# Utilizing Genetic Algorithm-Based Feature Selection to augment Rule-Extraction of a Three Layer Neural Network for SARS-COV1 classification

Venuk Ganearachchi
Research School of Computer Science,
The Australian National University,
Canberra ACT 2601,
u5695667@anu.edu.au

**Abstract.** A well cited drawback of a neural network is that it is difficult to explain their ability to derive accurate results, often referred to as their "black box" nature (Chakraborty et al, 2019). However, there are certain techniques proposed by authors such as Gedeon and Turner (1993) that attempt to do so. The goal of this paper is to contribute to this area of research by providing a framework for rapid approximation of rules-extraction, based on the SARS-COV1 dataset by Mendis, Gedeon and Koczy (2005). We first implement genetic algorithm-based feature selection to identify ideal features for modelling. Subsequently, we propose an alternative methodology to Gedeon and Turner's explanation procedure (1993), where we will use a decision tree classifier to extract rules from the neural network. Finally, the rules that are generated from the decision tree will be analyzed. To summarize the key findings of this paper, the decision tree modification allowed for rapid and accurate extraction of easily understandable rules. Additionally, the usage of genetic algorithm-based feature selection over correlation-based feature selection improved classification accuracy by approximately 2% in both models, and potentially reduced the propensity of the models to overfit to training data.

**Keywords:** Three Layer Neural Network, Characteristic Input method, SARS-COV1, Blood Pressure, Temperature, Decision Tree Model, Rules Generation, Correlation-based Feature selection, Genetic Algorithm-based Feature Selection, Logistic Regression

## 1.0 Introduction

### 1.1 Dataset

This paper was based on four datasets utilized by Mendis, Gedeon and Koczy (2005). The label for each dataset corresponds with the title of the dataset, either "high blood pressure", "normal", "pneumonia" or "sars". Each dataset has 23 features (excluding label) and 1000 samples. These features include temperature readings taken at four different times a day, diastolic and systolic blood pressure levels, and whether the individual has experienced nausea or abdominal pain. A sample/row of each dataset corresponds with values collected for that individual under the label attached to their dataset title. It is important to note that in these datasets some features may be binary, with a value of 0 or 1 representing complete non-activation and activation respectively. Whereas other features may be continuous in nature, depicting fuzzy membership values. The dataset is therefore normalized to a scale between 0 and 1. Information regarding appropriate pre-processing due to the nature of this data is available under sections 2.1 and 2.2 of this paper.

### 1.2 Problem Description

Mendis et al (2005) applied these datasets in the context of aggregation of fuzzy signatures, however, they will not be the focus of this specific paper. Instead, our goal is to create a framework for rapid generation of rules based on the output and representations of a neural network; as a result, we need to maximise classification accuracy to ensure the most accurate rules are generated. This underlying objective further informed the decision to use genetic algorithms for feature selection as they have been shown to produce better results in studies conducted by individuals such as Ernawati et al (2018) and Lin et al (2014). We employ a three-layer feed-forward neural network that can differentiate between people with SARS, pneumonia, high blood pressure and those without any medical condition (normal/healthy) based on the medical data depicted in section 1.1. An attempt is then made to extract rules that can explain the behavior of this neural network by creating an alternative framework based on the findings of Gedeon and Turner (1993) as well as others, by utilizing a decision tree classifier.

Using neural networks to classify various medical conditions is an active area of research. It has progressed rapidly from the usage of purely numerical data such as temperature and blood pressure readings, as is the case in Mendis et al (2005), to more advanced image classifications such as usage of convolutional neural network (CNN) in COVID-19 classification by Asmaa et al (2021). Usage of neural networks in this area and subsequent rule-extraction is an area worth exploring as it could lead to breakthroughs that allow for quicker diagnosis and treatment of patients, therefore benefitting humanity.

## 2.0 Methodology

### 2.1 Data Inspection

Firstly, the four datasets mentioned in section 1.1 were analysed for any discrepancies. Factors such as missing values, value ranges and outliers among other variations were investigated. Each individual dataset comprised of 24 features and 1000 samples. It was found that all datasets did not have any unusual "dirty" data which include items such as missing and non-standard representations of the same data (Kim et al, 2003). Additionally, there were no class imbalances with respect to the dataset, with an even split between all four labels. All variables were either discrete or continuous and normalized to a scale of between 0 and 1 indicating the "fuzziness" of the data. Therefore, no significant amount of preprocessing needed to be conducted.

An interesting characteristic of the datasets that informed further pre-processing decisions and feature selection is that each dataset was found to contain features that may be more relevant to each specific dataset. These relevant features will have continuous fuzzy values, whereas for other features, that are largely irrelevant, they will have discrete values of either 0 or 1. For example, as shown in the table below, the "HighBP" dataset mostly has 0s and sometimes 1s for temperature data, which is approximately 8 features which do not provide any meaningful contribution to the overall objective and may be redundant in that dataset. Whereas, for features relating to blood pressure, such as systolic and diastolic blood pressure readings, data is continuous in nature and more important. An example of this is provided in the summary statistics below:

**Table 1.** Summary statistics of a selection of features in the "HighBP" Dataset

| Feature | 8am_temp(S) | 12pm_temp(M) | 4pm_temp(H) | 8pm_temp(S) | BP_dias.(S) | BP_sys.(M) | BP_dias.(M) |
|---|---|---|---|---|---|---|---|
| Count | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| Mean | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.350271 | 0.351131 |
| Std. Dev | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.059822 | 0.059192 |
| Min | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.200000 | 0.200000 |
| 25% | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.306875 | 0.309475 |
| 50% | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.352200 | 0.349800 |
| 75% | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.391925 | 0.389125 |
| Max | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.500000 | 0.500000 |

In the table above, (S) denotes the fuzzy value "slight", (M) denotes "medium" or "moderate" and (H) denotes "High". We can also see some patterns emerge from this brief description. For example, if temperature is "slight" at 8am, as denoted by the value 1, it will also be "slight at 8pm. Furthermore, there appears to be a correlation between the BP_sys and BP_dias values (which represents systolic and diastolic blood pressure values respectively). Due to these reasons, we decided to conduct a correlation-based feature selection to determine if any features could be removed that did not add much value, this is detailed in section 2.3.

