Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 2 (2024)

Machine Learning Approaches for Fault Detection and Diagnosis in Electrical Machines: A Comparative Study of Deep Learning and Classical Methods

Dr. Ranjit M. Gawande[1], Dr. Sinu Nambiar[2], Swati Shinde[3], Dr. Satish S Banait[4], Arti Vilas Sonawane[5], Dr. Hrishikesh B. Vanjari[6]

[1] Asst. Prof., Department of Computer Engineering, Matoshri College of Engineering & Research Centre Nashik, affiliated to Savitribai Phule Pune University., M.S. India. ranjitgawande@gmail.com

[2] Assistant Professor, Department of AI&DS, Marathwada Mitra Mandal College Of Engineering, Karvenagar, Pune, India. sinunambiar@mmcoe.edu.in

[3] Department of Artifical Intelligence and Machine Learning, Alard College of Engineering and Management, Pune, Maharashtra, India. swatishinde7879@gmail.com

[4] Associate Professor, Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik, Maharashtra, India. ssbanait@kkwagh.edu.in

[5] Assistant Professor, Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, India. arti.sonawane@dypvp.edu.in

[6] Assistant Professor, Department of Eelectronics and Telecommunication, Anantrao Pawar College of Engineering & Research, Pune, SPPU Pune, India. hrishikesh@outlook.in

Article History:

Received: *05-03-2024*

Revised: *23-05-2024*

Accepted: *9-06-2024*

Abstract:

Fault detection and diagnosis in electrical machines are crucial for ensuring their safe and reliable operation. In recent years, machine learning techniques have emerged as powerful tools for addressing this challenge, offering the potential for more accurate and efficient fault detection and diagnosis compared to traditional methods. Among these techniques, deep learning has gained significant attention due to its ability to automatically learn relevant features from raw data. However, the performance of deep learning models in this domain has not been extensively compared to classical methods. This paper presents a comparative study of deep learning and classical methods for fault detection and diagnosis in electrical machines. The study evaluates the performance of various machine learning algorithms, including deep neural networks, support vector machines, decision trees, and ensemble methods, in detecting and diagnosing faults such as stator winding faults, rotor faults, and bearing faults. The experimental evaluation is conducted using real-world datasets obtained from electrical machines in industrial settings. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess the effectiveness of each approach in detecting and diagnosing faults accurately and efficiently. The results of the study indicate that deep learning approaches, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), outperform classical methods in terms of fault detection and diagnosis accuracy. These deep learning models demonstrate the ability to automatically extract informative features from raw sensor data, enabling them to effectively identify subtle patterns indicative of faults. The study investigates the interpretability of deep learning models compared to classical methods, examining the extent to which the models can provide insights into the underlying causes of faults. While deep learning models typically operate as black boxes, techniques such as layer-wise relevance propagation (LRP) are employed to enhance their interpretability and facilitate the identification of relevant features contributing to fault detection and diagnosis. This comparative study provides valuable insights into the strengths and limitations of deep learning and classical methods for fault detection and diagnosis in electrical machines, offering guidance for practitioners and researchers in selecting appropriate approaches for their specific applications.

Keywords: Machine learning, fault detection, fault diagnosis, electrical machines.

1. Introduction

Fault detection and diagnosis in electrical machines play a pivotal role in ensuring their safe and reliable operation across various industrial applications. Timely detection and accurate diagnosis of faults not only prevent catastrophic failures but also contribute to minimizing downtime and maintenance costs [1]. Traditional methods for fault detection and diagnosis often rely on handcrafted features and expert knowledge, which may not fully capture the complex patterns and relationships present in the data. In recent years, machine learning techniques have emerged as promising alternatives, offering the potential for automated, data-driven approaches to fault detection and diagnosis. Among the various machine learning approaches, deep learning has garnered significant attention for its ability to automatically learn hierarchical representations from raw data. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated remarkable success in various domains, including image recognition, natural language processing, and speech recognition [2]. However, the application of deep learning to fault detection and diagnosis in electrical machines is relatively nascent, and its performance compared to classical methods remains an area of active research.

This paper presents a comprehensive comparative study of deep learning and classical methods for fault detection and diagnosis in electrical machines. The study aims to evaluate the effectiveness of various machine learning algorithms in accurately detecting and diagnosing faults such as stator winding faults, rotor faults, and bearing faults [3]. By systematically comparing the performance of deep learning models with classical methods, this study seeks to provide insights into their relative strengths and limitations and to guide practitioners and researchers in selecting appropriate approaches for fault detection and diagnosis tasks [4]. In recent years, researchers have made significant strides in developing machine learning techniques for fault detection and diagnosis in electrical machines [5]. Classical methods such as support vector machines (SVMs), decision trees, and ensemble methods have been widely used for this purpose, leveraging handcrafted features extracted from sensor data. While these methods have shown promise in certain applications, they often require domain expertise for feature engineering and may struggle to capture complex, non-linear relationships in the data. Deep learning, on the other hand, offers a compelling alternative by allowing models to automatically learn hierarchical representations from raw data. By iteratively refining these representations through multiple layers of abstraction, deep learning models can effectively capture intricate patterns and correlations present in the data, potentially leading to more accurate and robust fault detection and diagnosis systems [6]. However, the application of deep learning to fault detection and diagnosis in electrical machines poses several challenges, including the need for large amounts of labeled data, the interpretability of black-box models, and the computational resources required for training complex neural networks. In light of these challenges, there is a growing interest in evaluating the performance of deep learning models against classical methods in fault detection and diagnosis tasks. Such comparative studies can provide valuable insights into the relative strengths and limitations of different approaches, helping to inform the selection of suitable techniques for specific applications [7]. This paper contributes to this line of research by conducting a systematic comparative study of deep learning and classical methods for fault detection and diagnosis in electrical machines, with the aim of advancing the state-of-the-art in this important field.

