**A Tree-Based Classifiers Fault Diagnosis Classification of Dry Clutch with Deep-Learning Networks for Feature Extraction**

*Authors , Kaushik Rajesh; Dr. Arun Balaji; Dr. Naveen Venkatesh; Dr. Sugumaran V*

**Abstract:**

Despite the flow of technological advancements and life made easier models of automobile manufacturing, people have a strong attachment towards the Manual Transmission. A major component that enables an easy travel in manual cars is a good clutch system. These are prone to faults due to their excessive usage especially in crowded and traffic situations. Faults in such a major component of the car can lead to further damage in internal components, difficulty in changing gear and ride discomfort. Hence, it is necessary to detect these various faults and solve them at the earliest. In this study, an attempt was made to realise 5 different fault conditions in a clutch system; namely, release fingers worn out (RFW), pressure plat broken (PPB), pressure plate worn out (PPW), tangential strip bent (TSB) and friction material loss (FML) using various deep-learning models (transfer learning) and tree-based classifiers. Firstly, sets of vibration signals was acquired for each of these faulty conditions as well as a good condition. This was then converted into a spectrogram plot and stored as image data. On this image database the concept of transfer learning including 5 different deep-neural networks like, AlexNet, DenseNet-201, GoogleNet, ResNet-50 and VGG16 were used to extract features. Then using the ID3 algorithm in WEKA software, features were selected meticulously to reduce the computation and ensure maximum accuracy. Finally using 20 different tree-based models, classification was performed and the result that was obtained showed a promising accuracy result. Using the features extracted by AlexNet and classified by FunctionalTree (FT) classifier, a test accuracy of 98.33% with training accuracy 100% and cross-validation accuracy 98.54% was achieved.

**Keywords:** transfer learning, deep learning, manual transmission, clutch system, AlexNet, GoogleNet, DenseNet-201, ResNet-50, VGG16, FunctionalTree, ID3 algorithm

**Introduction:**

The need for fault diagnosis in the clutch system is a vital endeavour and can reduce the time effort of labourers. Especially as the task of checking for faults in this system is a highly skilled activity as mentioned by (James, 2022). Hence there have been ample amounts of research works on this, using various techniques. (L. Xue et al., 2022) have used the method of extracting statistical features from vibration signals and using naïve bayes algorithm to classify the faulty conditions. This method has been performed for a wet clutch system. Another interesting method incorporated by (J. Xue et al., 2021) is the Hilbert Spectrum Entropy, used to identify the tiny differences caused by faults in the system in a more accurate fashion. Further emphasising on the importance of the clutch system is the work by (Lv & Wu, 2020). In which a fault diagnosis of the Dual Clutch Transmission is achieved using a Variable Force Solenoid (VFS) valve model and the Genetic Algorithm (GA). A clutch failure is something that can be very unexpected and sudden.

This causes great loses in time and money as well as endangering occupants’ safety and damage of internal components(Ompusunggu et al., 2013). In order to ensure that these faults are kept to minimum and smooth working of the clutch, it is essential to use some of these fault diagnosis techniques enabling optimum performance (Makarova et al., 2018). Above are techniques used in the past for this fault diagnosis of the clutch system, however today every domain has seen the integration of machine learning and artificial intelligence. This has enabled the field to grow rapidly and advance further. However before imparting a new approach, it is essential to ensure that it fits the application. Many researchers have implemented machine learning and deep learning techniques for fault diagnosis of various other automobile components:

* Naïve Bayes Algorithm – Naïve Bayes is a supervised learning-based algorithm used for classification. It is a preferable method as it is very simple and has low computational cost and time. It is a classifier based on probability, meaning it predicts the occurrence rate of a particular event. (Sun et al., 2020)
* Support Vector Machine – SVM is one of the most popular supervised learning algorithms. It is very elegant, as it makes its classification based on a single mathematical equation. SVM is adept in choosing points that create a hyperplane that can be used as a decision boundary. Various optimizers can be added onto this method to further increase its efficiency and accuracy. (Men et al., 2023)
* K-Nearest Neighbour – KNN stores available data and classifies any incoming data based on the information it already has. It is a non-parametric classification algorithm, although it has a fast and efficient computation time it is only limited to provided data and is not able to make assumptions and interpretations. Because of this it is also called as a lazy learner algorithm. It works on purely similarity measure. (H. Wang et al., 2020)
* Logistic Regression – This is a widely used technique for classification as well as regression problems. Logistic regression gives a probabilistic value which lies between 0 and 1. The curve of the log likelihood function gives the probability or likeliness of something for example fault classification. (M. Wang et al., 2023)
* AlexNet – AlexNet is a pre trained deep-learning model. This was built on a very large dataset and hence it is robust even to variations in image. (Chakrapani & Sugumaran, 2023)However, the only downside is that it uses large convolution filters (5 x 5), as this does not solve the problem of gradient vanishing.(Gu et al., 2022)
* DenseNet201 – The greatest advantage of densenet pre-trained model is that it can achieve high accuracy as well as significantly reduce the computational time and thus making it very efficient. (Liu et al., 2022) (Y. Wu et al., 2022)
* ResNet50 – This model can achieve better accuracies and higher performance as it gave a way to overcome the vanishing gradient problem(He et al., 2021). However due to which it is a much more complex model (Chakrapani et al., 2023).
* GoogleNet – The size of this pre-trained model is much less than all the other models allowing for faster computation. However, some changes in the architecture are proposed going forward.(G. Yang et al., 2022)
* VGG16 – VGG16 can handle datasets with noise and provide a robust solution to image classification problems. (Cai et al., 2023) Also it is the best model in terms of its error which is only 7% test error. However, the main disadvantage of this model is that it has a large training time and computational power is also more because of its 500MB size (L. Yang et al., 2023).

All these models and classifiers are used purely for classification in the above examples. There are disadvantages accompanied with advantages in both ML and Deep-Learning approaches. Accumulatively, the major drawback of deep learning is the high computational time for classification. On the other hand, it is very effective for feature selection as it is a meticulous process (Q. Wu et al., 2019) . Hence a novel solution would be to use deep-learning models for feature selection and use the best machine learning models for classification.

A full end-to-end fault diagnosis consists of (1) data acquisition, (2) data pre-processing, (3) data extraction and selection and (4) data classification and fault diagnosis. (Chakrapani & Sugumaran, 2023). The above research works talk in detail about the feature extraction and fault classification steps but data acquisition and pre-processing are equally as important. There are many different methods explored for data acquisition, but the one that has gained the most traction due to its high accuracy results is vibrational signal acquisition (Keshtan & Nouri Khajavi, 2016). Once the vibration signals are acquired, they need to be pre-processed by plotting them. Many different plots have been used thus far. (Jaikrishna M. et al., 2023) have used radar plots to represent the vibrational data that they have got, followed by deep learning (DNN) models for classification. (Zhang et al., 2022) have utilized polar PMU data plots again followed by CNN. PMU is a synchronous phasor measurement unit and is a promising approach to fault diagnosis due to its speed and accuracy. The pre-processing stage is performed to enable the feature extraction stage to occur without complications and accurately. This is carried out using Colour Recurrence plots by (Petrauskiene et al., 2022).

The most used method of image plotting is the spectrogram plot. This view of the data is invaluable, as it can capture, analyse, and save even small changes in both amplitude and frequency of the signal. The spectrogram approach can combine the merits of both frequency and time domains, whilst showing the relationship between them. Hence it is of great use when trying to extract valuable features from it. This spectrogram plot has been used for drill fault diagnosis (Tran & Lundgren, 2020), roller bearing fault diagnosis (Kulevome et al., 2023), centrifugal pump fault diagnosis (Chennai Viswanathan et al., 2023) and electric grid faults as well (Ardito et al., 2022).

