Noname manuscript No.
(will be inserted by the editor)

Failing And Not Falling (F&!F): Data-Enabled Classification Learning of Aircraft Accidents and Incidents

Jarrod Carson · Kane Hollingsworth · Rituparna Datta · Aviv Segev

Received: date / Accepted: date

Abstract Journey by aircraft is the only option for long distance transportation and also one of the frequently used modes of transportation of passengers. As a result, safety of passengers and efficiency of the aircraft depend on maintaining efficient running conditions. Although many safety standards are followed in the design of the aircraft and thus there are fewer accidents, it is necessary to perform a thorough analysis to avoid risks that may occur during flight time. In the present work, we propose a maintenance strategy, Failing And Not Falling (F&!F), based on the Federal Aviation Administration (FAA) data in the United States. We work with the dataset of Boeing 737. The data consists of 72 features with 137,236 records which describe an aircraft accident or incident. These features are used to predict whether an incident will be identified during aircraft maintenance or during aircraft operation and what specific type of incident will occur. The prediction method is based on the integration of a decision tree and a unique neural network at each node of the decision tree. The results obtained using different architectures show how deep the neural networks should be, how to identify the relevant features, and the success of combining decision trees and neural networks. Moreover the neural networks and the decision tree approach also successfully identified the important features of maintenance. This method can be used for the maintenance of any data in multiple domains.

Keywords Machine learning · decision trees · neural networks · maintenance · aircraft

Jarrod Carson, Kane Hollingsworth, Rituparna Datta, Aviv Segev
Department of Computer Science
University of South Alabama
Mobile, AL, USA
E-mail: {jmc1627@jagmail., kmh1622@jagmail., rdatta@, segev@}southalabama.edu

1 Introduction

An aircraft is a very complicated dynamic system with interactions between several components. As a result, with the increase in the life of an aircraft, all the parts are exposed to different environmental, abnormal, and stress conditions which not only deteriorate the performance of the aircraft but also have an adverse effect on the structural components. Hence, maintenance of each sub-component as well as the overall system is necessary. The goal of all airline companies is to have a better overhaul, repair, and maintenance strategy to ensure the safety of passengers, improved quality of service, and minimization of running cost within minimum budget and schedule. A failure in one of the components of the aircraft may damage the whole system. Moreover, aircraft have structural, mechanical, heat generation, and electrical components, to name a few, and different domain experts are necessary for the maintenance of each one. The tremendous advancement of fast processing computers and prediction based machine learning algorithms started attracting researchers and industrialists to integrate maintenance with computer based predictive maintenance where a machine learning algorithm can serve as a domain expert of each component. The predictive maintenance is performed to decide when either the whole aircraft, a structure, or a subsystem is to be maintained. Unlike different types of maintenance practiced by experts, predictive maintenance of aircraft needs to identify a fault in a system or a subsystem while the entire system is in operation. An equipment maintenance prediction assists the business of an airline to grow by planning the maintenance method prior to failure. It can inturn save cost [1], time [2] and unnecessary maintenance activities [3] unlike condition based and periodical maintenance. In addition, predictive maintenance can also help to achieve better reliability [4] and efficiency [5] for the performance of the overall system. Predictive maintenance can be obtained in two different ways. Out of these two, the first is based on a classification method where the possibility of failure is predicted in the next N number of steps. The second one is based on regression in which the remaining time before failure is predicted. These two types of approach have the capability to save costs compared to the other type of maintenance methods, as in the case of predictive maintenance the schedule of maintenance is essential. As a result, all aircraft manufacturers and airline industries are directing their interest to the predictive maintenance concept as a method that anticipates before the actual failure.

All main components and sub-components of an aircraft must be in functioning condition during the run time, and as a result the performance of each system component must be continuously monitored to ensure safety of passengers as well as to avoid grounding of aircraft, both of which are directly related to the profit of the company. Predictive maintenance is the way to achieve necessary features, such as safety, smoothness of operation, and avoidance of unusual breakdown. The steps in aircraft predictive maintenance involve the collection, handling and processing of data. Each system component and subsystem component are integrated with different sensors which provide

real time data to monitor the system performance. The equipment failure can be determined based on the sensor data and data analytics method. In this way, the decision of maintenance is transformed into a data science problem, which is called predictive analytics. Machine learning and decision trees can be used to analyze the sensor data and predict the failure that may occur during run time.

We propose a Failing And Not Falling (F&!F) method for classifying the accidents and incidents according to the attributes collected from the aircraft system. An accident is an unexpected event that may result in property damage and results in an injury or illness. An incident, however, is an unexpected event that may result in property damage but does not result in an injury or illness. Incidents are also called "near misses" or "near hits." Our method is based on integrating neural networks with decision trees. A neural network is placed in each decision node of the decision tree.

In this paper, the dataset is obtained from Boeing 737 aircraft[1, 2]. The Boeing 737 is a twin-engine airplane operated in short-medium ranges from sea level (less than 6000 ft).

The remainder of this paper is organized as follows. In Section 2 we present comprehensive past research efforts in the field of aircraft maintanance, decision trees, machine learning, and neural networks. Then in Section 3, we present our proposed method based on neural network and decision trees called Failing And Not Falling (F&!F). Section 4 describes the extensive experiments and results with detailed discussion. Finally, the paper is concluded with the conclusions drawn from the proposed method and obtained results with a few potential directions of future research work.

2 Related Work

2.1 Aircraft Maintenance

According to federal aviation regulations, all aircraft must undergo maintenance after flying a certain number of hours [10]. Maintenance is carried out at night to allow for better aircraft utilization, and therefore aircraft remain overnight at a maintenance location every three to four days, according to the aircraft type, and a balance-check is performed periodically. After the schedule is set, the aircraft are routed to fulfill these maintenance requirements [8].

If maintenance is not performed properly, the aircraft may experience an accident [31, 32]. The airline companies must have a strategy to deal with crisis in case of an accident. An adaptive method for crisis ontology design can be found in [25] which can be used to represent knowledge in rapid response situations. The technique extends the ontology during a crisis and tailors it to the needs of the ongoing crisis. The extension of the above method is available

[1] https://www.faa.gov/data_research/aviation_data_statistics/data_downloads/
[2] https://av-info.faa.gov/dd_sublevel.asp?Folder=%5CSDRS

in [26]. The method merges ontologies and logic rules to represent the humanitarian needs and recommend appropriate humanitarian responses. The main advantage of the method is to identify humanitarian needs and to prioritize humanitarian responses automatically so that the decision makers are not overwhelmed with massive and unrelated information and can focus more on implementing the solutions.

Aircraft maintenance scheduling is one of the major decisions an airline makes during its operation [7]. When a flight schedule is set and aircraft are assigned to it, the aircraft maintenance-scheduling problem is to determine which aircraft should fly which segment and when and where each aircraft should go through the different levels of maintenance required by the Federal Aviation Administration. The objective is to minimize the maintenance cost as well as any costs associated with the re-assignment of aircraft to the flight segments.

A self-regulatory model was developed by McDonald et al. [6] to examine different safety management systems and safety culture in aircraft maintenance organizations, with emphasis on the human and organizational aspects. The model was effective in analyzing the relevant features of each organization's safety management system, although it underestimated the roles of planning and change.

Factors related to situation awareness in aviation maintenance teams were investigated by Endsley and Robertson [9]. In many environments, situation awareness was found to be critical for performance and error prevention. The research showed barriers and problems for situation awareness both across and within teams involved in aviation maintenance.

A predictive line maintenance optimization of redundant aeronautical systems is proposed in [30]. The optimization problem formulation was subjected to different wear conditions. The Kalman filter was used for degradation trends. Minimization of operation cost was performed based on dispatch requirements, delays, cancellations, and equipment costs.