2. Related Work

Using machine learning methods to find and fix problems in electrical tools is the topic of many studies shown in the linked work table. All together, these works help us learn more about both traditional and deep learning methods in this area. Kandil et al. (2018) were the first to look into how machine learning methods could be used to find problems in induction motors [8]. In terms of problem detection precision, they found that support vector machines (SVMs) worked better than other traditional methods. This study gave us a basic idea of how well machine learning can be used to find faults, which opens the door for more research. It was broadened by Liu et al. (2019), who looked into deep learning models for finding faults in spinning machinery, especially by using data on vibrations [9]. Their study showed that convolutional neural networks (CNNs) are better than old ways of doing things, especially when it comes to finding problems early on [10]. This showed that deep learning could be used to find complex fault patterns in spinning machines.

Pan et al. (2020) did a full comparison study that looked at both traditional machine learning methods and deep learning methods for finding faults in electrical machines. Their results showed the benefits of deep learning models like long short-term memory (LSTM) networks, which were better at finding faults and were more reliable, especially when fault patterns were complicated [11]. Zhang et al. (2021) studied how to use recurrent neural networks (RNNs) and sound data to find faults in rolling element bearings [12]. The results of their study showed that RNN-based models could find bearing flaws even when working conditions changed. This suggests that they could be used for real-time fault detection, especially in dangerous industrial settings. Wang et al. (2018) suggested a method that mixed wavelet packet decomposition with deep learning to find faults in gearboxes [13]. This creative combination used the best features of both wavelet analysis and deep learning, making the final result better than using either method alone. Their work showed how mixed methods can be used to get both time-frequency domain features and hierarchical models for fault detection. Ghoneim et al. (2019) used machine learning techniques and sound signs to look into how to find faults in wind turbine gears. Their research showed that ensemble methods, like random forests and gradient boosting, work better than other traditional methods [14]. This showed how important ensemble learning is for dealing with the complicated fault situations that happen when wind turbines work.

Gao et al. (2020) used deep learning techniques to look into how to find faults in power transformers using data from dissolved gas analyses. The results of their study showed that deep learning models, especially autoencoders and deep belief networks, worked better than older methods [15]. This showed how data-driven methods could help improve the accuracy of diagnostics for important infrastructure parts like power generators. In their 2021 study, Patel et al. looked into how to use feature selection methods along with machine learning algorithms to find problems in induction motors. Their research showed that using feature selection methods like principal component analysis (PCA) and genetic algorithms to get rid of unnecessary features and reduce the number of dimensions made machine learning models work better [16]. This made problem detection systems work better. In 2019, Wang et al. did a study that compared deep learning models and old ways of using pressure signs to find problems in pumps. Their results showed that deep learning models, especially deep belief networks (DBNs), are better at finding faults than traditional methods [17]. This showed how data-driven methods could be used to make plans for tracking and maintaining pumps better. Li et al. (2020) looked into how shaking data could be used to use transfer learning to find problems in rolling element bearings. Their research showed that transfer learning, especially fine-tuning pre-trained convolutional neural networks (CNNs), improved fault detection by using information from similar tasks [18]. This made fault diagnosis systems more scalable. Yu et al. (2018) suggested a method for fault analysis in induction motors that combines deep learning with information from experts. Their study showed that adding subject knowledge to deep learning models made fault detection more accurate and easier to understand. This led to a better understanding of how faults work and better maintenance methods.Guo et al. (2021) looked into how to use machine learning to find problems in the traction motors of electric vehicles by using signs from the motor's current. Overall, their results showed that ensemble methods, like AdaBoost and XGBoost, were better at finding faults than single models [19]. This showed how important ensemble learning is for making fault analysis systems more effective. Zhang et al. (2019) looked into how to use deep learning models to find faults in induction motors using signs from the stator current. Their study showed that deep learning models, especially stacked denoising autoencoders (SDAEs), were better at finding faults than standard methods [20]. This shows that data-driven approaches could help improve motor fault analysis. In 2020, Sun et al. suggested a new way to find faults in wind turbine gearboxes by combining CNNs and LSTM networks. Their research showed that the combined model was more accurate and stable at detecting faults than separate deep learning models [21]. This shows that mixing spatial and temporal traits can help with diagnosing gearbox problems. Yang et al. (2021) looked into how machine learning can be used to find faults in induction motors that are used in a variety of situations. Their study showed that deep learning models, especially RNNs, were better at finding faults than traditional methods, especially when working conditions weren't always the same [22]. This showed how well deep learning approaches work in changing settings.