In this paper, a novel methodology of using pre-trained neural networks for feature extraction from a spectrogram plot obtained from vibrational signals, followed by the use of tree-based classifiers for classification is proposed. Tree-based classifiers have generally been the go-to for most classification problems (Balaji & Sugumaran, 2023).

**Table 1**

Tree-based classifiers used for fault diagnosis

|  |  |  |
| --- | --- | --- |
| BFTree | J48 | Simple Cart |
| ForestPA | J48 graft | Naïve Bayes Tree |
| FunctionalTree | J48 Consolidated | JCHAIDStar |
| HoeffdingTree | LMT | ExtraTree |
| OptimizedForest | CDT | DecisionStump |
| RandomForest | RandomTree | Naïve Bayes Tree |
| CSForest | REPTree |  |

In the present study, the best of the previously used pre-trained DNN models such as, AlexNet(Chakrapani & Sugumaran, 2023), GoogleNet (G. Yang et al., 2022), ResNet50 (He et al., 2021), DenseNet201 (Liu et al., 2022) and VGG16 (L. Yang et al., 2023) are used for feature extraction. Experiments were conducted by varying the hyperparameters of the more successful decision tree classifiers in order to achieve a greater accuracy. The scope of alteration of the hyperparameters were kept to a minimum to ensure that over-fitting of data does not occur.

The technical contributions of this paper are as follows:

* An experimental setup was made to resemble and recreate a working single plate automotive dry clutch system. The faults that are most common in this system are release fingers worn out (RFW), pressure plat broken (PPB), pressure plate worn out (PPW), tangential strip bent (TSB) and friction material loss (FML). These 5 faulty conditions were induced and vibration signals for each were obtained.
* The vibration signals were then fed into a MATLAB program to convert them into a spectrogram plot. These spectrogram images had to be resized to fit the input parameters of the pre-trained DNN models.
* 5 DNN models were used for feature extraction namely, AlexNet, ResNet50, DenseNet201, GoogleNet and VGG16.
* Then the relevant features were selected by using ID3 algorithm in WEKA software. The Tree was visualised and the features that contributed to the highest cross-validation accuracy were taken as the final selected features.
* Various tree-based algorithms were used to classify the selected features.
* The best performing algorithm along with the DNN used to extract those features are identified as the best combination for dry clutch fault diagnosis.
* A novel approach of combining DNN transfer learning models and machine learning classifiers for achieving higher accuracy in dry clutch fault diagnosis has been envisioned in this study.

**2. Experimental Studies**

This section discusses the various elements involved in setting up and conducting the experiment required to acquire vibration signals. This setup was setup to mimic the real time dry clutch system currently in place in automobiles. First an accelerometer is stuck onto the apparatus with no faults and the good signal is obtained. Following which each fault is manufactured and tested on the system and then the accelerometer is place and these faulty signals are also obtained. The overall methodology is portrayed below in Fig. 1.

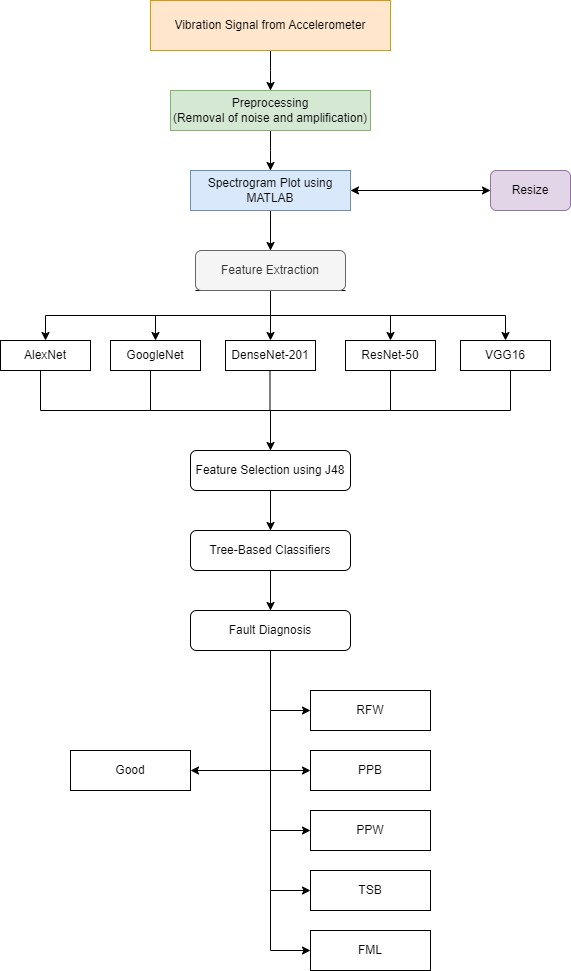


Fig. 1 Overall Methodology of Fault Diagnosis

*2.1 Experimental Setup*

In this study, a dry friction clutch plate of Maruthi Zen was used to enact and simulate the experiments that were required. This part is commercially available in the market. The test setup consists of a clutch, a flywheel, four bearings, one coupling, a load cell and an AC motor. Fig. 2 represents the test rig used in this study.

An alternating current (AC) motor with a constant rotational speed of 1400 rpm is connected to a flywheel. This is achieved using a shaft of 200 mm length and 25 mm diameter using muff coupling. The flywheel is coupled with a clutch assembly using bolts while bearings were used to support the shafts. A piezoelectric type sensor was used to obtain vibration signals. Accelerometers are the go-to choice when it comes to condition monitoring, as it can sense faults at high frequencies. A uniaxial accelerometer with a sensitivity of 10.26 mV/g was used in this study. By means of adhesive glues, the piezo-electric sensor is mounted on the plane surface (on the bearing) of the experimental setup which is further connected by a cable to the DAQ system.

The DAQ system consists of four analog input channels (A0, A1, A2, and A3). The accelerometer is attached to one end of the cable, and the DAQ system’s analog input–output (AIO) port is connected to the other. NI-LabVIEW is interfaced with the transducer signal and the system. A signal conditioning unit is included with the Data Acquisition System (NI DAQ). The signals are routed through an analog to digital converter and load converter before being stored directly in the storing device.(Chakrapani & Sugumaran, 2023)

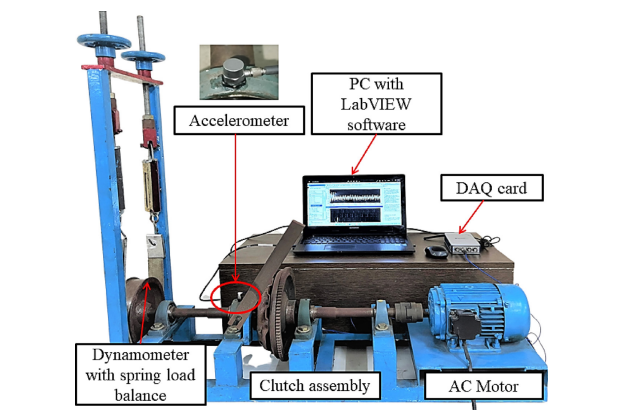


Fig. 2 Experimental test rig of dry clutch system

*2.2 Data Acquisition*

Data acquisition (DAQ) is the process of collecting sampling signal that measure the real-world conditions and converts them into digital numeric values. In the present study, fault diagnosis of the dry clutch system is done using the help of an accelerometer. The accelerometer is a piezo electric sensor with sensitivity of 10.26mV/G. The sensor is stuck onto the bearing of the system. The output obtained is fed into the NI 9234 DAQ using an USB where the analog signals are converted into digital numeric values. The parameters considered during this collection are:

* Sampling length – 8192 steps
* Sampling frequency – 25 kHz
* Number of Instances for each condition: 100

*2.3 Fault Conditions in Clutch System*

The clutch system is one of the most important components in the automobile as it is a crucial part of the powertrain. It plays a major role in the transfer of power from the engine to other components of the transmission system and hence a fault in the clutch system would be diabolical.