Cognitive error models have looked into the unsafe actions that lead to many accidents in safety-critical environments [33]. Most models of accident causation are established on the idea that human errors are in the context of contributing factors. Yet published information on possible connections between specific errors and contributing factors is lacking. A survey using a self-completed questionnaire [11] reported that of a total of 619 safety occurrences involving aircraft maintenance, 96% were related to the actions of maintenance personnel. The research indicated the types of errors involved and the contributing factors associated with those actions. Each type of error was linked with a particular set of contributing factors and specific occurrence outcomes. The associations included links between memory lapses and fatigue and between rule violations and time pressure.

A short-term planning methodology of the line maintenance activities of an airline operator, at the airports, during turn-around time was proposed by Papakostas [13]. The methodology offered decision making for deferring maintenance actions that impact the dispatching of the aircraft, with the goals

of high fleet operability and low maintenance cost. A multi-criteria mechanism assessed a set of generated maintenance plan alternatives on the basis of health assessment and other information regarding operational and financial constraints at the operator's fleet level. An alternative was defined as the possible allocation of all deferred maintenance tasks to a set of suitable airport resources. The decision making criteria were cost, remaining useful life, operational risk, and flight delay.

Recent statistics on causes of aviation accidents and incidents show that to increase air-transportation safety the impact of human errors on operations must be reduced [12]. Aviation maintenance employees work under high-pressure conditions; they have strict time constraints and stringent guidelines. The primary advantages of computer-based systems for the training or support of technicians are that computers store and recall facts and can help humans clearly understand them. These features can help minimize errors from procedure violations, misinterpretation of facts, or insufficient training. Currently many factors, such as unwieldy hardware, the need to put markers on the aircraft, and the need to quickly create digital content, appear to interfere with effective aviation maintenance implementation in industry.

2.2 Decision Trees, Machine Learning, and Neural Networks

Previous research discussed the mapping of decision trees into a multilayer neural network structure that can be used for the systematic design of a class of layered neural networks, called entropy nets [15]. The research described a number of important issues such as automatic tree generation, incorporation of incremental learning, and generalization of knowledge acquired during the tree design phase. The work presented the number of neurons required in each layer as well as the desired output, thus leading to a faster progressive training procedure that enables each layer to be trained separately.

Another research compared the efficacy of particle identification in physics through artificial neural networks and boosted decision trees [16]. On the basis of studies of Monte Carlo samples of simulated data, the research found that particle identification with boosting algorithms performs better than artificial neural networks. In other works, prediction of electricity energy consumption and sound pressure level was analyzed using traditional regression analysis, decision trees, and neural networks [17, 22].

Comparison of neural networks [24], naive Bayes [21], and decision tree [23] classifiers for the automatic analysis and classification of attribute data from training course web pages was performed [18]. The work presented a naive Bayes classifier and used the same data sample through the decision tree and neural network classifiers to calculate the success rate of the classifier in the training courses domain. The results showed that the naive Bayes classifier was the best choice for the training courses domain.

One of the recent studies integrated Principal Component Analysis (PCA) with deep neural networks to predict multiple decay state coefficient [27]. The

results were compared with a different number of hidden layers. Another study predicted aerofoil self-noise at the early stage of the design using neural networks and hybridization of PCA with neural networks [28]. The results were compared between neural networks, PCA-neural networks, and different regression techniques. The results showed that the PCA-neural network outperformed all other techniques. In a recent work [29], a method is proposed to communicate between specialized neural networks. The developed method utilized independent sets of neural networks trained for specific tasks, while transferring knowledge among the neural networks that allows them to evolve chaining the input and output information. The method can allow different neural networks to be plugged in and knowledge transfer to evolve. It can also allow additional information to be requested, when the task at hand is difficult or hard to resolve. The method is known as OINNIONN - Outward Inward Neural Network and Inward Outward Neural Network Evolution. As the method can transfer the learning model, it can be useful for predictive maintenance of aircraft.

Most prior work compared classification methods such as neural networks, decision tree induction, and linear discriminant analysis [19,20,34,35]. Analysis of variance is used to identify any significant differences in the results of the methods. The issues of finding the most appropriate network size and using an independent validation set to determine when to stop training the network are also discussed. However, the integration of decision trees and neural networks as a unique classifying method for aircraft maintenance has not been described in the literature.

3 Failing & !Falling (F&!F): A Decision Tree and Neural Networks Classifying Method

3.1 Uniqueness of the Problem

Compared to many other classification problems, the issue of determining a cause of an aircraft accident or incident usually depends on multiple attributes. Furthermore, any solution found consisting of a classification method would not identify the cause but identify the main feature, or subsystems, which contributed to the accident or incident.

The problem of classification is based on the assumption that many values of the data are missing. In addition, certain records have fields which might have been mislabelled due to erroneous handling or just to save time during the recording of an incident.

Last but not least, an error which leads to an accident could easily cause a large number of casualties. Since the aircraft types are shared by multiple organizations, the chance of a rare incident reoccurring is relatively high. Therefore, the classification of an incident at an early stage is critical.

3.2 Problem Setting

Since we aim to use a decision tree and multiple neural networks for learning to classify incidents and accidents, the first step is quantifying the input in numeric values. Many of the inputs are formatted as textual labels or free text affiliated with the value such as: Aircraft Make, Aircraft Model, and Part Name. These labels for each input attribute were assigned a numerical value to represent all possible labels. The numerical value was designed to have a uniform distribution $U(0, 1)$, with -1 for no value.

The next issue was to define the depth of each neural network in each node of the decision tree. Due to the success of deep neural networks in multiple domains, we analyzed how deep the neural network should be to optimize the results. We analyzed how many hidden layers, n, are required to optimize the neural network performance. Although theoretically it could be assumed the deeper the better $n \longrightarrow \infty$, in reality, we show in the experiments (Section 4) that the optimum number of hidden layers is reached fairly quickly at 3-4 layers. Furthermore, after a fixed number of added hidden neural network layers, the results suddenly drop to be equivalent to guessing. In our experiments, adding any additional hidden layers above 29 results in a drop to 50% of the performance results measured, which is equivalent to guessing in a binary decision tree.

Theoretically, the number of inputs in the neural network should be equal to the number of variables which are available. The assumption is that the neural network can learn to ignore the attributes which do not contribute to the optimization of the solution. In reality, these are two different tasks which should be handled by different neural networks:

- Classifying the important contributing input variables.
- Optimizing a single decision in a classification.

For classifying the important contributing input variables, a neural network was trained using all 72 inputs. Once the neural network results have converged to a fixed value, we evaluate the weights. The weights between the input layer and the first hidden layer represent the importance of each input variable. These weights were recorded for each input variable. Then the absolute value of every weight is taken and the average of these absolute values is recorded for each variable. The inputs with the highest averages are determined to be the significant inputs. Input variables with low mean weight value have less contributing effect to the optimization of the classification and therefore were removed.

The experiments show that limiting the number of input variables contributed to the performance of the classification. The best number of input variables can be determined by organizing the input variables in descending absolute average weight order and either adding or removing one variable at a time until there is a change in the output performance. It should be noted that a large mean weight difference does not always correlate to a large difference

in the performance. However, the descending order of the mean weight is an important factor contributing to the output performance.