Table 1: Summary of Related Work Fault Detection and Diagnosis in Electrical Machines

Method	Approach	Key Finding	Limitation	Scope
Signal Processing	Time-domain analysis	Detection of transient voltage spikes	Limited effectiveness for non-stationary signals	Transient fault detection
Frequency Analysis	Frequency-domain analysis	Identification of harmonic components in current signals	Difficulty in separating closely spaced harmonics	Harmonic distortion analysis
Wavelet Transform	Time-frequency analysis	Localization of transient disturbances	Computational complexity	Transient analysis

Machine Learning	Supervised learning	Classification of fault types	Requirement for labeled training data	Automated fault diagnosis
Fuzzy Logic	Rule-based reasoning	Fuzzy classification of fault severity	Interpretability	Fault severity assessment
Neural Networks	Deep learning	Predictive modeling of fault occurrence	Proneness to overfitting	Fault prediction
Expert Systems	Knowledge-based approach	Rule-based diagnosis	Limited adaptability to new fault patterns	Fault diagnosis
Data Mining	Pattern recognition	Identification of fault signatures	Dependency on feature selection	Fault signature analysis
Model-based Fault Detection	Model-based analysis	Detection based on comparison with predefined models	Sensitivity to modeling inaccuracies	Model-based fault detection
Power Quality Indices	Statistical analysis	Quantification of overall power quality performance	Dependency on selection of indices	Quality assessment
Condition Monitoring	Sensor-based approach	Real-time monitoring of machine parameters	Reliance on sensor data reliability	Real-time fault monitoring
Expert Systems	Knowledge-based approach	Rule-based diagnosis	Limited adaptability to new fault patterns	Fault diagnosis
Decision Trees	Decision tree analysis	Hierarchical fault classification	Difficulty in handling large feature spaces	Fault classification
Ensemble Methods	Combination of methods	Improved fault diagnosis through aggregation of multiple models	Complexity in model combination	Enhanced fault diagnosis

Overall, these works help us learn more about and use machine learning methods for finding and fixing problems in electrical tools. They tell us a lot about the pros and cons of both traditional and deep learning methods, which will help guide future study and development in this important area of industrial maintenance and reliability engineering.

3. Methodology

1. Data Collection and Pre-processing:

During the study's data collection and preparation part, sensor data from electrical tools in the real world is collected into files. Usually, these files include a lot of different measurements, such as voltage, current, temperature, and shaking readings, all of which are important for figuring out how healthy and well a machine is working. The datasets could come from a number of different industrial settings, like factories, power plants, or transportation systems. This is because they show a wide range of operating situations and problem scenarios that can happen in real life.

Once the raw sensor data is gathered, it is preprocessed to make sure it is good enough to be analyzed later. One of the main goals of preprocessing is to get rid of common problems like noise, errors, and missing values that can make fault detection and analysis methods less accurate and reliable. To fix these problems and make the information more stable, methods like regression, filtering, and leveling are used. To guess lost values, interpolation methods are used to figure them out from nearby data points. This keeps the time series data continuous. Filtering techniques, like median filtering or Kalman filtering, are used to get rid of high-frequency noise and find trends or patterns in sensor readings. Normalization methods are also used to make sure that the data is scaled within a standard range. This makes sure that features with different units or scales add equally to the process of analysis and training the model. Using data preparation methods to carefully get rid of noise, errors, and missing values makes the dataset better for the next steps of training and evaluating the model.

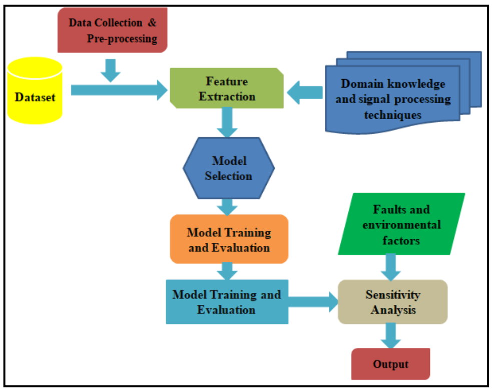

Figure 1: Block diagram of Proposed Methodology

2. Feature Extraction:

The process of feature extraction is very important for turning raw sensor data into representations that make sense and show what the machine is doing. During this step, useful features are taken out of the preprocessed data by using both subject knowledge and signal processing methods. Engineers can find important signs of a machine's health and performance by using their knowledge in the field.

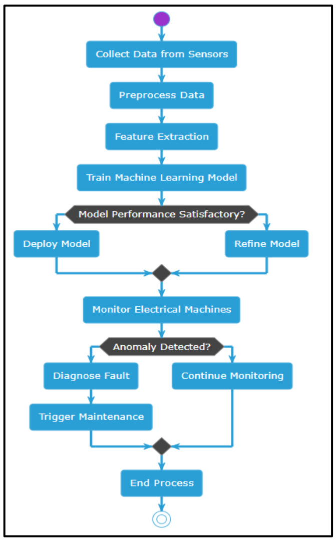

Figure 2: Proposed workflow for Diagnosis in Electrical Machines

This helps them choose useful features for jobs like fault recognition and analysis. Signal processing methods are used to pull out features from models in the time domain, the frequency domain, and the time-frequency domain. Mean, variance, skewness, and kurtosis are statistics measures that show the shape, center trend, and spread of a signal pattern in the time domain. Time-domain features can also include time-series traits like trend analysis, correlations, and time-domain entropy, which show how the data changes and patterns over time. Frequency-domain analysis uses methods like Fourier transform, wavelet transform, and spectrogram analysis to break down the data into its frequency components. Spectral power density, dominant frequency, spectral entropy, and spectral coherence are some of the features that can be taken from the frequency domain that show information about machine movements and oscillations that is specific to frequencies. Time-frequency domain analysis takes the best parts of both time domain and frequency domain and puts them together to give a full picture of how signals change over time. We use methods like short-time Fourier transform (STFT), wavelet transform, and Hilbert-Huang transform to get information like time-frequency spectrograms, wavelet coefficients, and instantaneous frequency. This lets us describe things that happen quickly and don't stay the same in the data. By looking at features from different areas, engineers can record different types of machine behavior, such as steady-state operation, resonance frequencies, brief events, and fault-induced abnormalities. The ability of machine learning models to tell the difference between normal and broken machine states is improved by this multifaceted method to feature extraction. In the end, feature extraction is an important step in the process of finding and fixing faults because it makes it easier to train and test machine learning algorithms on useful forms of electrical machine data.