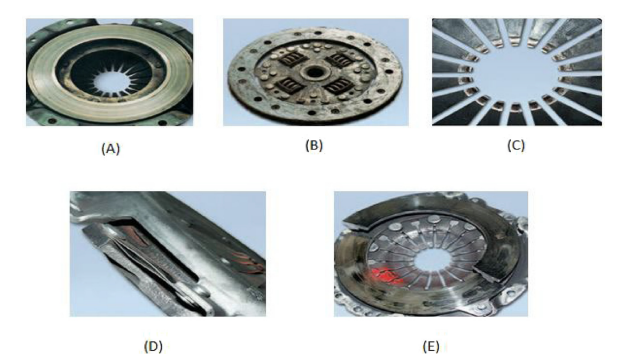


Fig. 3 Faults in dry clutch system (A) pressure plate worn out (B) frictional material loss (C) release fingers worn out (D) tangential strip bent (E) pressure plate broke

There are many faults that could occur and the reason for these occurrences are many. However, in this study, we have chosen the 5 most frequently occurring faults to the dry clutch system and examined the causes and symptoms of these faults as described in Table 2. Pressure plate, friction plate, release fingers, release bearings, diaphragm spring and clutch housing all together form the dry clutch system. Hence fault of these components is going to be measured in this paper. Additionally, these faults are caused due to prolonged usage, corrosion over time, moisture, and improper usage. Presence of such faults will degrade the performance and safety of the vehicle. The five different faults are shown in Fig. 2 above.

Table 2 Faults in clutch system

|  |  |  |
| --- | --- | --- |
| Fault Condition | Reason for Failure | Symptoms |
| Pressure Plate Worn out | Worn-out facings, incorrect release bearing clearance, partially disengaged clutch | Clutch Slip |
| Pressure Plate Broken | Pressure plate overheating due to continuous clutch slip, worn facing, oil contamination in facings, faulty slave cylinder | Clutch disengagement failure |
| Tangential strip bent | Drive train play, improper driving practice, wrong clutch component, irregular bolting | Clutch judder, clutch disengagement failure |
| Friction material loss | Heat generation between clutch plate and flywheel | Clutch clip and clutch judder |
| Release Fingers Worn out | Seized release bearing, incorrectly adjusted release system, excessive pre-load on bearing, damaged release bearing | Clutch slip, clutch judder, noisy clutch, disengagement failure |

**3. Feature Extraction using CNN pre-trained models**

In this section, one of the most important steps of data pre-processing is discussed. Feature extraction is performed to discriminate the most characteristic features from the signal shown in Fig. 5. From the acquired data, 5 different Deep Learning Pre-trained models were used to extract features. This was performed, because training a machine learning model directly with all the data points, will often yield poor results and creates information redundancy. All these 5 deep-learning models have a fundamental base framework network i.e., Convolution Neural Network.

*3.1 Convolution Neural Networks (CNN)*

Convolutional Neural Networks is a type of deep learning network-based model that consists of various layers that is well-suited for image processing and image recognition tasks. It is made up of many different layers which includes convolution layers, pooling layers, and fully connected layers as shown in Fig. 4. Each different pre-trained model has different amounts of layers and different specifications. For example, for the 5 models chosen for this study, AlexNet requires input image size 227x227 whilst the other 4 networks namely, GoogleNet, VGG16, ResNet-50 and DenseNet-201 require 224x224 size images.

*3.1.1 AlexNet*

This is one of the best convolutional neural networks for large scale visual recognition. AlexNet has achieved great accuracy in image classification and feature extraction. AlexNet has 8 layers in total, 5 convolutional and 3 fully connected layers. An important feature of AlexNet is the use of ReLU (Rectified Linear Unit). ReLU activation is performed at the end of every layer except for the final fully connected layer. The unique advantage in using AlexNet is that the image can be directly given as an input for classification. Also the convolutional layers of the model can automatically detect edges of an image and extract features from the fully connected layer. Also it is possible to improve the quality of the recognition of visual patterns by increasing the number of convolution layers.

*3.1.2 GoogleNet*

The GoogleNet architecture was proposed by Google in 2014 and hence is a relatively new model. It was a very strong model was it reduced the error rate of image classification by a significant amount to the former winner AlexNet. This architecture uses 1x1 convolutions in the middle of its computations and also a technique called global average pooling. This enables it to create a very deep architecture. This 1x1 architecture is used to decrease the number of parameters and also increasing the depth of the architecture. For example if a 4x4 convolution having 48 filters is performed then around 72 million different operations need to done to compute this normally. However with this 1x1 architecture a mere 3.9 million operations is enough. This drastic decrease is the reason for a faster computational time of GoogleNet. Global Average Pooling is a method in which feature maps are all averaged to a 1x1. This decreases the number of trainable parameters to 0 and also increases accuracy.

*3.1.3 DenseNet-201*

DenseNet-201 is a pre-trained model with 201 layers. It is an extension from the original DenseNet architecture proposed by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger in their 2017 paper titled "Densely Connected Convolutional Networks." Every layer in this network is connected to every other layer and hence it is a very dense architecture. This enables feature re-usage and helps in achieving higher and faster training accuracies. Not only this but it is also highly accurate due to these elaborate connections and hence is able to identify complex patterns and images. The only problem in this method is that it has incredibly high computational costs and complexity. It is also not very easy to adapt this network to different specifications and hence relies heavily on large amounts of exact training data.

*3.1.4 ResNet-50*

ResNet-50 is a very powerful deep learning architecture known for its incredible depth and high image recognition accuracy. It has the ability to capture complex features and also has built in regularization methods, such as normalization and dropout. This helps in preventing things such as overfitting and it also improves the generalization of the model. ResNet-50 is a special variant of the Residual Network, which was introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 paper titled "Deep Residual Learning for Image Recognition." ResNet-50 as evident in the name has 50 layers and it has also become a very popular choice for deep-learning applications. The fundamental innovation in ResNet architecture is the use of residual blocks. This introduces skip connections that allow gradients to flow more easily during training and hence helps to overcome the gradient vanishing problem. The only downside is again the computational costs, and the complexity of the model. Also to fully leverage the depth and the capacity of ResNet-50, a very large training dataset is required.

*3.1.5 VGG16*

VGG16 is short for Visual Geometry Group 16. It was invented by the Visual Geometry Group at the Oxford University. It is known for its simplicity and uniform architecture. It consists of 16 layers, 13 convolutional layers followed by 3 fully connected layers. It has been trained on large datasets and hence can be used for tasks such as classification and feature extraction. Researchers and practitioners can leverage the learned features to improve performance on their own datasets. Its deep architecture allows for a greater accuracy in classification and also enables it to learn more complex hierarchical features. The drawback of this network is its large size and hence computational costs. It contains layers that are fully connected to other layers and this requires a lot of computation. Also due to this it is very liable to overfitting. Unlike modern architectures such as ResNet and DenseNet, it does not contain skip connections or residual blocks.