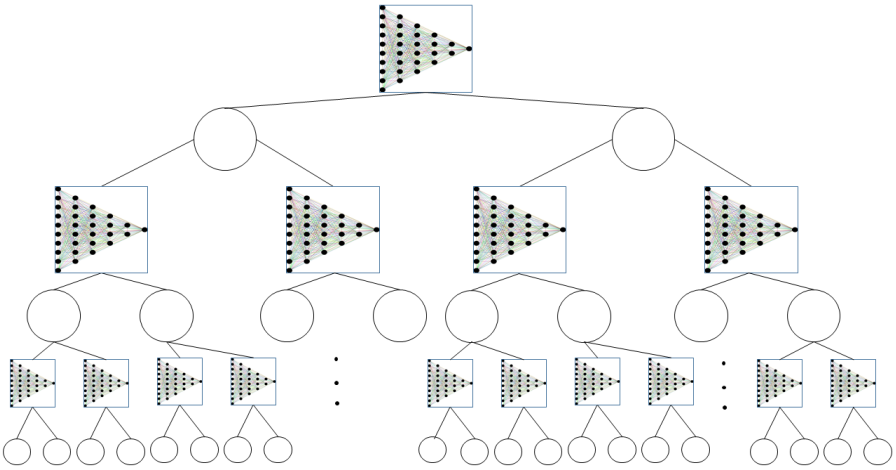


Fig. 1: Integration of Decision Tree - Neural Network in F&!F Method Where Each Classification Node is a Specialized Neural Network.

3.3 Integrating Decision Trees and Neural Networks

Next, we aim to optimize a single decision in a classification. We integrate the decision tree approach with neural networks. We create a decision tree with a neural network at each node of the decision tree, displayed in Fig. 1.

The unification of the decision tree and neural network approach allows us to integrate the advantages of both methods. The neural network works well while classifying into categories where the boundaries of classification are less distinct, but performance drops when there is a large number of categories. The decision tree works with a large number of categories which are distinctly classified.

The decision tree is built based on a set of possible results which can occur (accidents or incidents). For this, we choose the best possible result attribute with the highest information gain. To define information gain, we define a measure commonly used in information theory, called entropy, which characterizes the (im)purity of an arbitrary collection of examples [14].

Entropy $H(S)$ is a measure of the amount of uncertainty in the dataset.

$$H(S) = \sum_{c \in C} -p(c) log_2(p(c))$$

Where,

S - The dataset for which entropy is being calculated in the current iteration.

C - The set of the classes in $S, C = 0, 1$.

$p(c)$ - The proportion of the number of elements in class c to the number of elements in set S.

If $H(S) = 0$ then the set S is perfectly classified.

Information gain $IG(A)$ is the measure of the difference in entropy from before to after the set S is split on a result attribute A. This measures how much of the uncertainty S was reduced after splitting set S on result attribute A.

$$IG(A, S) = H(S) - \sum_{t \in T} p(t)H(t)$$

Where,

$H(S)$ - Entropy of set S.

T - The subsets created from splitting set S by result attribute A such that

$$S = \bigcup_{t \in T} t.$$

$p(t)$ - The proportion of the number of elements in t to the number of elements in S.

$H(t)$ - Entropy of subset t.

The information gain can be calculated for each remaining attribute. The attribute with the largest information gain can be used to split the set S on each iteration.

After selecting the attribute with the largest information gain, we build a neural network based on the previous criteria discussed in Section 3.2. For each maintenance problem, we construct a neural network which is designed to classify only if the problem occurs. Each neural network at each node of the decision tree consists of all the result attributes which could lead to a possible accident or incident. It should be noted that the result attributes represent the problem and are different from the input attributes filtered in the previous section.

Each leaf of the decision tree includes a neural network with a binary classification task which improves its performance. Since each neural network is tailored to a specific classification, the overall performance of the system does not depend on the performance of a single neural network.

The actual implementation does not necessarily require the implementation of neural networks for all possible problem attributes since many categories of problems in the area of maintenance can be classified under one classification category. Furthermore, a maintenance investigator can sometimes easily identify the correct cause at a higher level of the classifications.

4 Experiments and Results

4.1 Data

The Federal Aviation Administration collects all preliminary accident and incident information reported to the Office of Accident Investigation and Prevention. The data includes accident and incident data categorized by the aircraft manufacturer. The experiments focused on the Boeing 737 dataset.

The dataset included 72 variables used as inputs to each of the neural networks. Each neural network had a single neuron classifying whether the record belongs to the specified category. 137,236 records of the Boeing 737 were used in the experiments. The data was split into 75% training, 15% testing, and 10% validation. The testing data is used to test the accuracy and F1 of the neural network. The F1 is basically a weighted average between precision and recall. The measure F1 is chosen as a better representation of our model's performance as it takes into account the Precision and Recall values rather than correct predictions as Accuracy does. The validation data makes sure that there is no overfitting.

The following Table 1 details the input description. In addition to the 72 variables, the last value, Discrepancy, has a free text description of the accident or incident. The Discrepancy field was used to validate the value of the neural network results.

4.2 Data Preprocessing

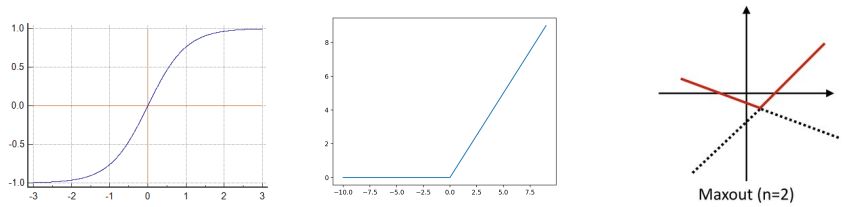
Before the data is fed into the neural networks, it is first preprocessed so that it may be interpreted by them. First the data is read from a CSV file in chunks, about 5000 records at a time, to a Pandas dataframe. Then a set of dictionaries is created, with one dictionary corresponding to one column/variable in the data, for mapping unique, non-numeric values in each variable to a numeric identifier. Then each column is parsed for unique values. If a value is encountered which is not in the dictionary for a given variable, then it is added to the dictionary along with a numeric identifier, and the identifier is incremented. This starts at an identifier of 100 and increases by 100 for each unique value found in a variable so as to adequately space the numeric values apart when normalizing the data later. However, when a null value is encountered, it is assigned -1 instead to separate it from non-null data. Once each variable has been parsed in a chunk, the dictionaries are used with a mapping function to convert all the non-numeric values in that chunk to their numeric identifiers. After this, the chunk is then written to a new CSV file with a similar name to the unprocessed CSV file to preserve the original. If more data exists in the CSV file, then another chunk is read and appended to each dictionary. The final step in the preprocessing stage, once all of the chunks have been converted, is the creation of a text file containing variable name headers and all of the numeric identifiers for each variable and which value each identifier

Table 1: Accident and Incident Data.