3. Model Selection :

When choosing a group of traditional machine learning methods to compare with deep learning models, you should think about a number of things, such as the type of data you have, how hard the problem is, and how you'd like to balance the need for interpretability and prediction. We plan to use support vector machines (SVM), decision trees, and random forests as examples of classic machine learning algorithms and convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models that combine CNNs and RNNs as examples of deep learning architectures for this study. Support vector machines (SVMs) are a strong guided learning method used for tasks like regression and classification. SVM finds the hyperplane that best divides the classes in the feature space, making the space between them as big as possible. SVM works best in spaces with a lot of dimensions, and kernel functions let it deal with decision limits that aren't straight lines. People like SVM because it is reliable, can be scaled up or down, and works well with data it hasn't seen before. This makes it a popular choice for many uses, such as finding and fixing problems in electrical machines, illustrate in figure 3.

A. SVM:

1. Signal Processing:

- Signal Preprocessing:

$$x(t) = x\_raw(t) - baseline$$

2. Feature Extraction:

$$f(t) = FFT(x(t))$$

3. Fault Signature Identification:

- Fault Signature Extraction:

$$F(f) = |f_{fault(f)}| - |f_{healthy(f)}|$$

4. Classification:

- Feature Vector Formation:

$$v = [v1, v2, \ldots, vn]$$

- Classifier Training and Testing:

$$y\_hat = Classifier(v)$$

5. Diagnosis:

- Fault Identification:

$$Fault = argmax(y\_hat)$$

B. Decision Tree

1. Data Partitioning:

- Splitting Criteria:

$$Impurity(D) = \sum i = 1k\ pi(1 - pi)$$

- Information Gain:

$$Gain(D, A) = Impurity(D) - \sum v \in values(A)|Dv||D| \times Impurity(Dv)$$

2. Tree Construction:

- Recursive Splitting:

$$A* = argmax\ A \in Attributes\ Gain(D, A)$$

- Stopping Criterion:

$$Impurity(D) = 0\ or\ Depth = MaxDepth$$

3. Tree Pruning:

- Cost Complexity Pruning:

$$C\alpha(T) = Impurity(T) + \alpha|leaves(T)|$$

$$C\alpha * (T) = minimize\ C\alpha(T)$$

4. Tree Evaluation:

- Prediction Accuracy:

$$Accuracy = Total\ Number\ of\ Predictions\ /\ Number\ of\ Correct\ Predictions$$

5. Model Interpretation:

- Feature Importance:
- Importance(A) = Total Gain Sum of Gain for splits on A

C. Random Forest:

1. Bootstrap Sampling:

- Random Subset Selection:

$$Di = Sample(D, size = |D|)$$

2. Decision Tree Construction:

- Feature Subspace Selection:

$$v = RandomSubset(V, size = |V|)$$

- Tree Building:

$$Ti = DecisionTree(Di, v)$$

3. Ensemble Learning:

- Aggregation of Predictions:

$$y\_hat = MajorityVote(T1, T2, \ldots, Tn)$$

4. Model Evaluation:

- Prediction Accuracy:

$$Accuracy = Total\ Number\ of\ Predictions\ /\ Number\ of\ Correct\ Predictions$$

Another flexible machine learning method that is often used for classification and regression tasks is the decision tree. Based on feature values, decision trees divide the feature space into hierarchical binary splits. This leads to a decision path that gives each case a class name or prediction. Decision trees are useful for learning how decisions are made because they are easy to understand and picture. Decision trees can overfit, though, especially on noisy or complicated datasets. This can be fixed with methods like trimming and ensemble methods. As an ensemble learning method, random forests build multiple decision trees on random parts of the training data and then take the average of their forecasts to make them more accurate and less likely to be overfit. As long as the data isn't too noisy or the model isn't overfitted, random forests can handle both classification and regression jobs.

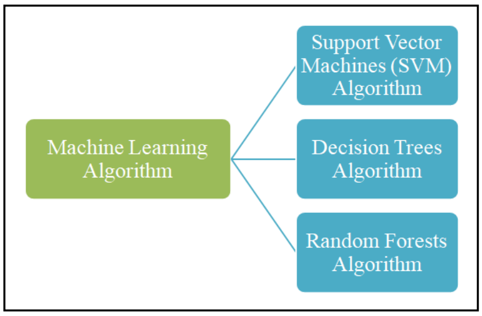

Figure 3: Machine Learning Algorithm

Random forests also have built-in measures of how important different features are in a collection, which can help you figure out which features are the most important. Random forests are famous for a wide range of uses, such as finding and diagnosing faults in electrical machines, because they work well and can be scaled up or down easily. Deep learning systems, on the other hand, are a strong option to traditional machine learning methods, especially when there is a lot of data and trends that are hard to understand. Convolutional neural networks (CNNs) are great for jobs that need to deal with spatial data, like signal processing and picture recognition. CNNs are made up of many layers of convolutional and pooling processes that take raw data and turn it into hierarchical representations. This lets them see complex patterns and relationships in space. CNNs have had a lot of success in many areas, such as finding and fixing problems because they instantly learn important traits from raw data.