*3.2 Convolution Layer*

The convolution layer is one of the first and most important layers in the network. The number of learnable kernels can be calculated using the parameters of this layer. Such kernels can spread across the input range with low spatial dimensionality. An activation map is created for every image that reaches the convolutional layer as an input. The kernels that are triggered when a particular feature is detected are determined by the scalar product for each data point that is passed into this layer. This is referred to as activation. The center element of the kernel vector is placed over each input from the input vector grid. Then convolution is performed and the neighbouring values are compared and changed. These layers may also reduce the complexity of the model. Three hyper parameters namely, depth (no. of filters), stride (movement of filter in one direction) and zeropadding (adding zeros around the border of input image) will optimize the performance of convolution layers.

*3.3 Pooling Layer*

Pooling layers often follow convolution layers. The need for pooling layers is to optimize the search of the pixels in a faster and more efficient way. It is capable of reducing the dimensionality of the grid obtained from the convolution method previously and condenses it down to a single dimension. This reduces the parameters and also the complexity of the model. Pooling layer works over each input activation map and uses the ‘‘MAX’’ function to scale its dimensionality. Since, pooling layers are destructive by nature; only two commonly known forms of max pooling are available. Both the stride and filters in the max pooling layer are fixed as 2 × 2 allowing the layer to expand the spatial dimensionality of the input throughout.

*3.4 Fully Connected Layer*

Fully connected layers form the ultimate layer of CNN network. The output from the final pooling or convolution layer will be the input to the fully connected layer, which is flattened and then fed into the fully connected layer. The final layer uses an activation function like soft-max or sigmoid function to classify the probability of identifying a particular class for a given input image.

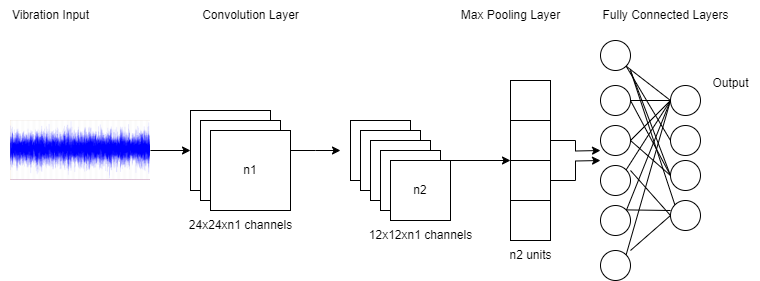


Fig. 4 General architecture of CNN

**4. Feature Selection**

In this section of the study, an important stage is described. Feature Selection is performed due the high amounts of features generated by the pre-trained models when extraction was done. This is a way to optimize the search for the most useful features and eliminate the unnecessary information, hence improving model building accuracy and efficiency.

Table 3. Accuracy using J48 algorithm before and after feature selection

|  |  |  |
| --- | --- | --- |
| CNN Models | Accuracy (%) | |
| Before Feature Selection | After Feature Selection |
| AlexNet | 92.83 | 95.83 |
| GoogleNet | 88.67 | 95.83 |
| DenseNet-201 | 88.16 | 95 |
| ResNet-50 | 92.33 | 95 |
| VGG16 | 88.67 | 90 |

There were 1000 features that were extracted from the vibration signals for each CNN model. This dataset containing all the features was fed into WEKA software and J48 algorithm was used to generate a decision tree. First the dataset was loaded onto WEKA and cross-validation accuracy was generated with all the features present. Then the tree was visualised Fig.6 and the most important nodes starting from top-bottom, left-right was arranged.

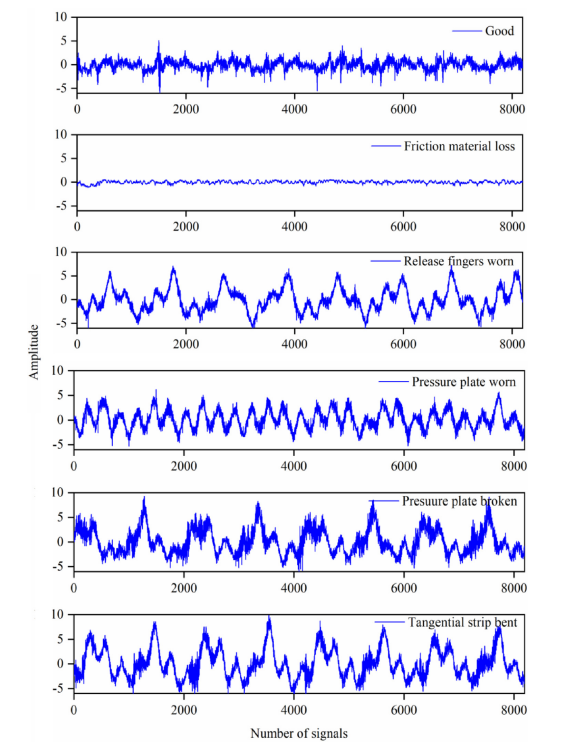


Fig. 5 Vibration plots for dry clutch system conditions (1) good, (2) FML, (3) RFW, (4) PPW, (5) PPB AND (6) TSB.

After listing out the required attributes in order, these particular attributes were selected on WEKA and the rest were removed. Now starting from the bottom of the list upwards the accuracies obtained using a combination of each of these attributes was noted. For each CNN the same process was followed and the attributes were reduced further, from 1000 to 20. Then with the visualisation of the tree, it was furthered reduced. This allows for greater accuracy. The difference in the size of the decision tree after feature selection is shown in Fig.7.