Operator Control Number	**Difficulty Date**
Submission Date	**Operator Designator**
Submitter Designator	**Submitter Type Code**
Receiving Region Code	**Receiving District Office**
SDR Type	**JASC Code**
Nature Of Condition A	**Nature Of Condition B**
Nature Of Condition C	**Precautionary Procedure A**
Precautionary Procedure B	**Precautionary Procedure C**
Precautionary Procedure D	**Stage Of Operation Code**
How Discovered Code	**Registry N Number**
Aircraft Make	**Aircraft Model**
Aircraft Serial Number	**Aircraft Total Time**
Aircraft Total Cycles	**Engine Make**
Engine Model	**Engine Serial Number**
Engine Total Time	**Engine Total Cycles**
Propeller Total Time	**Propeller Total Cycles**
Part Make	**Part Name**
Part Number	**Part Serial Number**
Part Condition	**Part Location**
Part Total Time	**Part Total Cycles**
Part Time Since	**Part Since Code**
Component Make	**Component Model**
Component Name	**Component Part Number**
Component Serial Number	**Component Location**
Component Total Time	**Component Total Cycles**
Component Time Since	**Component Since Code**
Fuselage Station From	**Fuselage Station To**
Stringer From	**Stringer From Side**
Stringer To	**Stringer To Side**
Wing Station From	**Wing Station From Side**
Wing Station To	**Wing Station To Side**
Butt Line From	**Butt Line From Side**
Butt Line To	**Butt Line To Side**
Water Line From	**Water Line To**
Crack Length	**Number Of Cracks**
Corrosion Level	**Structural Other**
Discrepancy	

represents. This allows both the user and the software to determine which identifier maps to which unique value for any given variable. Some variables are skipped entirely in the preprocessing stage if they are numeric and it has been decided that their numeric values are significant. The variables which were not mapped are Aircraft Total Time, Aircraft Total Cycles, Part Total Time, Part Total Cycles, Engine Total Time, and Engine Total Cycles. We chose these variables to be skipped as their numeric values would lose meaning if mapped to an arbitrary integer value. For example: Aircraft Total Time represents the total amount of time the aircraft has been in use. We determine this to be a potentially significant variable for predicting aircraft incidents. However, if we were to map a value of 3,892 hours to an integer of 300 and a value of 1,765 hours to an integer of 700 then the magnitude of usage time could lose its value. Additionally, as it is unlikely that no two aircraft will share the same amount of flight time mapping these values to unique integers could also add unnecessary complexity to the data with the volume of unique integer values.

4.3 Methods



(a) Tanh (b) Rectifier (c) Maxout

Fig. 2: Activation Functions Used in the Present Study

The following activation functions were used in the experiments:
Tanh - Hyperbolic Tangent Function (Fig. 2a))

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Tanh with Dropout - Tanh with a dropout activation function of a 0.5 ratio for each hidden layer.
Rectifier (default) - Positive part of its argument (Fig. 2b).

$$f(x) = x^+ = max(0, x)$$

Rectifier with Dropout - Rectifier with a dropout activation function of a 0.5 ratio for each hidden layer.
Maxout - Given an input $x \in R^d$, a maxout hidden layer implements the function (Fig. 2c).

$$h_i(x) = max_{j \in [1,k]} z_{ij}$$

where $z_{ij} = x^T W_{ij} + b_{ij}$, and $W \in R^{dmk}$ and $b \in R^{mk}$ are learned parameters.

Maxout with Dropout - Maxout with a dropout activation function of a 0.5 ratio for each hidden layer.

Multiple neural network configurations were analyzed for best performance. For the following experiments, a neural network with 3 hidden layers with 60, 40, and 20 neurons respectively was used.

4.4 Pseudocode - Neural Network

1: $CSVData$: Data read in from a preprocessed csv file.
2: $trainData$: Subset of CSVData used for training neural network.
3: $testData$: Subset of CSVData used for testing neural network.
4: $validData$: Subset of CSVData used for validating neural network.
5: nn: H2O DeepLearningEstimator neural network model.
6: $nnMetrics$: H2O dataframe containing performance metrics from testing the neural network.
7: $resultsCSV$: CSV file for containing neural network performance metrics.
8: $ImportH2O, H2ODeepLearningEstimator$
9: $CSVData = open(\text{"}csvfile.csv\text{"}, \text{"}read\text{"})$
10: $H2O.init()$
11: $H2O.read(CSVData)$
12: $trainData = 75\%$ of CSVData
13: $testData = 15\%$ of CSVData
14: $validData = 10\%$ of CSVData
15: $nn = DeepLearningEstimator(hiddenLayers, activationFunction)$
16: $nn.train(in_vars, out_vars, trainData, validData)$
17: $nnMetrics = nn.test(testData).performanceMetrics$
18: $resultsCSV = open(\text{"}results.csv\text{"}, \text{"}write\text{"})$
19: $resultsCSV.write(nnMetrics)$ =0

4.5 Experiments

The dataset is first preprocessed to conduct the experiments. The Pandas library in Python is used for preprocessing and simulation experiments. The strings in each column/variable of the dataset are parsed and mapped to an integer value. The starting value is selected as 100 and is increased by 100 for every unique subsequent string. This process is repeated for every variable individually. However, in the case of an integer input or floating point variables, magnitude is important (such as the total flight time of a 737), and the values are not mapped for that variable and are simply skipped. For all variables, a null value is mapped to -1. The data set is labeled in the present study. While classifying the inputs, we used the significant inputs from the root of the

decision tree where only “maintenance” or “non-maintenance” was classified. Through further experimentation we determine whether the significant input is changed at each node of the tree.

The dataset is classified into two categories: whether the problem with the aircraft occurred during maintenance and not during maintenance. Thereafter, the data is further classified into whether or not the problem involved cracks and whether or not the problem involved the fuselage. In this way, by classifying fuselage after classifying maintenance, we mean that we first identified that the problem occurred during maintenance and then identified that the problem involved the fuselage.

The first set of experiments analyzed how deep the deep neural networks should be. We analyzed a binary classification of accident or incident identification during Maintenance or Non-Maintenance. We increased the neural network hidden layers from 1 to 100 and checked how the F1 and accuracy results change. The Stopping tolerance = 0.0000001 was used for all of our experiments. This precise value of stopping tolerance ensures convergence of the Neural Network with higher performance. These experiments analyzed what the correct structure of the neural network would be.

The second set of experiments analyzed whether the bigger the data set, the better the results. We organized the input variables in descending order of the mean value of the weight connecting the input layer and the first hidden layer. We increased each variable in descending order of weight value and compared the Area Under the Curve (AUC), Accuracy, Precision, Recall, and F1 values. Each of these values was compared with the six types of activation functions.