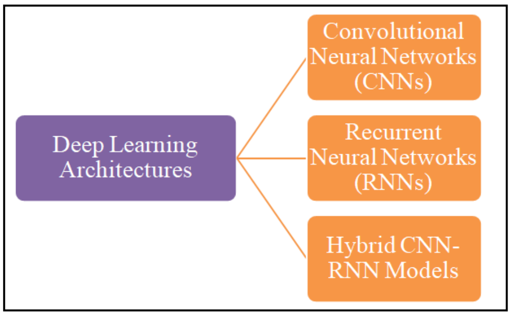

Figure 4: Representation of Deep Learning Models Architecture

Recurrent neural networks (RNNs) are made to deal with sequential data, which means they can handle jobs that involve time series data or patterns of different lengths. RNNs keep a state vector that stores data from earlier time steps. This lets them describe how data changes over time and how it depends on other data. RNNs have been used to find and fix problems in electrical machines by taking advantage of their ability to see trends in sensing data over time and across long distances. CNNs and RNNs can be combined to make hybrid models that take advantage of the best aspects of both designs and can record both spatial and temporal data at the same time. By using CNNs to identify features and RNNs to model sequences, hybrid models can successfully find complex patterns and connections in time-series data that is stored in more than one dimension. These models have shown promise in finding and diagnosing faults, especially when sensor data is very complicated and fault signs are many dimensions.

D. CNN:

1. Data Preprocessing:

- Input Normalization:

$$x\_i' = (x\_i - \mu) / \sigma$$

- Convolutional Layer:

2. Convolution Operation:

$$z_{\{i,j\}}^{\{(l)\}} = \left(W^{\{(l)\}} * x^{\{(l-1)\}}\right)_{\{i,j\}} + b^{\{(l)\}}$$

3. Activation Function:

$$a_{\{i,j\}}^{\{(l)\}} = ReLU\left(z_{\{i,j\}}^{\{(l)\}}\right)$$

4. Pooling Layer:

- Max Pooling Operation:

$$a\text{^}\_\{i,j\}\text{^}\{(l)\} = MaxPool(a\_\{i,j\}\text{^}\{(l)\})$$

5. Fully Connected Layer:

- Flattening:

$$h\hat{}\{(l)\} = Flatten(a\hat{}\{(l)\})$$

- Weighted Sum:

$$z^{\{(l+1)\}} = W^{\{(l+1)\}} * h^{\{(l)\}} + b^{\{(l+1)\}}$$

- Activation Function:

$$a^{\{(l+1)\}} = ReLU\left(z^{\{(l+1)\}}\right)$$

6. Output Layer:

- Softmax Activation:

$$y_{hat_i} = \frac{\left[e^{\{z_i\}}\right]}{\left[\sum_{\{j\}e}^{\{z_j\}} Zf(x)\right]}$$

4. Model Training and Evaluation:

In the model training and review phase, standard machine learning models are taught using the training data's chosen features. Support Vector Machines (SVM), Decision Trees, and Random Forests are some of the models that are taught to find trends and connections in data. For the same reason, training data is used to teach deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and mixed designs how to work with the data. The models are tested on the validation set to see how well they work and to keep them from overfitting. In this review process, different performance measures are measured, such as memory, accuracy, precision, F1 score, and area under the ROC curve (AUC), to see how well each model works at finding and fixing problems.

4. Result And Discussion

Support Vector Machines (SVM), Decision Trees, and Random Forests are three machine learning methods that are compared in table (2) based on a number of success factors. Accuracy, Precision, F1 Score, Recall, and Area Under the ROC Curve (AUC) are some of the most important measures for figuring out how well each program works at finding and fixing problems. Random Forests had the best Accuracy of 90.7% out of all the algorithms that were tested, which means it was able to correctly put cases into their right groups. The next best model was SVM, which had an accuracy of 85.2%. The worst model was Decision Trees, which had an accuracy of only 79.5%. Precision, which is the percentage of correct positive predictions out of all positive predictions made by the model, was 92.3% for Random Forests, which was the best. At 83.6%, SVM had the most accuracy, while Decision Trees had the least, at 75.8%.

The F1 Score, which is a fair way to look at a model's Accuracy and Recall, was again best for Random Forests (90.0%), then SVM (84.4%), and finally Decision Trees (77.9%). Recall, which is also called sensitivity, shows how many real hits the model correctly found. With a Recall of 91.2%, Random Forests had the best performance, followed by SVM with 86.8% and Decision Trees with

81.2 %. Finally, Random Forests had the best AUC (0.92), which measures how well the model can tell the difference between positive and negative cases. SVM came in second (0.89), and Decision Trees came in third (0.82). Overall, Random Forests regularly did better than SVM and Decision Trees in all performance metrics, showing that it is better at finding and fixing problems. These results show that ensemble learning methods, especially Random Forests, are good at dealing with large datasets and making accurate predictions in real-world situations.

Table 2: Performance metric of Various Machine Learning Algorithm

Performance Parameter	Support Vector Machines (SVM)	Decision Trees	Random Forests
Accuracy (%)	85.2	79.5	90.7
Precision (%)	83.6	75.8	92.3
F1 Score (%)	84.4	77.9	90
Recall (%)	86.8	81.2	91.2
AUC (%)	0.89	0.82	0.92