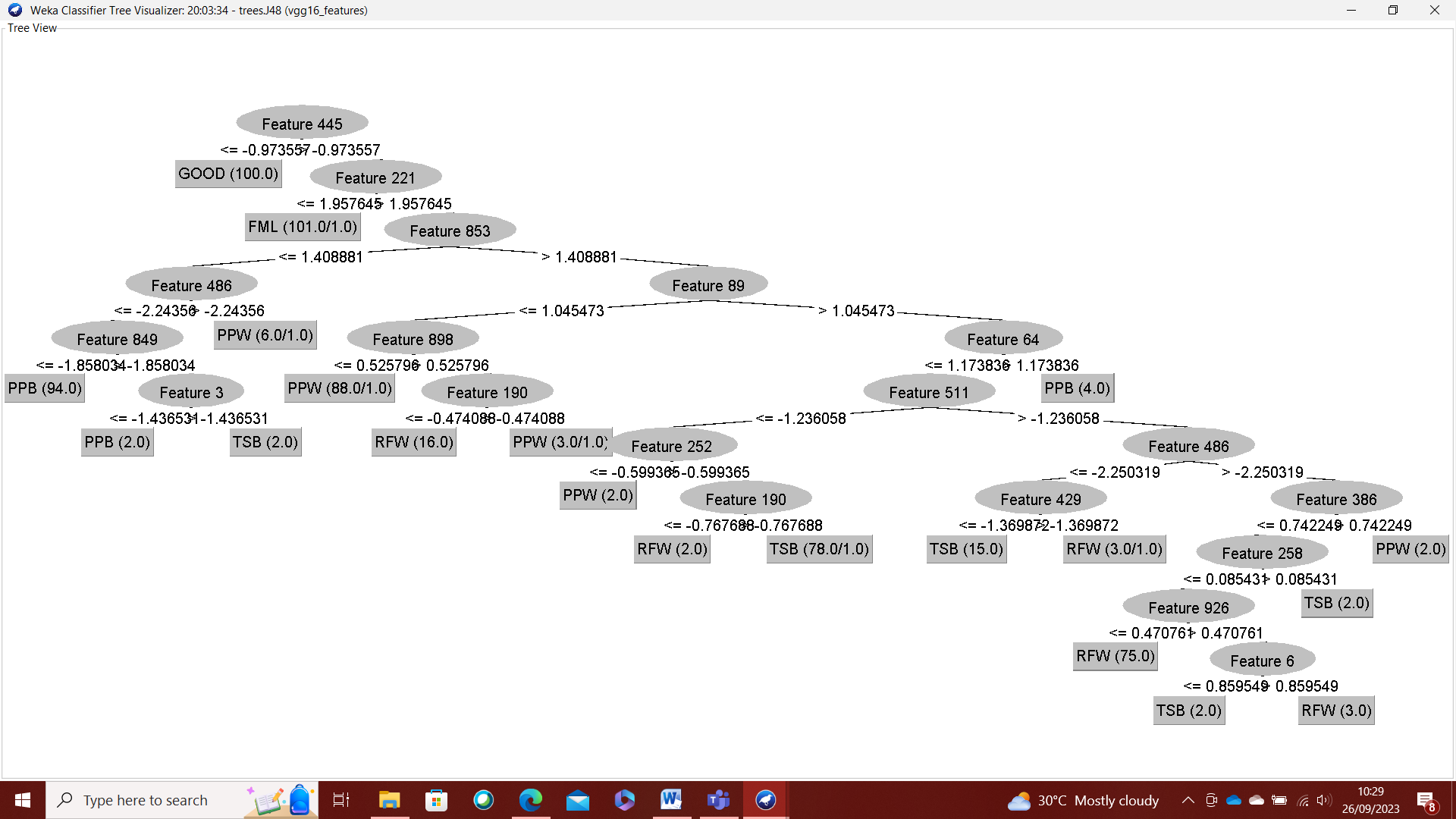


Fig.6 Feature Extraction Tree of VGG16 network

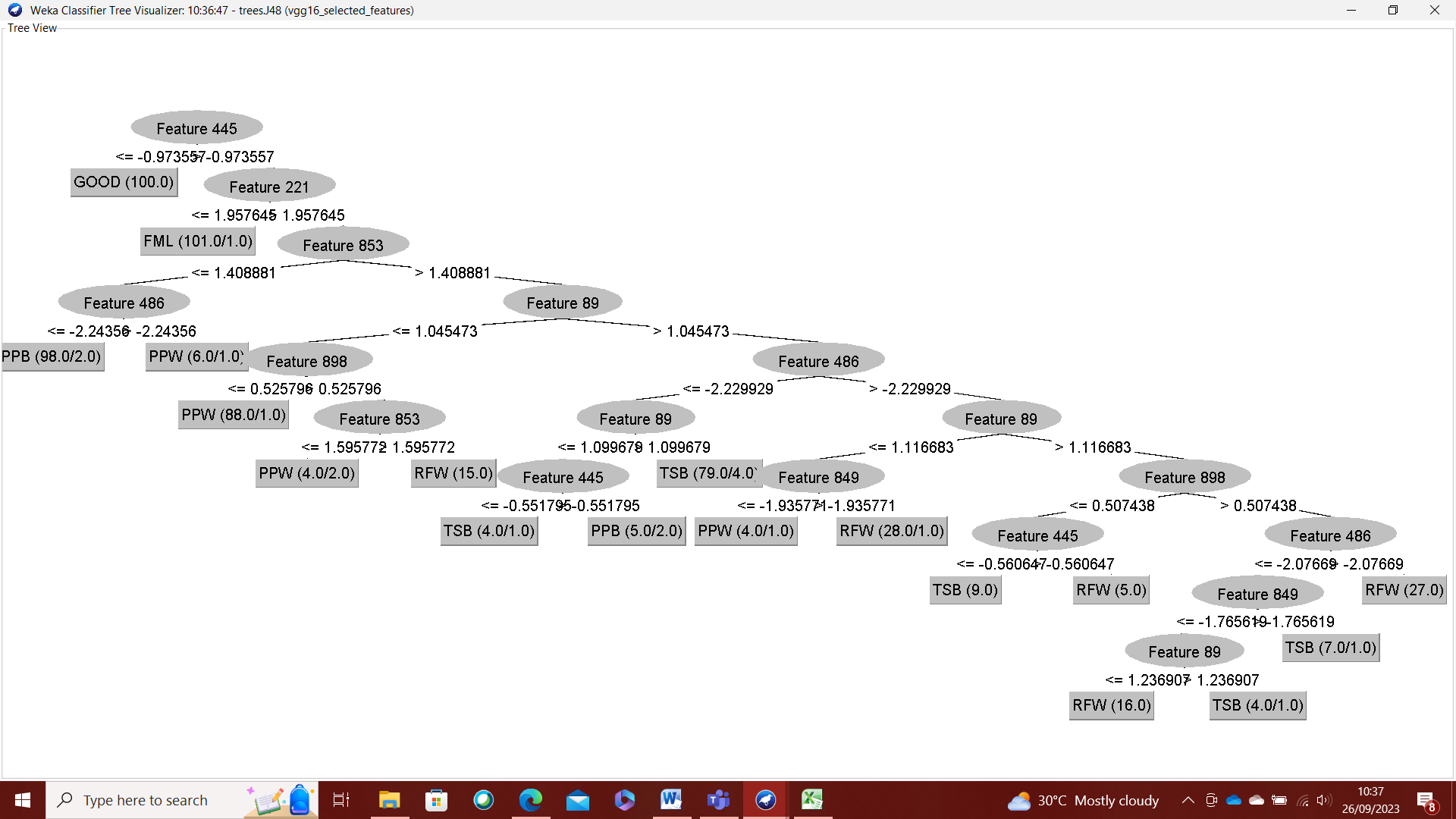


Fig. 7 Feature Selection Tree of VGG16 network

The tree of the VGG16 network after feature selection (Fig.7) has much less nodes and rules than the tree before feature selection. Also the classification accuracy has also improved from 88% to 92% after feature selection. Hence this shows the great importance of feature selection before model building. The best network for this particular fault diagnosis of dry clutch system will be determined by the best accuracy achieved after using the tree-based classifiers.

**5. Results and Discussion**

In this section, the performance of the 20 tree-based classifiers namely, ForestPA, FT, HoeffdingTree, OptimizedForest, RandomForest, CSForest, BFTree, J48, J48graft, J48Consolidated, LMT, CDT, RandomTree, REPTree, SimpleCart, NBTree, JCHAIDStar, ExtraTree and DecisionStump for fault diagnosis of dry clutch system is evaluated. The overall experiment was carried out on WEKA software tool and for each tree the training, cross-validation and test accuracies were obtained. When taking the top 5 classifiers that had the highest test accuracies and performing hyper parameter tuning, there were no changes in the accuracy. Hence it is important to note that there would be over-fitting of the data if the parameters are changed drastically.

Table 4. Performance of decision tree classifiers on AlexNet selected features

|  |  |  |  |
| --- | --- | --- | --- |
| Tree Classifier | TRAIN | CV | TEST |
| ForestPA | 99.375 | 97.92 | 98.333 |
| FT | 100 | 98.54 | 98.333 |
| HoeffdingTree | 98.333 | 98.333 | 98.333 |
| OptimizedForest | 100 | 97.29 | 98.333 |
| RandomForest | 100 | 97.5 | 98.333 |
| CSForest | 98.75 | 97.5 | 97.5 |
| BFTree | 98.75 | 95 | 96.67 |
| J48 | 98.95 | 95.41 | 95.833 |
| J48graft | 98.96 | 95.833 | 95.833 |
| LMT | 98.96 | 98.333 | 95.833 |
| CDT | 94.79 | 92.5 | 95 |
| RandomTree | 100 | 91.875 | 95 |
| REPTree | 94.79 | 93.125 | 95 |
| SimpleCart | 97.5 | 95.21 | 95 |
| NBTree | 99.58 | 93.125 | 94.166 |
| J48Consolidated | 98.54 | 95.21 | 93.333 |
| JCHAIDStar | 96.04 | 90.833 | 90 |
| ExtraTree | 100 | 89.17 | 79.17 |
| DecisionStump | 32.71 | 31.875 | 32.5 |

From the results obtained in Table 4, it is clearly evident that FT is the best classifier as it gives 98.33% testing accuracy. Although 4 of the other classifiers, ForestPA, HoeffdingTree, OptimizedForest and RandomForest give 98.33% too, FT also has the higher training and cross-validation accuracies and hence can be clearly shown as the best classifier for this instance. The confusion matrix of this classifier is shown in Fig. 9.

