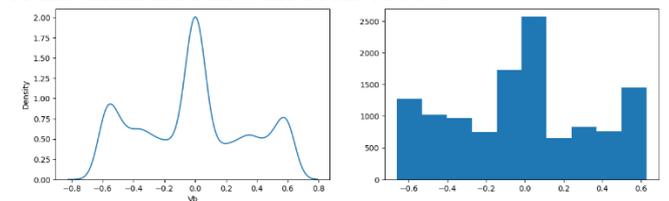


Figure 9. Histogram plot of the V_b variable.

The histogram of the V_c variable in figure 10 shows that most of the values fall within the range of -0.05 to 0.05. There are

also about 750 counts of voltages with a value of 0.6 and -0.6. The rest of the values are below the count of 550.

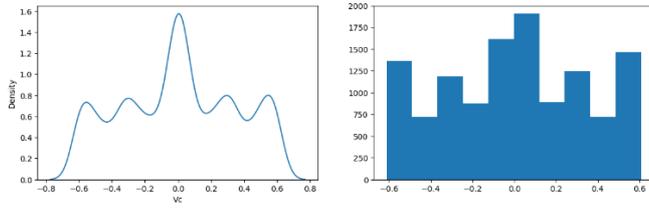


Figure 10. Histogram plot of the Vc variable.

Figure 11 shows the Pearson correlation heatmap for the fault detection dataset. A negative correlation of -0.51 between V_a and V_b means that the value of V_a increases, the value of V_b decreases. There is also a negative correlation of -0.52 between V_b and V_c . There is a negative correlation of -0.53 between I_b and I_c . Figure 12 shows the Pearson correlation heatmap for the fault classification dataset. The correlation heatmap shows a negative correlation of -0.57 between V_b and V_c and a negative correlation of -0.53 between I_b and I_c .

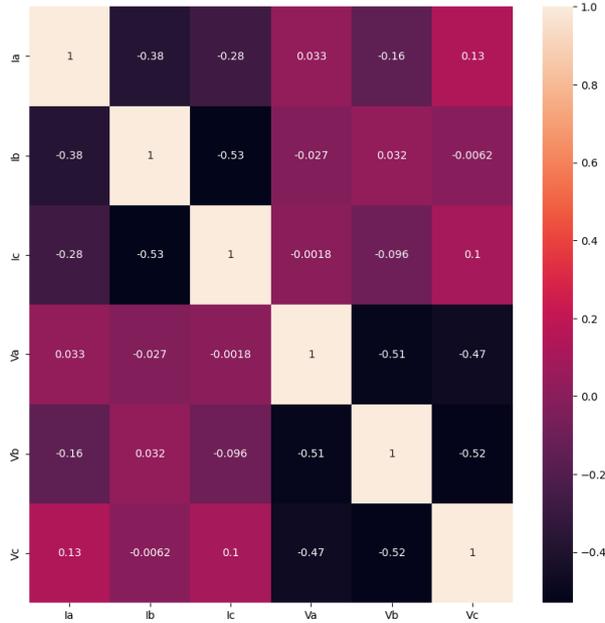


Figure 11. Correlation heatmap of the variables in the electric fault detection dataset.

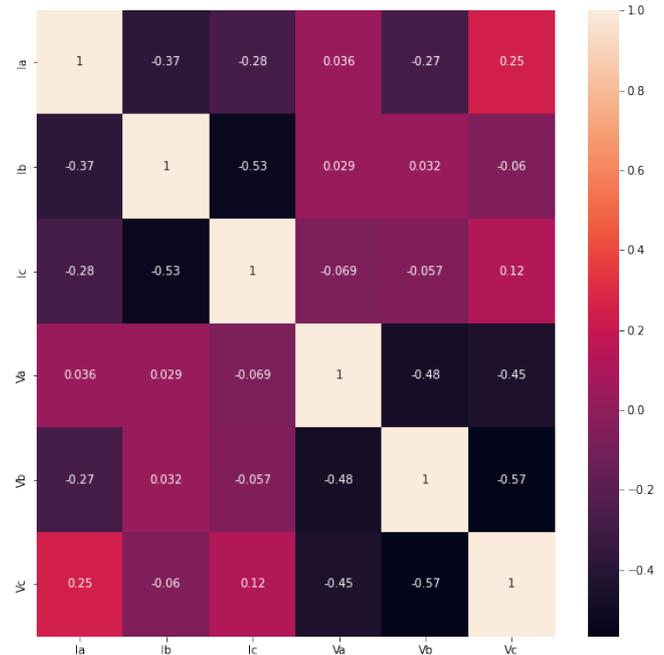


Figure 12. Correlation heatmap of the variables in the electric fault classification dataset.