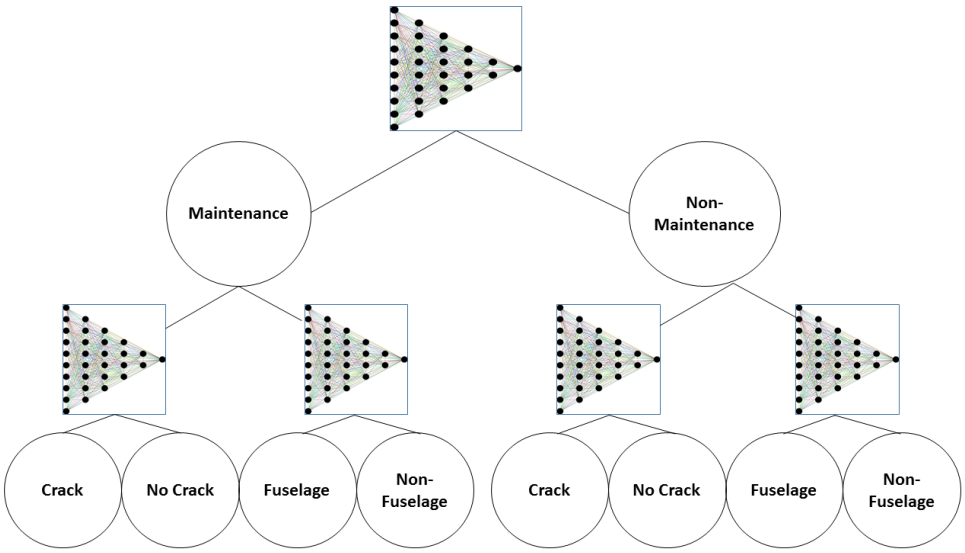


Fig. 3: Decision Tree - Neural Network Experiment

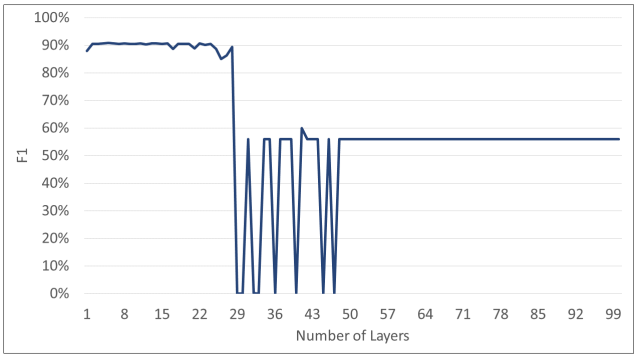


Fig. 4: F1 vs. Number of Layers

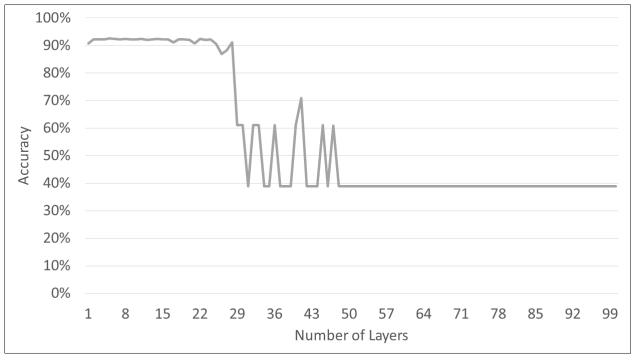


Fig. 5: Accuracy vs. Number of Layers

The third set of experiments analyzed the advantages of integrating the decision tree and the neural network methods. An outline of the set of experiments performed is described in Fig. 3. First a neural network was used to perform a binary classification into categories Maintenance or Non-Maintenance. Each of the classified records was then again classified into Crack or No Crack and Fuselage or No Fuselage. The classification into each of these subcategories was performed using all the data previously classified into the main category. In other words, a record from Maintenance and Crack could also belong to either Fuselage or Non-Fuselage but not to both. As can be seen from Fig. 3, four different neural networks were used to classify to the eight different sub-classifications.

4.6 Results

Fig. 4 and Fig. 5 display results of the analysis of the appropriate depth of the deep neural network. Results peak at three hidden layers and continue

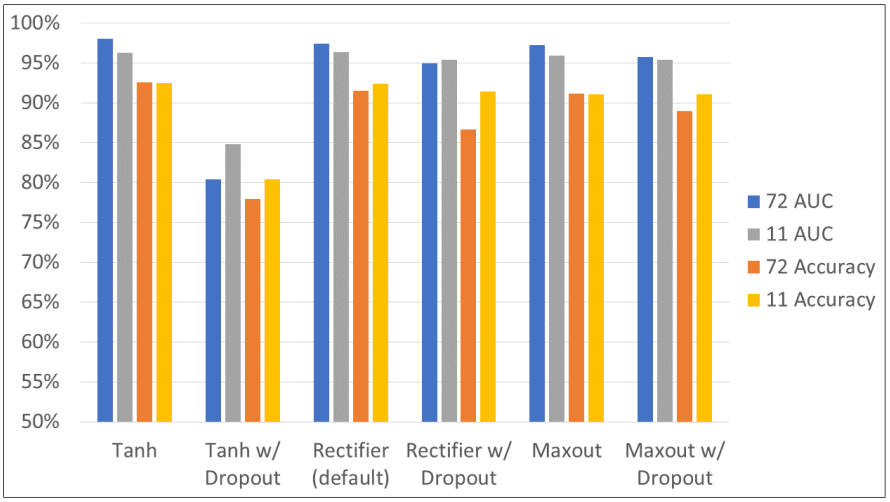


Fig. 6: AUC/Accuracy of All Inputs vs. Significant Inputs

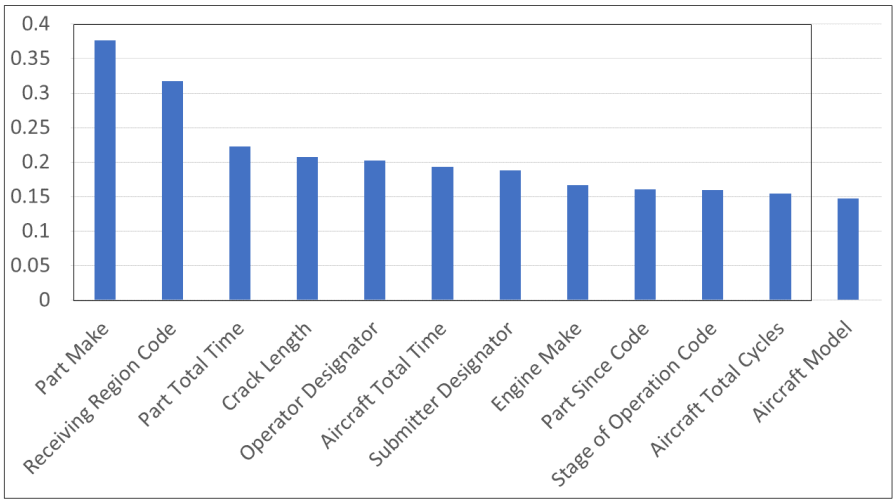
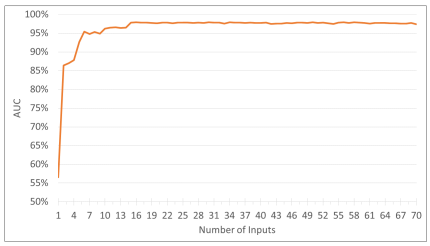
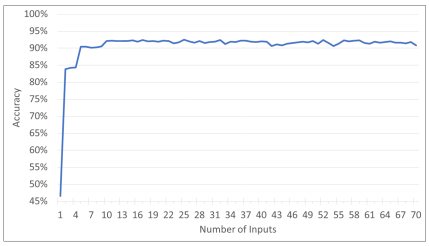


Fig. 7: Average Mean for Leading First Layer Input Weights

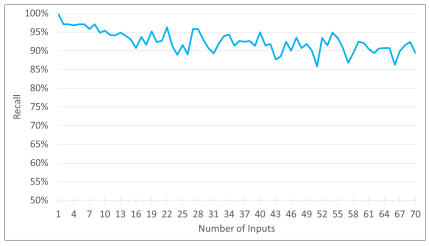
around the same F1 (Fig. 4) and Accuracy (Fig. 5) levels until 29 hidden layers. From this point onward, the results show that the network would be too deep. The network results show that over 29 hidden layers is equivalent to guessing in a binary classification. The F1 value becomes slightly above 50% and the Accuracy slightly below 50%. It seems that a three hidden layer neural network is accurate and fast enough to perform the task of classification. For the experiments that we performed the maximum amount of time that the



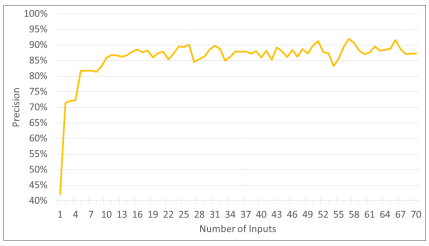
(a) AUC vs. Inputs



(b) Accuracy vs. Inputs



(c) Recall vs. Inputs



(d) Precision vs. Inputs

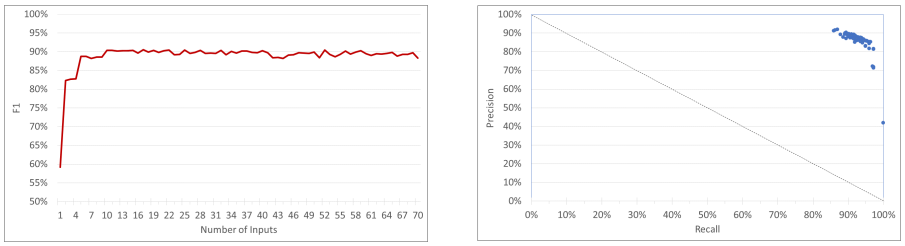
Fig. 8: AUC, Accuracy, Precall, and Precision vs. Number of Inputs

networks took to train the neural models spanned from 20 to 30 seconds for each network.