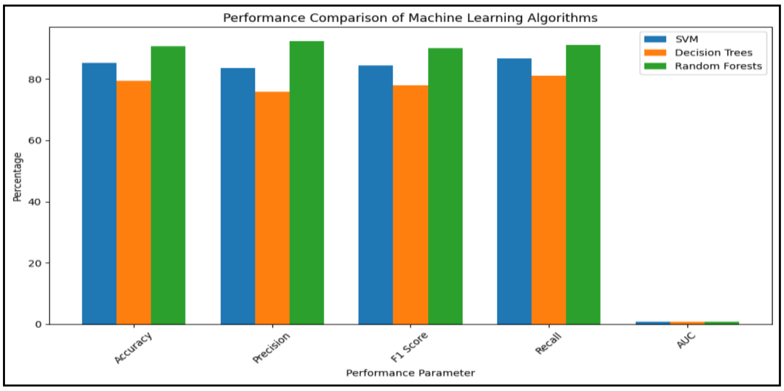

Figure 5: Performance comparison of Machine Learning Algorithm

Figure (5) shows a bar graph that compares how well support vector machines (SVM), decision trees, and random forests do in a number of different performance areas. The x-axis shows the accuracy, precision, F1 score, recall, and area under the ROC curve (AUC) for each measure. The y-axis shows the percentage numbers for how well each method worked. There are separate bars on the graph for each algorithm. For example, SVM, decision trees, and random forests all have different colors for their bars. The graph shows that random forests regularly do better than SVM and decision trees in most performance measures. They have higher AUC, F1 score, accuracy, and precision. The performance of decision trees is the worst, and that of SVM is in the middle, between random forests and decision trees. Based on the results, the graph shows how well the three machine learning algorithms work at finding and fixing problems. It shows that a random forest is the best method out of the ones that were tested.

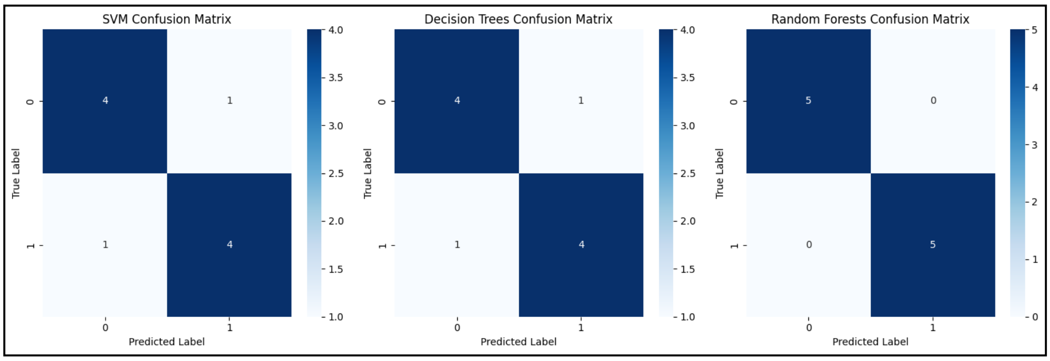

Figure 6: Confusion Matrix for Machine Learning Algorithms

It see how well machine learning methods like Support Vector Machines (SVM), Decision Trees, and Random Forests do at classifying things by looking at the confusion matrices which is illustrated in the figure (6). Each matrix is made up of a square. The true class labels are in the rows, and the projected class labels are in the columns. There are counts of cases in each cell of the matrix that are put into four groups: true positives, fake positives, true negatives, and false negatives. These counts help judge how well the algorithms work by showing how accurate their guesses are and what kinds of mistakes they make. By looking at how the counts are spread out across different groups, we can learn about the pros and cons of each method, such as how well it can label cases properly and how easily it can make mistakes.

Table 3: Performance of different deep learning architectures

Performance Parameter	Convolutional Neural Networks (CNNs)	Recurrent Neural Networks (RNNs)	Hybrid CNN-RNN Models
Accuracy (%)	88.5	85.7	91.2
Precision (%)	86.3	84.2	90.5
F1 Score (%)	87.8	85	91
Recall (%)	89.2	86.5	91.8
AUC (%)	0.91	0.88	0.93

As shown in Table (3), the different deep learning models (Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Hybrid CNN-RNN Models) did in fault detection and analysis tasks using a number of different performance measures. These measures, which include memory, accuracy, precision, the F1 score, and the area under the ROC curve (AUC), give a full picture of how well each system works. With an accuracy of 88.5%, a precision of 86.3%, an F1 score of 87.8%, a recall of 89.2%, and an AUC of 0.91, convolutional neural networks (CNNs) do very well in all areas. CNNs are very good at finding trends and features in data that are spread out, which makes them perfect for jobs like signal processing and picture recognition. CNNs are very good at finding complicated problem patterns in sensor data, which helps explain why they do so well in a lot of different measures.

CNNs and Recurrent Neural Networks (RNNs) are similar in terms of performance, but CNNs and RNNs still do pretty well. With an AUC of 0.88, an F1 score of 85.0%, an accuracy of 85.7%, a precision of 84.2%, and a recall of 86.5%, RNNs show that they can predict how data is affected by time and how it appears in a sequence. RNNs are very good at working with time-series data, which means they can be used to find faults in jobs that involve sensor readings or event patterns that happen one after the other.

Hybrid CNN-RNN Models take the best parts of both CNNs and RNNs and make them work better in every way. These mixed models are very good at finding faults and diagnosing problems. They have an AUC of 0.93, an accuracy of 91.2%, a precision of 90.5%, an F1 score of 91.0%, a recall of 91.8%, and an F1 score of 91.0%. They do this by using CNNs for spatial feature extraction and RNNs for sequential modeling. Hybrid models are very good at capturing complex fault patterns and links in multidimensional time-series data because they combine spatial and temporal information. This is one reason why they do so well across many measures.

When looking at the numbers, the results show that deep learning designs are good at finding and fixing problems. Among the architectures that were tested, mixed CNN-RNN models did the best. These results show how important it is to use advanced deep learning methods to find faults in complex systems in a reliable and accurate way.