E. Results for the Electric Fault Detection Models

After training the recurrent neural network for 100 epochs the results obtained are a training accuracy of 99.60%, a training precision of 99.64%, and a training recall of 99.50%. The decision tree algorithm gave a training accuracy of 99.77%, a training precision of 99.55%, and a training recall of 99.95%. The result is summarized in Table III. Figure 13-14 show the confusion matrices for RNN and DT performance in detection scenario. The recurrent neural network classified 1083 cases correctly as having an electrical fault (true positives). It classified 1299 cases correctly as having no electrical fault (true negatives). It wrongly classified 2 cases as having electrical faults (false positives). It wrongly classified 17 cases as having no faults (false negatives). The decision tree model classified 1094 cases correctly as having an electrical fault (true positives). It classified 1298 cases correctly as having no electrical fault (true negatives). It wrongly classified 3 cases as having electrical faults (false positives). It wrongly classified 6 cases as having no faults (false negatives).

Table III: Results for the Electric Fault Detection Models Training.

Performance Metrics	Recurrent Neural Network	Decision Tree Algorithm
Accuracy	99.60%	99.77%
Precision	99.64%	99.55%
Recall	99.50%	99.95%

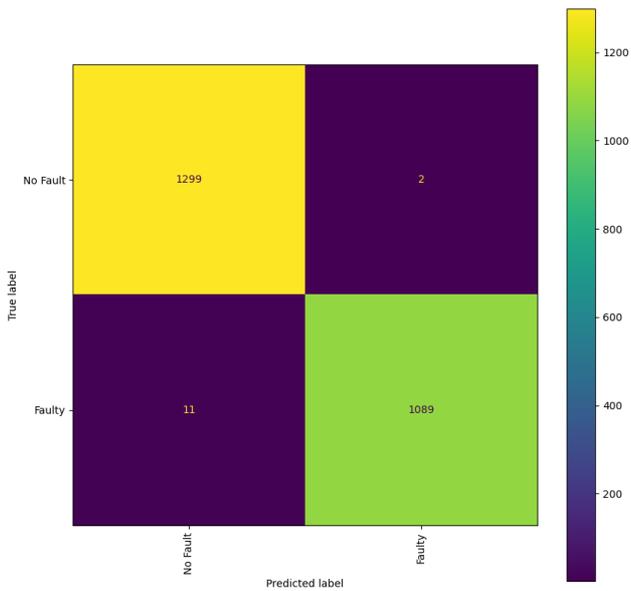


Figure 13. Confusion Matrix for Recurrent Neural Network in Fault Detection.

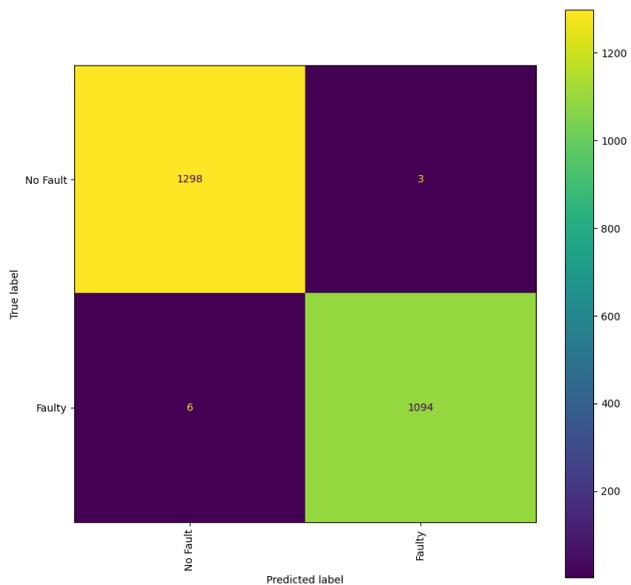


Figure 14. Confusion Matrix for Decision Tree in Fault Detection.