Fig. 6 presents the classification results into the Maintenance and Non-Maintenance categories as the number of input variables increases. Fig. 6 shows the AUC and Accuracy of all six activation functions comparing the results of using only the top 11 mean weight variables versus using all possible 72 variables. The results show the accuracy is almost the same, -0.12%, and up to 4.77% better when using only the top 11 variables. The AUC is less consistent and varies from -1.76% to 4.44% for using the top 11 identified variables versus all 72. The results show the advantage of the method of identifying the top variables before using the neural network as a classifying tool.

Fig. 7 shows the average mean for the leading inputs weight value between the input layer and the first layer. These identify the main variables which are relevant for high accuracy results. The top 11 variables appear in the circumference box. The results show that issues such as Part Make, Receiving Region Code, and Part Total Time can clearly be identified as the most relevant classifiers. The list of the leading main contributors for the accident and incident reports ends with Aircraft Total Cycles. The Aircraft Model is already identified as a less unique classifier for the type of issue involved. These values included the different models of the Boeing 737.

Fig. 8a presents the AUC as the number of inputs increases. Fig. 8b presents the Accuracy as the number of inputs increases. The inputs in the x-axis are arranged in descending order of mean weights leading from the input layer of the neural network to the first hidden layer. The AUC continues to increase



(a) F1 vs. Inputs (b) Precision vs. Recall

Fig. 9: F1 vs. Number of Inputs, Precision vs. Recall

as the number of inputs increases up to 16 inputs. However, the accuracy does not improve over 11 inputs which were identified as the important variables.

The AUC difference can be viewed as a less accurate value for measuring performance. In this case, it can be attributed to the low number of values measured to create the curve. This could explain the difference when measuring the area with AUC versus comparing a single Accuracy result.

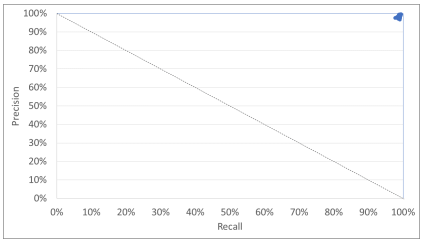
Similarly, Fig. 8c presents the Recall and Fig. 8d presents the Precision as the number of variables with the descending mean weight increases. These results display that the recall actually declines, from 100% to above 85% as more variables are added. However, the precision increases and stabilizes after the top 11 weighted variables are included. The results show that the recall has a slight drop as more variables are added. However, the precision is determined by the leading or "more important" variables.

These results can be be viewed more clearly when viewing the F1 value appearing in Fig. 9a. As the number of highly weighted variables is added the value peaks up to 11 variables. From 11 variables the F1 is stable at around 90%. Furthermore, the Precision vs. Recall in Fig. 9b shows that most results are clustered in the top right except for the initial values with high recall.

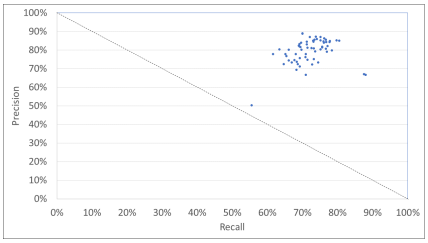
The results show the correct identification of the important inputs by the method of classifying mean weights in descending order. The additional input variables which do not seem to improve the results can be attributed to constant values, variables which are dependent on other inputs, or values which are inconsistent with the expected results.

Fig. 10 describes the precision versus recall of the lower level classification of the decision tree presented in Fig. 3. Each neural network performance on the second classification level is presented.

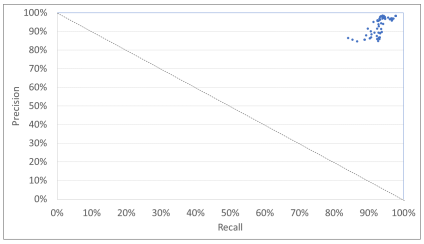
Fig. 10a shows a very high precision and recall level of classifying a Crack after previous Maintenance classification has been performed. Similarly, classifying a record as Non-Maintenance and then classifying the Crack has good precision and recall results (Fig. 10c). This shows the advantage of using the decision tree with the neural network to classify correctly well defined categories.



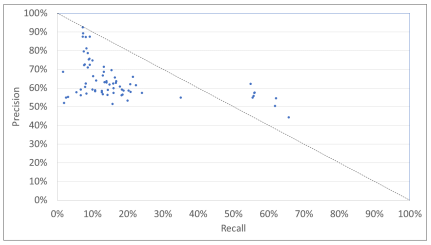
(a) Precision vs. Recall for Maintenance and Crack



(b) Precision vs. Recall for Maintenance and Fuselage



(c) Precision vs. Recall for Non Maintenance and Crack



(d) Precision vs. Recall for Non Maintenance and Fuselage

Fig. 10: Precision vs. Recall for Maintenance, Crack, and Fuselage

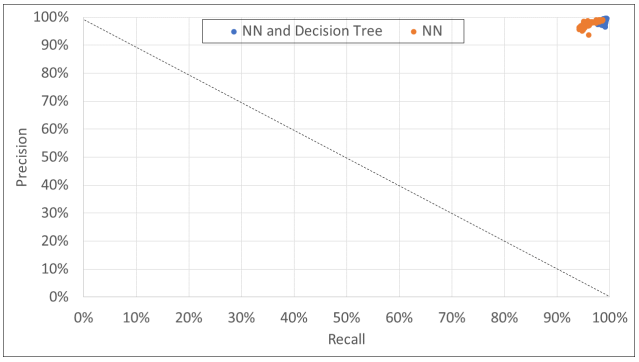


Fig. 11: Methods Comparison

On the other hand, Fig. 10b and Fig. 10d show what happens when the decision tree and neural network are not aligned correctly. In this case, classifying Fuselage after classifying Maintenance has slightly lower precision versus recall results. However, the classification of Fuselage after the classification as Non-Maintenance has been performed is centered around the diagonal, which represents guessing in a binary classification. This shows the incorrect decision tree structure. One less likely possible explanation is that all Fuselage classifications are only identified during Maintenance. Another, more likely, option is

that this part of the decision tree is not properly constructed. In other words, Fuselage under Non-Maintenance cannot be classified. This means that the dataset did not have a sufficient number of cases where problems with the fuselage occurred during a time when the aircraft was not undergoing maintenance, and therefore the present method could not accurately be classified with our current methods.

At least one more layer of sub-classification needs to be added in order to correctly identify this issue. Another concept should be added to the decision tree below Non-Maintenance before trying to identify whether there is a Fuselage problem.

Finally, Fig. 11 shows the advantage of our F&!F method integrating the neural networks with the decision tree compared to the commonly used method which uses just neural networks for classification. The figure shows the precision and recall as the number of inputs increase. The F&!F method outperforms the method of using only neural networks for both precision and recall.