The results in Table 5 show the accuracies of decision tree classifiers that have been used on the selected features using GoogleNet. The best classifier was J48 with a test classification accuracy of 95.83%. The overall accuracy of the classifiers using this CNN features was 88.505%. This average value can be a little skewed due to the drastically low accuracy of DecisionStump. Disregarding DecisionStump as a classifier GoogleNet features have an average accuracy of 91.57%. The comparison of the average test classification accuracy for all the deep learning networks are shown in Table 9.

Table 5 Performance of decision tree classifiers on GoogleNet selected features

|  |  |  |  |
| --- | --- | --- | --- |
| Tree Classifier | TRAIN | CV | TEST |
| SimpleCart | 97.29 | 89.375 | 95.833 |
| CSForest | 97.5 | 93.54 | 95.83 |
| J48 | 98.95 | 92.08 | 95.83 |
| BFTree | 97.5 | 89.79 | 95.83 |
| LMT | 97.29 | 95 | 95 |
| ForestPA | 97.29 | 92.08 | 95 |
| J48graft | 98.95 | 92.08 | 95 |
| OptimizedForest | 100 | 93.95 | 94.16 |
| HoeffdingTree | 95.21 | 94.375 | 94.16 |
| RandomForest | 100 | 93.95 | 93.33 |
| RandomTree | 100 | 93.95 | 93.33 |
| FT | 98.75 | 94.58 | 92.5 |
| J48Consolidated | 96.45 | 89.79 | 91.66 |
| ExtraTree | 100 | 80.41 | 86.66 |
| NBTree | 98.75 | 85.625 | 85.83 |
| CDT | 90 | 90.21 | 83.33 |
| REPTree | 90 | 89.16 | 83.33 |
| JCHAIDStar | 93.125 | 85.83 | 81.66 |
| DecisionStump | 33.33 | 33.33 | 33.33 |

In all of the classification, it is evident that DecisionStump does not play a good role in fault diagnosis and hence can be eliminated as an option. The maximum accuracy achieved by DecisionStump is 33.33% which is a huge drop from the other classifiers. Out of all the classifiers ForestPA seems to be the most reliable in terms of overall accuracy as it places in the top half of the table of classifiers for every CNN extracted feature. The highest accuracy of ForestPA was 98.33%, which is the highest accuracy out of any of the others; however FT is prioritised ahead of this because of Forest PA’s lower training and validation accuracy.

The overall average accuracy would help in determining the best CNN for feature extraction. This comparison is clearly visualised in Fig. 8, highlighting the fact that AlexNet not only had the highest overall accuracy of 98.33% but also the highest average accuracy of 95%.

Table 6 Performance of decision tree classifiers on DenseNet-201 selected features

|  |  |  |  |
| --- | --- | --- | --- |
| Tree Classifier | TRAIN | CV | TEST |
| FT | 99.58 | 96.875 | 97.5 |
| LMT | 99.16 | 97.08 | 97.5 |
| ForestPA | 97.5 | 93.33 | 95.83 |
| OptimizedForest | 100 | 96.04 | 95.83 |
| RandomForest | 100 | 95.83 | 95.83 |
| J48Consolidated | 96.875 | 87.08 | 95.83 |
| HoeffdingTree | 98.125 | 97.08 | 95 |
| CSForest | 97.71 | 91.25 | 95 |
| J48 | 98.54 | 89.16 | 95 |
| SimpleCart | 95 | 86.04 | 95 |
| J48graft | 98.54 | 88.125 | 94.16 |
| BFTree | 97.5 | 86.25 | 93.33 |
| CDT | 92.92 | 84.79 | 91.67 |
| ExtraTree | 100 | 80.625 | 88.33 |
| RandomTree | 100 | 84.16 | 85.83 |
| REPTree | 89.16 | 83.95 | 85 |
| JCHAIDStar | 94.375 | 82.91 | 84.16 |
| NBTree | 99.58 | 91.45 | 83.33 |
| DecisionStump | 33.125 | 33.125 | 32.5 |

Table 7 Performance of decision tree classifiers on ResNet-50 selected features

|  |  |  |  |
| --- | --- | --- | --- |
| Tree Classifier | TRAIN | CV | TEST |
| CSForest | 98.75 | 96.25 | 98.33 |
| FT | 100 | 98.33 | 97.5 |
| OptimizedForest | 100 | 97.5 | 97.5 |
| RandomForest | 100 | 97.5 | 97.5 |
| LMT | 100 | 97.71 | 97.5 |
| ForestPA | 97.91 | 95.41 | 96.66 |
| HoeffdingTree | 98.96 | 98.75 | 96.66 |
| J48graft | 99.16 | 95.21 | 95.83 |
| J48 | 99.16 | 95.83 | 95 |
| BFTree | 99.375 | 96.25 | 93.33 |
| SimpleCart | 97.91 | 96.25 | 92.5 |
| NBTree | 99.58 | 91.46 | 91.66 |
| J48Consolidated | 98.75 | 93.33 | 91.66 |
| RandomTree | 100 | 91.875 | 90.83 |
| CDT | 95.41 | 92.71 | 90 |
| REPTree | 95.83 | 92.5 | 90 |
| JCHAIDStar | 95 | 90 | 85 |
| ExtraTree | 100 | 85.21 | 80.83 |
| DecisionStump | 33.33 | 33.33 | 33.33 |

Table 8 Performance of decision tree classifiers on VGG16 selected features

|  |  |  |  |
| --- | --- | --- | --- |
| Tree Classifier | TRAIN | CV | TEST |
| ForestPA | 97.5 | 93.75 | 91.66 |
| FT | 96.66 | 92.91 | 95 |
| HoeffdingTree | 92.5 | 91.875 | 91.66 |
| OptimizedForest | 100 | 94.79 | 90 |
| RandomForest | 100 | 94.79 | 90.83 |
| CSForest | 96.04 | 90.625 | 90.83 |
| BFTree | 98.33 | 92.08 | 88.33 |
| J48 | 97.71 | 91.25 | 90 |
| J48graft | 97.71 | 91.45 | 90.83 |
| LMT | 96.04 | 92.71 | 95.83 |
| CDT | 92.92 | 90 | 87.5 |
| RandomTree | 100 | 88.33 | 86.66 |
| REPTree | 92.91 | 90.83 | 87.5 |
| SimpleCart | 97.08 | 92.5 | 85.83 |
| NBTree | 98.75 | 90.625 | 87.5 |
| J48Consolidated | 97.71 | 90.83 | 89.16 |
| JCHAIDStar | 93.54 | 86.25 | 89.16 |
| ExtraTree | 100 | 82.5 | 83.33 |
| DecisionStump | 33.33 | 33.33 | 33.33 |