F. Results for the Electric Fault Classification Models

After training the recurrent neural network for 400 epochs the results obtained are a training accuracy of 86.18%, a training precision of 86.40%, and a training recall of 86.05%. The decision tree algorithm gave a training accuracy of 96.93%, a training precision of 97.27%, and a training recall of 96.38%. The result is summarized in Table IV.

Table IV: Results of the Electric Fault Classification Models.

Performance Metrics	Recurrent Neural Network	Decision Tree Algorithm
Accuracy	86.18%	96.93%
Precision	86.40%	97.27%
Recall	86.05%	96.38%

Figure. 15 shows the confusion matrix for RNN in the electrical fault classification. The recurrent neural network classified 473 cases correctly as having No Fault. It classified 226 cases correctly as belonging to A-G class. It classified 52 cases correctly as belonging to A-B-C class. It classified 220 cases correctly as belonging to A-B-G class. It classified 186 cases correctly as belonging to A-B-C-G class. It classified 201 cases correctly as belonging to B-C class. The recurrent neural network misclassified 7 cases actually belonging to A-B-G class as A-G. It has misclassified 40 cases belonging to A-B-C-G class as A-B-C, 1 case belonging to A-B-C-G class as A-B-C, and 167 cases belonging to A-B-C class as A-B-C-G.

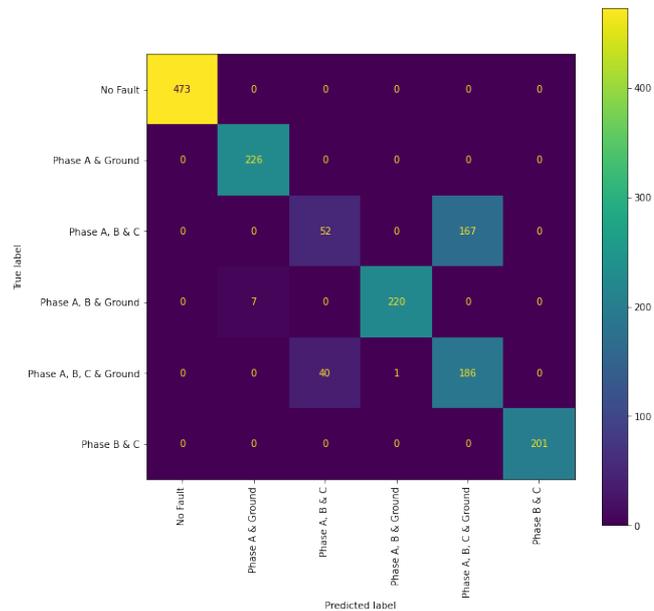


Figure 15. Confusion Matrix for Recurrent Neural Network in Electrical Fault Classification Problem.

Figure. 16 shows the confusion matrix for DT in the electrical fault classification. The decision tree model classified 473 cases correctly as having No Fault. It classified 224 cases correctly as belonging to A & Ground class. It classified 111 cases correctly as belonging to A-B-C class. It classified 226 cases correctly as belonging to A-B-G class. It classified 129 cases correctly as belonging to A-B-C-G class. It classified 200 cases correctly as belonging to B-C class. The decision tree model misclassified 1 case actually belonging to B-C class as No Fault. It misclassified 6 cases actually belonging to A-B-C-G class as No Fault. It misclassified 4 cases actually belonging to A-B-C class as No Fault. It misclassified 1 case actually belonging to A-B-G class as Phase A-G. It misclassified 92 cases actually belonging to A-B-C-G class as A-B-C. It misclassified 104 cases actually belonging to A-B-C class as A-B-C-G. It misclassified 2 cases actually belonging to A-G class as A-B-G.