4.7 Activation Function/Hidden Layer Tests

Tests were performed to determine how different activation functions and hidden layer architecture contributed to the overall performance of the neural networks, which was measured through Accuracy, AUC, Precision, Recall, Precision-Recall AUC, and F1 scores. The activation functions used were Rectifier, Tanh, Maxout, Rectifier w/ Dropout, Tanh w/ Dropout, and Maxout w/ Dropout, where each of the dropout rates was 0.5. The hidden layer architectures all consisted of three layers with differing quantities of neurons in each layer. The layers began with [40, 30, 10] (the default architecture used for most other experiments due to its high performance) and subsequent architectures tested followed the pattern: [40, 30, 9], [40, 30, 8], ..., [40, 30, 5], [40, 25, 10], ..., [40, 5, 5], [35, 30, 10], ..., and ended with [10, 15, 5]. These architecture tests were conducted alongside the activation function tests so that each architecture was tested with each of the six activation functions. The tests concluded that a neural network using the Tanh activation function performed better than the other activation functions used, with each of the dropout functions performing the worst. The best test resulted in a 92.4675% accuracy and a hidden layer of [35, 15, 9].

4.8 Pearson/Spearman Correlation Tests

Pearson and Spearman correlation tests were also performed to find statistical correlation between the variables when comparing them to the discrepancy column. These tests were run on the enumerated data. Aircraft Total Cycles was shown to be the most significant variable with both the Pearson and the Spearman test. These results could be implemented/applied in the future to find out which significant inputs could be used for training the neural networks.

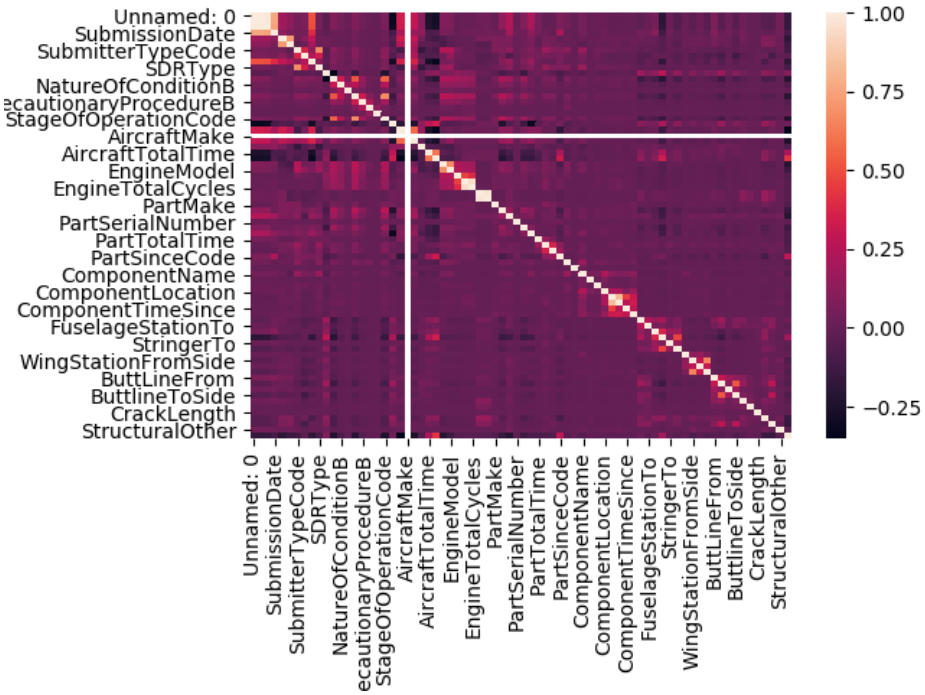


Fig. 12: Correlation Heatmap Crack

4.9 Heatmap Information

The heatmaps that are represented in Fig. 12 and Fig. 13 demonstrate the correlation of every variable to each other using the Pearson/Spearman statistical correlation algorithm to find out how closely associated each variable is with another. The closer the number is to 1, the higher the positive linear correlation is with the variable being compared and the greener the area is in the heatmap. The closer the number is to -1, the higher the the negative linear correlation is with the variable being compared and the redder the area is on the heatmap. When the number is zero, it means that there is no linear correlation between the two variables and the area is yellow. So, the closer the number is to | 1 | the more association the variables have with each other.

4.10 Adaptive Learning Rate

The adaptive learning rate H2O uses for its gradient descent algorithm was tweaked by manipulating two of its factors: Rho, which is the adaptive learning rate time decay factor, and Epsilon, the adaptive learning rate time smoothing factor. Rho relates to memorizing past weight updates and affects the influence of past gradients. Epsilon assists the model with encountering and overcoming

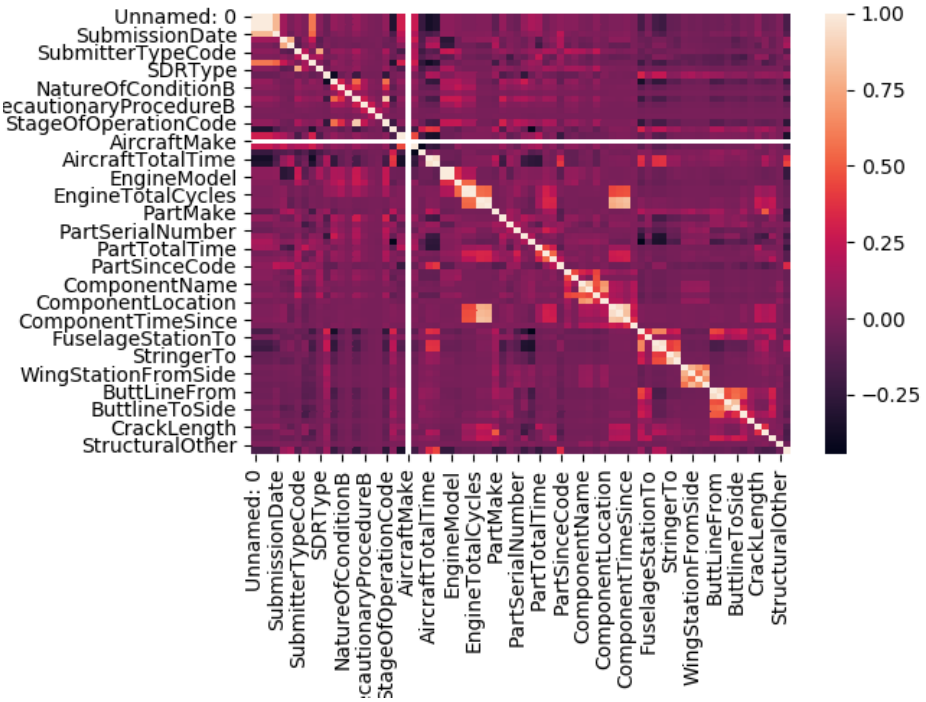


Fig. 13: Correlation Heatmap Crack Spearman

local minima to more successfully find a global minimum. In our tests with Rho, we trained and tested the same neural model configuration with Rho values ranging from 0.01 to 0.99 and incrementing by 0.01 each time. From these tests we recorded the AUC, Accuracy, Precision, Recall, Precision-Recall AUC, and F1 score. The tests showed that the model performed logarithmically better and peaked at 0.99, which is also the default value used by the H2O Deep Learning model. The same tests were conducted with the Epsilon parameter, but this time the values ranged from 1E-10 to 1E-5 and the power was incremented by 0.1 each time. Like the Rho tests, the model performance improved logarithmically until an approximate Epsilon value of 5E-7. The default value for Epsilon is 1E-8. These tests were also conducted for the L1 Regularization metric to reduce the possibility of overfitting while still maintaining adequate model performance regarding the same recorded variables from the above tests. With a default value of 1E-5, tests were conducted from 1E-6 to 1E-4 with L1 being incremented by 0.1 each time. These tests showed little to no improvement in the model's performance, and the conclusion is that it did not affect model performance in relation to our data.