Table 9 Comparison of the average test accuracy of the 5 pre-trained networks

|  |  |  |
| --- | --- | --- |
| Pre-trained Network | Average including DecisionStump | Average excluding DecisionStump |
| AlexNet | 91.71 | 95 |
| GoogleNet | 88.51 | 91.57 |
| DenseNet-201 | 89.29 | 92.45 |
| ResNet-50 | 90.08 | 93.24 |
| VGG16 | 86.57 | 89.53 |

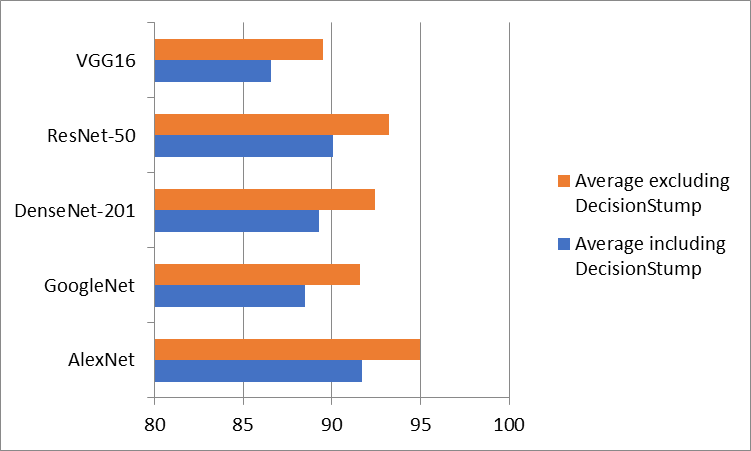


Fig. 8 Comparison of average test accuracy including and excluding DecisionStump

Table 6, 7 and 8 shows the performance of the decision tree classifiers for DenseNet-201, ResNet-50 and VGG16 models respectively. The DenseNet model features were able to give a maximum accuracy of 97.5% with the Functional Tree. This is another instance where Functional Tree has produced a high accuracy classification. Although it was not as high as the FT with the use of AlexNet, this was 5% higher than the average accuracy of the other classifiers. Overall this statistic is a great indicator as to how good of a classifier Functional Tree decision tree is in dry clutch fault diagnosis.

For the ResNet-50 model we can see that the highest classification accuracy is achieved by CSForest decision tree with a test accuracy of 98.33%. Although this is one of the highest accuracies achieved, the final best model was not given to this classifier because of the time taken to build the model. It took 0.14 s to build the model with CSForest classifier, whilst in AlexNet using Functional Tree it took a mere 0.01s. However, we can see a trend, where Functional Tree seems to be very efficient in all cases of classification.

Following this trend, for the VGG16 extracted features, FT again ranks in the top 5 with 95% accuracy which is only 0.83% off the best classifier LMT. LMT has classification accuracy of 95.83% which is the highest and far ahead of the average for VGG16. However by comparison as shown in Table 9, the VGG16 accuracies have been on average lower than the other pre-trained models. Hence this can also be ruled out. So overall the best classifier is Functional Tree and the deep-learning neural network that worked best for feature extraction was AlexNet. The confusion matrix of this combination classification is shown in Fig.9.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **a** | **b** | **c** | **d** | **e** | **f** |
| **a** | **20** | **0** | **0** | **0** | **0** | **0** |
| **b** | **0** | **20** | **0** | **0** | **0** | **0** |
| **c** | **0** | **0** | **20** | **0** | **0** | **0** |
| **d** | **0** | **0** | **0** | **20** | **0** | **0** |
| **e** | **0** | **0** | **0** | **0** | **20** | **0** |
| **f** | **0** | **0** | **2** | **0** | **0** | **18** |

Fig. 9 Confusion Matrix of FT classifier on AlexNet features (a) FML, (b) GOOD, (c) PPB, (d) PPW, (e) RFW and (f) TSB

**6. Conclusion**

In this paper, deep-learning pre-trained networks were used to extract relevant and important features from vibrational data acquired form a dry clutch system setup. J48 algorithm was used for feature selection and later 20 of the best tree-based machine learning classifiers were used to diagnose the faulty and good conditions in a dry clutch system. An experimental rig was setup and vibrational plots for 6 conditions, 5 faulty and 1 good was obtained.

The vibration signals form these conditions namely, release fingers worn out (RFW), pressure plat broken (PPB), pressure plate worn out (PPW), and tangential strip bent (TSB) and friction material loss (FML) were transformed into a spectrogram plot using MATLAB. These spectrogram plots were saved as images and this image dataset was subjected to resizing to fit the specifications of the pre-trained neural networks, AlexNet, GoogleNet, DenseNet-201, ResNet-50 and VGG16.

Only for AlexNet the size was 227x227, for all the others it was 224x224. Once this was fed into the networks the final fully-connected layer of each model was observed to obtain the final selected attributes. This was then fed into WEKA tool and J48 algorithm was used to select the features that contributed heavily to the fault diagnosis. Finally there was a 70-30 train test split followed by 20 machine learning tree-based classifiers for classification of faults. The best classifier was Functional Tree on AlexNet features with a test accuracy of 98.33%, cross-validation accuracy of 98.54% and a training accuracy of 100%.

Comparing this technique’s results to other state-of the art methods in Table 10 shows the monumental step forward in the accurate fault diagnosis of clutch systems. It would be great to deploy on-board each vehicle this type of accelerometer and fault diagnosis model software to be fully informed on the status of the dry clutch whilst driving the vehicle. The only limitation is that it costs a lot for the sensor to be implemented into the automobile. However this can also be reduced by the use of micro-electromechanical systems (MEMS).

Table 10 Comparison of different state of the art techniques for clutch fault diagnosis

|  |  |  |
| --- | --- | --- |
| Technique | Classification Accuracy (%) | Reference |
| BayesNet | 94.5 | Chakrapani and Sugumaran (2021) |
| Support Vector Machine | 92.5 | (J. Xue et al., 2021) |
| Fuzzy Neural Network | 95.0 | (Xueyuan 2010) |
| Naïve Bayes | 90.6 | (J. Xue et al., 2021) |
| Functional Tree and AlexNet Features (proposed technique) | 98.33 | This paper |

**References:**

Ardito, C., Deldjoo, Y., Noia, T. Di, Sciascio, E. Di, & Nazary, F. (2022). Visual inspection of fault type and zone prediction in electrical grids using interpretable spectrogram-based CNN modeling. *Expert Systems with Applications*, *210*, 118368. https://doi.org/10.1016/j.eswa.2022.118368

Balaji, A., & Sugumaran, V. (2023). Robust algorithm to learn rules for classification: A fault diagnosis case study. *FME Transactions*, *51*(3), 338–346. https://doi.org/10.5937/fme2303338B

Cai, Y., Xu, Z., Wen, Q., Shi, J., Zhong, F., Yang, X., Yang, J., & Zhou, H. (2023). Fault States Diagnosis of Marine Diesel Engine Valve Based on a Modified VGG16 Transfer Learning Method. *Mathematical Problems in Engineering*, *2023*, 1–14. https://doi.org/10.1155/2023/1225536

Chakrapani, G., Naveen Venkatesh, S., Aravinth, S., & Sugumaran, V. (2023). Transfer Learning-Based Fault Diagnosis of Single-Stage Single-Acting Air Compressor. *Journal of Vibration Engineering & Technologies*. https://doi.org/10.1007/s42417-023-01128-8

Chakrapani, G., & Sugumaran, V. (2023). Transfer learning based fault diagnosis of automobile dry clutch system. *Engineering Applications of Artificial Intelligence*, *117*, 105522. https://doi.org/10.1016/j.engappai.2022.105522