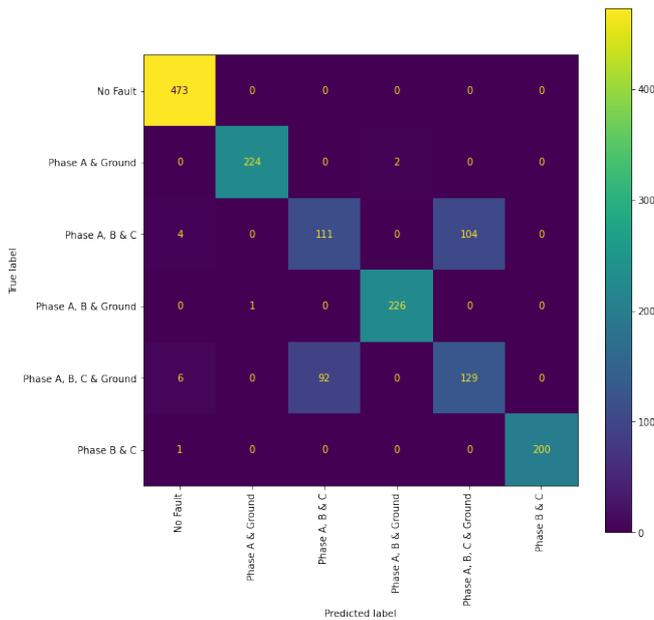


Figure 16. Confusion Matrix for Decision Tree Algorithm in Electrical Fault Classification Problem.

Conclusions

This paper has shed light on the critical role of machine learning techniques in addressing the challenges posed by electrical power system faults detection and classification. The analysis, focusing on the comparison between recurrent neural networks and decision trees, has provided valuable insights into their respective strengths and limitations in the context of fault detection and classification.

The findings of this study underscore the significance of accurate and early detection of electrical faults to mitigate potential risks and prevent disruptions in industrial operations. The results indicate that the recurrent neural network exhibited strong performance in detecting faults when compared to the decision tree model. While the DT algorithm demonstrated slightly superior accuracy in certain scenarios, the RNN exhibited its prowess in generalization, showing a lower susceptibility to overfitting. This characteristic of the RNN becomes particularly vital when dealing with complex, real-world fault scenarios that may not be well-represented by training data alone.