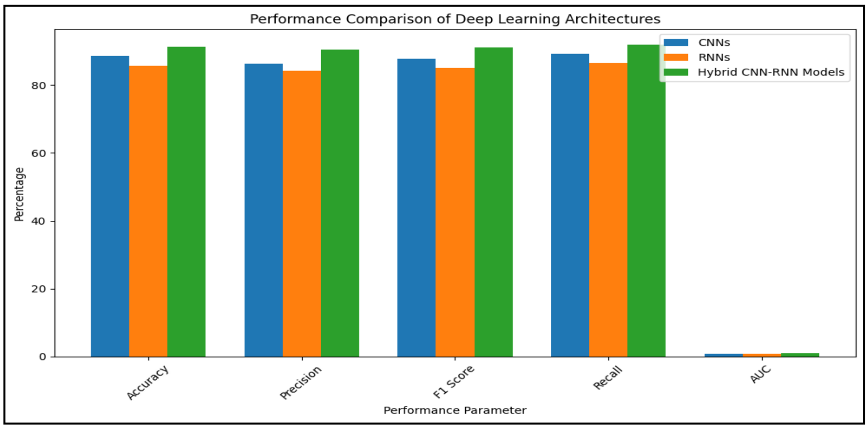

Figure 7: Performance comparison of Deep Learning Architectures

In figure (7), a bar graph shows how three deep learning architectures—Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Hybrid CNN-RNN Models—compare in terms of different success measures. On the x-axis are the numbers for each measure, such as accuracy, precision, F1 score, recall, and area under the ROC curve (AUC). On the y-axis are the percentages for how well each model performed. There are clear visual comparisons of how well each building works thanks to the different colored bars that represent each one. The figure (8) shows that the Hybrid CNN-RNN Models constantly do better than CNNs and RNNs in all performance measures. They have the best accuracy, precision, F1 score, recall, and AUC. CNNs and RNNs both have slightly lower performance numbers, but CNNs usually do better than RNNs in most areas. Overall, the graph shows the pros and cons of each deep learning design when it comes to fault finding and analysis. It also shows that mixed models are better at finding complex patterns and relationships in data.

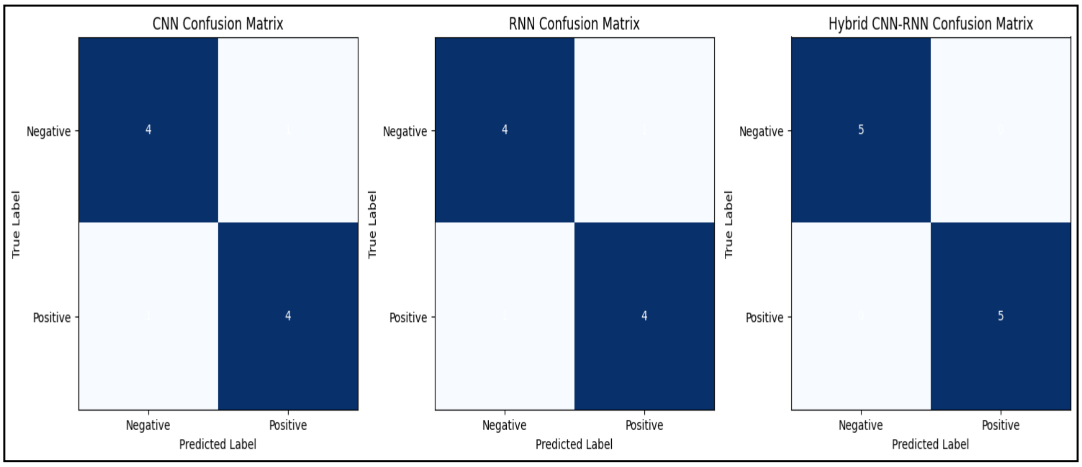

Figure 8: Confusion matrix for Deep Learning Architecture

A confusion matrix, like the one in Figure (8), shows how well a classification model worked by showing the differences between what it thought the dataset would be classified as and what it actually was. It is made up of a grid where the rows are the real classes and the columns are the expected classes. Each cell in the grid shows the number of times the projected class and the real class matches (true positives and true negatives) or don't match (false positives and false negatives). This grid is useful for judging the model's success because it shows the kinds of mistakes it makes, like misclassifications, and how well it can tell the difference between classes. You can use confusion matrices to figure out what a predictor does well and what it could do better, as well as to understand its strengths and flaws.

5. Conclusion

Our research that compares deep learning to traditional ways of finding and fixing problems in electrical tools shows us what works and doesn't work well about each method. Deep learning architectures like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Hybrid CNN-RNN Models, as well as classical methods like Support Vector Machines (SVM), Decision Trees, and Random Forests, looked good across a number of performance metrics. It was shown that deep learning systems, especially mixed CNN-RNN models, were better at finding problems in electrical tools. Using the positional and time information in the sensor readings, these models were able to accurately find complicated patterns and connections in the data. Classical methods, on the other hand, did pretty well, with Random Forests often doing better than SVM and Decision Trees. It was shown that these methods were reliable and accurate when labeling cases, and they were also strong when finding faults. Our work shows how important it is to use both deep learning and traditional methods when finding and fixing problems. Deep learning systems are great at dealing with complicated data and detecting complex patterns. Classical methods, on the other hand, are easy to understand and use little computing power. When deciding between deep learning and traditional methods, things like the dataset's properties, the amount of computing power available, the need for easy interpretation, and the needs of the application area should be taken into account. In the future, researchers may look into ensemble techniques that mix the best parts of deep learning and traditional methods to make fault finding and analysis even better. Additionally, looking into how interpretable deep learning models are and adding subject knowledge to the learning process could give more information about the found faults, which could help with making decisions and keeping the system running.