5 Conclusion

This paper introduces a maintenance strategy based on the Federal Aviation Administration (FAA) data in the United States. The problem is addressed using neural networks. To verify the efficiency of the method, many experiments have been performed. We tested the method using different architectures, different activation functions, and different hidden layers. The method and the neural network models are general enough to be applied to any kind of output data for prediction. The neural network was tested again with important features, and the similar prediction results confirm that it successfully identified the redundant features.

There are many directions of future research continuing from the present work. One direction is to fine-tune the model to make it adaptive. Other possible directions of research are to develop a decision tree for maintenance and air-traffic control, to develop a real time machine learning based maintenance strategy, and to integrate the developed method with other aircraft data using transfer learning.

References

1. Yiwei, W. A. N. G., et al. "A cost driven predictive maintenance policy for structural airframe maintenance." Chinese Journal of Aeronautics 30.3: 1242-1257, 2017.
2. Maillart Lisa, M., Pollock Stephen, M., "Predictive maintenance techniques and their relevance to construction plant." Journal of Quality in Maintenance Engineering 4, no. 1: 25-37, 1998.
3. Wang, Fangyuan, et al. "Aircraft auxiliary power unit performance assessment and remaining useful life evaluation for predictive maintenance." Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 234.6: 804-816, 2020.
4. Tabernero A., and Batlle B., "Predictive Maintenance in hydrogenerators." Hydro 17, no. 6: 11-19, 2007.
5. Daily, Jim, and Jeff Peterson. "Predictive maintenance: How big data analysis can improve maintenance." In Supply Chain Integration Challenges in Commercial Aerospace, pp. 267-278. Springer, Cham, 2017.
6. McDonald, N., Corrigan, S., Daly, C., Cromie, S., "Safety management systems and safety culture in aircraft maintenance organisations," Safety Science, vol. 34(1), pp. 151–176, 2000.
7. Sriram, C., Haghani, A., "An optimization model for aircraft maintenance scheduling and re-assignment,"Transportation Research Part A: Policy and Practice, vol. 37(1), pp. 29–48, 2003.
8. Gopalan, R., Talluri, K. T., "The aircraft maintenance routing problem," Operations Research, vol. 46(2), pp. 161–292, 1998.
9. Endsley, M. R., Robertson, M. M., "Situation awareness in aircraft maintenance teams," International Journal of Industrial Ergonomics, vol. 26(2), pp. 301–325, 2000.
10. Kinnison, H. A., Siddiqui, T., "Aviation Maintenance Management," McGraw-Hill Professional, New York, New York, United States, 2012.
11. Hobbs, A., Williamson, A., "Associations between errors and contributing factors in aircraft maintenance" Human Factors, 45(2), 186–201, 2003.
12. De Crescenzio, F., Fantini, M., Persiani, F., Di Stefano, L., Azzari P., Salti, S., "Augmented reality for aircraft maintenance training and operations support," IEEE Computer Graphics and Applications, vol. 31(1), pp. 96–101, 2011.
13. Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., Chryssolouris, G., "An approach to operational aircraft maintenance planning," Decision Support Systems, Vol. 48(4), pp. 604–612, 2010.

14. Quinlan, J. R., "Induction of decision trees," Machine Learning, vol. 1(1), pp. 81–106, 1986.
15. Sethi, I. K., "Entropy nets: from decision trees to neural networks," Proceedings of the IEEE, vol. 78(10), pp. 1605–1613, 1990.
16. Roe, B. P., Yang, H.-J., Zhu, J., Liu, Y., Stancu, I., McGregor, G., "Boosted decision trees as an alternative to artificial neural networks for particle identification," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,vol. 543(2–3), pp. 577–584, 2005.
17. Tso, G. K. F., Yau, K. K. W., "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," Energy, vol. 32(9), pp. 1761–1768, 2007.
18. Xhemali, D., Hinde, C. J., Stone, R. G., "Naive bayes vs. decision trees vs. neural networks in the classification of training web pages," International Journal of Computer Science Issues, 4 (1), pp. 16–23, 2009.
19. Curram, S. P., Mingers, J., "Neural networks, decision tree induction and discriminant analysis: an empirical comparison," Journal of the Operational Research Society, vol. 45(4), pp. 440–450, 1994.
20. West, D., "Neural network credit scoring models," Computers & Operations Research, vol. 27(11-12), pp. 1131–1152, 2000.
21. Letham, B., Rudin, C., McCormick, T., Madigan, D., "Interpretable classifiers using rules and bayesian analysis: building a better stroke prediction model," Annals of Applied Statistics. 9(3), pp. 1350–1371, 2015.
22. Pal, P., Datta, R., Segev, A., "A neural net based prediction of sound pressure level for the design of Aerofoil," Proceedings of Fuzzy And Neural Computing Conference (FANCCO), 2019.
23. Barros, R, C., Basgalupp, M. P., Carvalho, A. C. P. L. F., Freitas, A. A., "A survey of evolutionary algorithms for decision-tree induction," IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews, vol. 42(3) pp. 291–312, 2012.
24. Deng, H., Runger, G., Tuv, E., "Bias of importance measures for multi-valued attributes and solutions. Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN). pp. 293–300, 2011.
25. Segev, A., "Adaptive Ontology Use for Crisis Knowledge Representation," Int. J. of Information Systems for Crisis Response and Management, 1(2), pp. 16–30, April-June 2009.
26. Segev, A., Jihan, S. H., "Context Ontology for Humanitarian Assistance in Crisis Response," Proceedings of the 10 th International ISCRAM Conference- T. Comes, F. Fiedrich, S. Fortier, J. Geldermann and T. Müller, eds. Baden-Baden, Germany, May 2013.
27. Pal, P., Datta, R., Segev, A., Yasinsac, A., "Condition Based Maintenance of Turbine and Compressor of a CODLAG Naval Propulsion System using Deep Neural Network." 6th International Conference on Artificial Intelligence and Applications (AIAP-2019), 2019.
28. Pal, P., Datta, R., Rajbansi, D., Segev, A., "A Neural Net Based Prediction of Sound Pressure Level for the Design of the Aerofoil." In Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing, pp. 105-112. Springer, Cham, 2019.
29. Segev, A., Datta, R., Benton, R., Curtis, D., "OINNIONN: outward inward neural network and inward outward neural network evolution." In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 79-80. 2019.
30. Vianna, W., O., L., Yoneyama, T., "Predictive maintenance optimization for aircraft redundant systems subjected to multiple wear profiles." IEEE Systems Journal 12, no. 2: 1170-1181. 2017.
31. Marx, D., A., Curtis R., G., "Human error in aircraft maintenance." Aviation psychology in practice: 87-104, 1994.
32. Wang, T., Lu-Han, C.,. "Psychological and physiological fatigue variation and fatigue factors in aircraft line maintenance crews." International Journal of Industrial Ergonomics 44, no. 1: 107-113, 2014.
33. Hobbs, A., Williamson, A., "Associations between errors and contributing factors in aircraft maintenance." Human factors 45, no. 2: 186-201, 2003.
34. Renato, D., M., Nascimento, C., L., "Prognostics of aircraft bleed valves using a SVM classification algorithm." In 2012 IEEE Aerospace Conference, pp. 1-8. IEEE, 2012.

35. Bonnie Lida, R., Hamblin, C., J., Chaparro, A., "Classification and analysis of errors reported in aircraft maintenance manuals." International Journal of Applied Aviation Studies 8, no. 2: 295, 2008.