REFERENCES

- [1] J. Hu, Z. Liu, J. Chen, W. Hu, Z. Zhang, Z. Chen, A novel deep learning-based fault diagnosis algorithm for preventing protection malfunction, *International Journal of Electrical Power & Energy Systems* 144 (2023) 108622.
- [2] M. Lashgari, S. M. Shahrtash, Fast online decision tree-based scheme for predicting transient and short-term voltage stability status and determining driving force of instability, *International Journal of Electrical Power & Energy Systems* 137 (2022) 107738.
- [3] H. Farzin, M. Monadi, M. Fotuhi-Firuzabad, M. Savaghebi, A reliability model for overcurrent relays considering harmonic-related malfunctions, *International Journal of Electrical Power & Energy Systems* 131 (2021) 107093.
- [4] K. M. Suliva, Optimization of voltage tolerance curve against voltage sag using cuckoo search algorithm, in 2020 IEEE Region 10 Conference (TENCON), IEEE, 2020, pp. 835–840.
- [5] I. Dobson, D. E. Newman, Cascading blackout overall structure and some implications for sampling and mitigation, *International Journal of Electrical Power & Energy Systems* 86 (2017) 29–32.
- [6] W. Yuan, T. Wang, D. Diallo, A secondary classification fault diagnosis strategy based on pca-svm for cascaded photovoltaic grid-connected inverter, in: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, Vol. 1, IEEE, 2019, pp. 5986–5991.
- [7] G. Devadasu, M. Sushama, A novel multiple fault identification with fast fourier transform analysis, in: *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, IEEE, 2016, pp. 1–5.
- [8] W. Dehina, M. Boumezhaz, F. Kratz, Diagnosis of rotor and stator faults by fast fourier transform and discrete wavelet in induction machine, in: *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, IEEE, 2018, pp. 1–6.
- [9] A. M. Stanisavljević, V. A. Katić, B. P. Popadić, B. P. Dumnić, R. J. Radišić, I. M. Kovačević, Reduced FFT algorithm for network voltage disturbances detection, in: *2016 International Symposium on Industrial Electronics (INDEL)*, IEEE, 2016, pp. 1–6.
- [10] V. A. Katić, A. M. Stanisavljević, Novel voltage dip detection algorithm using harmonics in the dip's transient stage, in: *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2017, pp. 351–356.
- [11] A. M. Stanisavljević, V. A. Katić, R. L. Turović, B. P. Dumnić, B. P. Popadić, Voltage dips detection using kalman filter in a microgrid with high level of distributed generation, in: *2018 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe)*, IEEE, 2018, pp. P–1.
- [12] V. A. Katic, A. M. Stanisavljevic, R. L. Turovic, B. P. Dumnic, B. P. Popadic, Extended kalman filter for voltage dips detection in grid with distributed energy resources., in: *ISGT Europe*, 2018, pp. 1–6.
- [13] U. B. Parikh, B. Das, R. P. Maheshwari, Combined wavelet-svm technique for fault zone detection in a series compensated transmission line, *IEEE Transactions on Power Delivery* 23 (4) (2008) 1789–1794.
- [14] Y. Q. Chen, O. Fink, G. Sansavini, Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction, *IEEE Transactions on Industrial Electronics* 65 (1) (2017) 561–569.
- [15] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, M. Gabbouj, Real-time fault detection and identification for mmc using 1-d convolutional neural networks, *IEEE Transactions on Industrial Electronics* 66 (11) (2018) 8760–8771.
- [16] Abdullah Farid, A., Hatem A. Khater and Selim, G. (2020). "Applying Artificial Intelligence Techniques to Improve Clinical Diagnosis of Alzheimer's Disease". *European Journal of Engineering Science and Technology*, 3(2), 58–79.
- [17] Hatem A. Khater, S. Mesbah, and A. Anwar. "Enhanced navigation system for AUV using mobile application." *International Journal of Engineering Inventions*, Volume 5, Issue 1, PP: 14-19 (2015).
- [18] Abdullah Farid, Hatem A. Khater and Selim, G "Applying Artificial Intelligence Techniques for Prediction of Neurodegenerative Disorders: A Comparative Case-Study on Clinical Tests and Neuroimaging Tests with Alzheimer's Disease", *Proceedings of the 2nd International Conference on Advanced Research in Applied Science and Engineering*, 2020.
- [19] X. Qu, B. Duan, Q. Yin, M. Shen, Y. Yan, Deep convolution neural network based fault detection and identification for modular multilevel converters, in: *2018 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2018, pp. 1–5.
- [19] A. Bagheri, I. Y. Gu, M. H. Bollen, E. Balouji, A robust transform-domain deep convolutional network for voltage dip classification, *IEEE Transactions on Power Delivery* 33 (6) (2018) 2794–2802.
- [20] Y. Dong, H. Cao, G. Xu, C. Yue, X. Ding, A novel method for multiple power quality disturbances classification using a multi-task convolution neural network, in: *2019 4th International Conference on Power and Renewable Energy (ICPRE)*, IEEE, 2019, pp. 274–278.
- [21] Y. Qin, Q. Qian, Y. Wang, J. Zhou, Intermediate distribution alignment and its application into mechanical fault transfer diagnosis, *IEEE Transactions on Industrial Informatics* 18 (10) (2022) 7305–7315.
- [22] O. A. Alimi, K. Ouahada, A. M. Abu-Mahfouz, A review of machine learning approaches to power system security and stability, *IEEE Access* 8 (2020) 113512–113531.

- [23] M. Jamil, S. K. Sharma, R. Singh, Fault detection and classification in electrical power transmission system using artificial neural network, Springer Plus 4 (1) (2015) 1–13.
- [24] L. R. Medsker, L. Jain, Recurrent neural networks, Design and Applications 5 (64-67) (2001) 2.
- [25] C. C. Aggarwal, C. C. Aggarwal, Recurrent neural networks, Neural Networks and Deep Learning: A Textbook (2018) 271–313.
- [26] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, Journal of Applied Science and Technology Trends 2 (01) (2021) 20–28.
- [27] H. Zhou, J. Zhang, Y. Zhou, X. Guo, Y. Ma, A feature selection algorithm of decision tree based on feature weight, Expert Systems with Applications 164 (2021) 113842.
- [28] Q. Liu, Z. Li, Y. Ji, L. Martinez, U. H. Zia, A. Javaid, W. Lu, J. Wang, Forecasting the seasonality and trend of pulmonary tuberculosis in Jiangsu province of China using advanced statistical time-series analyses, Infection and drug resistance (2019) 2311–23.