Chennai Viswanathan, P., Venkatesh, S. N., Dhanasekaran, S., Mahanta, T. K., Sugumaran, V., Lakshmaiya, N., Paramasivam, P., & Nanjagoundenpalayam Ramasamy, S. (2023). Deep Learning for Enhanced Fault Diagnosis of Monoblock Centrifugal Pumps: Spectrogram-Based Analysis. *Machines*, *11*(9), 874. https://doi.org/10.3390/machines11090874

Gu, H., Liu, W., Zhang, Y., & Jiang, X. (2022). A novel fault diagnosis method of wind turbine bearings based on compressed sensing and AlexNet. *Measurement Science and Technology*, *33*(11), 115011. https://doi.org/10.1088/1361-6501/ac8276

He, C., Jiang, Y., Sun, L., & Wu, B. (2021). Gearbox fault diagnosis based on transfer learning with RseNet50 model. *Journal of Physics: Conference Series*, *1986*(1), 012093. https://doi.org/10.1088/1742-6596/1986/1/012093

Jaikrishna M., A., S, N. V., V, S., Dhanraj, J. A., Velmurugan, K., Sirisamphanwong, C., Ngoenmeesri, R., & Sirisamphanwong, C. (2023). Transfer learning-based fault detection in wind turbine blades using radar plots and deep learning models. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, *45*(4), 10789–10801. https://doi.org/10.1080/15567036.2023.2246400

James, A. T. (2022). Diagnosis tree development for automobile clutch system faults based on operating parameter deviations. *Life Cycle Reliability and Safety Engineering*, *11*(1), 27–37. https://doi.org/10.1007/s41872-021-00182-z

Keshtan, M. N., & Nouri Khajavi, M. (2016). Bearings Fault Diagnosis Using Vibrational Signal Analysis by EMD Method. *Research in Nondestructive Evaluation*, *27*(3), 155–174. https://doi.org/10.1080/09349847.2015.1103921

Kulevome, D. K. B., Wang, H., & Wang, X. (2023). Rolling Bearing Fault Diagnostics Based on Improved Data Augmentation and ConvNet. *Journal of Systems Engineering and Electronics*, *34*(4), 1074–1084. https://doi.org/10.23919/JSEE.2023.000109

Liu, Z., Mao, L., Hu, Z., Huang, W., Wu, Q., & Jackson, L. (2022). A novel densely connected neural network for proton exchange membrane fuel cell fault diagnosis. *International Journal of Hydrogen Energy*, *47*(94), 40041–40053. https://doi.org/10.1016/j.ijhydene.2022.09.158

Lv, Z., & Wu, G. (2020). A Novel Method for Clutch Pressure Sensor Fault Diagnosis. *Vehicles*, *2*(1), 191–209. https://doi.org/10.3390/vehicles2010011

Makarova, I., Mukhametdinov, E., Mavrin, V., & Shubenkova, K. (2018). Improvement of the Vehicle Clutch’s Diagnosing System with the Use of Vibrodiagnostics. *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, 101–106. https://doi.org/10.1109/ITMC.2018.8691270

Men, Z., Hu, C., Li, Y.-H., & Bai, X. (2023). A hybrid intelligent gearbox fault diagnosis method based on EWCEEMD and whale optimization algorithm-optimized SVM. *International Journal of Structural Integrity*, *14*(2), 322–336. https://doi.org/10.1108/IJSI-12-2022-0145

Ompusunggu, A. P., Papy, J. M., Vandenplas, S., Sas, P., & Van Brussel, H. (2013). A novel monitoring method of wet friction clutches based on the post-lockup torsional vibration signal. *Mechanical Systems and Signal Processing*, *35*(1–2), 345–368. https://doi.org/10.1016/J.YMSSP.2012.10.005

Petrauskiene, V., Pal, M., Cao, M., Wang, J., & Ragulskis, M. (2022). Color Recurrence Plots for Bearing Fault Diagnosis. *Sensors*, *22*(22), 8870. https://doi.org/10.3390/s22228870

Sun, S., Przystupa, K., Wei, M., Yu, H., Ye, Z., & Kochan, O. (2020). Fast bearing fault diagnosis of rolling element using Lévy Moth-Flame optimization algorithm and Naive Bayes. *Eksploatacja i Niezawodność – Maintenance and Reliability*, *22*(4), 730–740. https://doi.org/10.17531/ein.2020.4.17

Tran, T., & Lundgren, J. (2020). Drill Fault Diagnosis Based on the Scalogram and Mel Spectrogram of Sound Signals Using Artificial Intelligence. *IEEE Access*, *8*, 203655–203666. https://doi.org/10.1109/ACCESS.2020.3036769

Wang, H., Yu, Z., & Guo, L. (2020). Real-time Online Fault Diagnosis of Rolling Bearings Based on KNN Algorithm. *Journal of Physics: Conference Series*, *1486*(3), 032019. https://doi.org/10.1088/1742-6596/1486/3/032019

Wang, M., Xu, X., & Yan, Z. (2023). Online fault diagnosis of PV array considering label errors based on distributionally robust logistic regression. *Renewable Energy*, *203*, 68–80. https://doi.org/10.1016/j.renene.2022.11.126

Wu, Q., Guo, Y., Chen, H., Qiang, X., & Wang, W. (2019). Establishment of a deep learning network based on feature extraction and its application in gearbox fault diagnosis. *Artificial Intelligence Review*, *52*(1), 125–149. https://doi.org/10.1007/s10462-019-09710-x

Wu, Y., Feng, Z., Liang, J., Liu, Q., & Sun, D. (2022). Fault Diagnosis Algorithm of Beam Pumping Unit Based on Transfer Learning and DenseNet Model. *Applied Sciences*, *12*(21), 11091. https://doi.org/10.3390/app122111091

Xue, J., Ma, B., Chen, M., Zhang, Q., & Zheng, L. (2021). Experimental Investigation and Fault Diagnosis for Buckled Wet Clutch Based on Multi-Speed Hilbert Spectrum Entropy. *Entropy*, *23*(12), 1704. https://doi.org/10.3390/e23121704

Xue, L., Jiang, H., Zhao, Y., Wang, J., Wang, G., & Xiao, M. (2022). Fault diagnosis of wet clutch control system of tractor hydrostatic power split continuously variable transmission. *Computers and Electronics in Agriculture*, *194*, 106778. https://doi.org/10.1016/j.compag.2022.106778

Yang, G., Wei, Y., & Li, H. (2022). Acoustic Diagnosis of Rolling Bearings Fault of CR400 EMU Traction Motor Based on XWT and GoogleNet. *Shock and Vibration*, *2022*, 1–12. https://doi.org/10.1155/2022/2360067

Yang, L., Xu, S., Yu, X., Long, H., Zhang, H., & Zhu, Y. (2023). A new model based on improved VGG16 for corn weed identification. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1205151

Zhang, X., Guo, Z., Zheng, Y., Liu, J., Yan, P., & Zheng, L. (2022). Power grid fault diagnosis using polar PMU data plots. *International Journal of Electrical Power & Energy Systems*, *141*, 108148. <https://doi.org/10.1016/j.ijepes.2022.108148>

 Xueyuan, W., 2010. Fault self-diagnosis system of automatic transmission based on fuzzy BP network. pp. 165–171

Chakrapani, G., Sugumaran, V., 2021. Health monitoring of automotive clutch system by using Bayes algorithms. IOP Conf. Ser. Mater. Sci. Eng. 1012, 012028. http: //dx.doi.org/10.1088/1757-899x/1012/1/012028.