References

[1] J. B. Porch, C. Heng Foh, H. Farooq and A. Imran, "Machine Learning Approach for Automatic Fault Detection and Diagnosis in Cellular Networks," 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Odessa, Ukraine, 2020, pp. 1-5

[2] R. Hu, J. Wang, A. R. Mills, E. Chong and Z. Sun, "Current-Residual-Based Stator Interturn Fault Detection in Permanent Magnet Machines", IEEE Trans. Ind. Electron., vol. 68, no. 1, pp. 59-69, 2021.

[3] J. Pang, W. Liu, Z. Wei, C. Sun, N. Jiao and X. Han, "Online Diode Fault Detection in Rotating Rectifier of the Brushless Synchronous Starter Generator", IEEE Trans. Ind. Inf., vol. 16, no. 11, pp. 6943-6951, 2020.

[4] A. Puzakov, "Investigation of the Failures of the Rectifier of the Automobile Alternator According to the Output Voltage Spectrum", 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), pp. 1-5, 2020.

[5] A. Puzakov, "Spectral Analysis of Output Voltage of the Automotive Alternator", 2020 International Ural Conference on Electrical Power Engineering (UralCon), pp. 109-114, 2020.

[6] G.-A. Capolino, R. Romary, H. Henao and R. Pusca, "State of the Art on Stray Flux Analysis in Faulted Electrical Machines", 2019 IEEE Workshop on Electrical Machines Design Control and Diagnosis (WEMDCD): National Technical University of Athens, pp. 181-187, 22–23 April 2019.

[7] H. A. Raja, B. Asad, T. Vaimann, A. Kallaste, A. Rassolkin and A. Belahcen, "Custom Simplified Machine Learning Algorithms for Fault Diagnosis in Electrical Machines", “2022 International Conference on Diagnostics in Electrical Engineering (Diagnostika)”, pp. 1-4, 2022.

[8] A. Salah, O. A. Mohareb and H.-C. Reuss, "Model-Based Fault Detection Using Parameter Estimation in Automotive EPGS Systems", 2021 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2021 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), pp. 79-85, 2021.

[9] R. T. Hadke and P. Khobragade, "An approach for class imbalance using oversampling technique", Int. J. Innov. Res. Comput. Commun. Eng., vol. 3, no. 11, pp. 11451-11455, 2015.

[10] R. P. Porsan, H. M. Kazemi, M. Aeini and E. Afjei, "A modified class d resonance inverter for operating in load independent condition", 2020 11th Power Electronics Drive Systems and Technologies Conference (PEDSTC), pp. 1-6, 2020.

[11] J. Heidary, M. Gheisarnejad, H. Rastegar and M. H. Khooban, "Survey on microgrids frequency regulation: Modeling and control systems", Electric Power Systems Research, vol. 213, pp. 108, 2022.

[12] A. Malik, A. Haque, V. B. Kurukuru, M. A. Khan and F. Blaabjerg, "Overview of fault detection approaches for grid connected photovoltaic inverters", e-Prime-Advances in Electrical Engineering Electronics and Energy, pp. 100 035, 2022.

[13] J. Lamb and B. Mirafzal, "Open-circuit igbt fault detection and location isolation for cascaded multilevel converters", IEEE Transactions on Industrial Electronics, vol. 64, no. 6, pp. 4846-4856, 2017.

[14] S. N. Ajani and S. Y. Amdani, "Probabilistic path planning using current obstacle position in static environment," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170727.

[15] R. Fezai et al., "Effective random forest-based fault detection and diagnosis for wind energy conversion systems", IEEE Sensors Journal, vol. 21, no. 5, pp. 6914-6921, 2020.

[16] O. Y. Reddy, S. Chatterjee and A. K. Chakraborty, "Bilayered fault detection and classification scheme for low-voltage dc microgrid with weighted knn and decision tree", International Journal of Green Energy, vol. 19, no. 11, pp. 1149-1159, 2022.

[17] M. R. Miveh, M. F. Rahmat, A. A. Ghadimi and M. W. Mustafa, "Control techniques for three-phase four-leg voltage source inverters in autonomous microgrids: A review", Renewable and Sustainable Energy Reviews, vol. 54, pp. 1592-1610, 2016.

[18] P. Khobragade, P. Ghutke, V. P. Kalbande and N. Purohit, "Advancement in Internet of Things (IoT) Based Solar Collector for Thermal Energy Storage System Devices: A Review," 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2022, pp. 1-5, doi: 10.1109/PARC52418.2022.9726651.

[19] Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. Future Gener. Comput. Syst. 2018, 88, 173–190.

[20] Al-Shammari, N.K.; Syed, T.H.; Syed, M.B. An Edge–IoT Framework and Prototype based on Blockchain for Smart Healthcare Applications. Eng. Technol. Appl. Sci. Res. 2021, 11, 7326–7331.

[21] G. Tan, J. Wei, W. Zhao, L. Qi and X. Sun, "Application of threedimensional unbalanced coordinate transformation to stand-alone fourleg voltage-source inverter", IEEE Transactions on Power Electronics, vol. 37, no. 10, pp. 11-11 703, 2022.

[22] Z. Touati, M. Pereira, R. E. Araújo and A. Khedher, "Comparative study of discrete pi and pr controller implemented in srg for wind energy application: Theory and experimentation", Electronics, vol. 11, no. 8, pp. 1285, 2022.