### 2.2 Data Pre-processing

As mentioned in section 2.1 there was no significant amount of "dirty" data, however, the data had to be prepared in other ways prior to it being fed into the neural network. Firstly, all individual datasets were given an additional column to depict their label, ranging from 0-3. Additionally, all columns were named, depending on their category, with respect to being "slight", "moderate" or "high". They were then concatenated into a single dataset comprising of 24 features and 4000 samples. This labelled, aggregate dataset was ready to be fed into the neural network. Upon the first iteration of the network, we discovered that the training accuracy and test accuracy was already 100%. This phenomenon of perfect classification accuracy was present across other models we employed such as logistic regression and decision tree. This could be due to the relatively simplistic nature of this dataset, where certain features more relevant to specific datasets, as described in section 2.1. We therefore decided to take several actions such as conducting correlation-based

feature selection to remove irrelevant features and improve run-time of the neural network. To increase the complexity of our task and relevance of our framework to real-world imperfect data, we also took model-specific actions such as introducing a dropout layer in the neural network, these actions will be detailed in their individual sections.

### 2.3 Correlation-based Feature Selection

A correlation-based feature selection method was initially chosen to reduce the number of irrelevant features under consideration. There are many variants of correlation-based feature selection, however, the main premise is that good feature sets contain features that are correlated with the class variable, yet uncorrelated with each other (Hall, 1999). Therefore, we omit the features which do not fulfill these criteria. A pairwise correlation analysis between each feature revealed that a substantial quantity of features is either strongly positively or inversely correlated with one another. This can be seen in Figure 1 below, which uses a heatmap and Pearson's method to calculate the correlation values. The interesting values in question are primarily clustered along the diagonal axis:

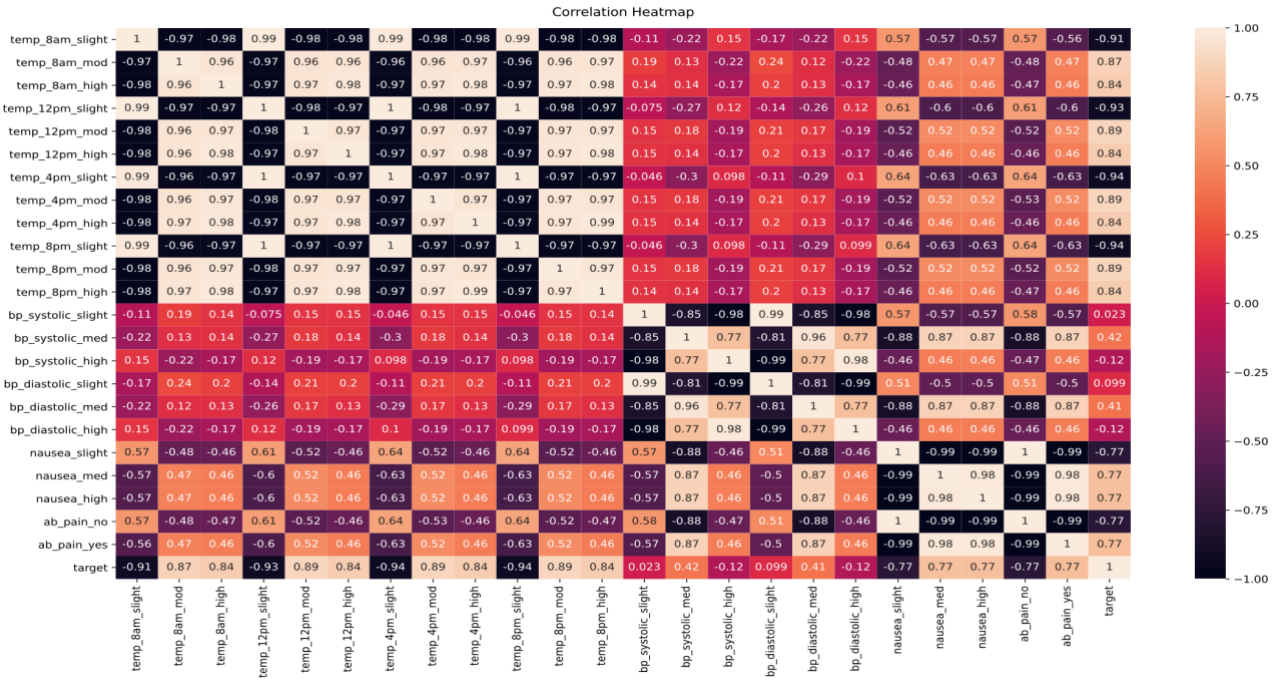


**Fig. 1.** Correlation heatmap between all features: note the lighter values, clustered around the diagonal axis

Temperature taken across each of the four different time readings appear to be highly correlated with one another. Temperature taken at 8a.m for example in the "high" category depicted above, had a 0.99 correlation with the same temperature category readings taken at 12p.m, 4p.m and 8p.m. This finding is consistent for systolic and diastolic blood pressure as well, where "high" systolic blood pressure and "high" diastolic blood pressure had a Pearson's correlation of 0.98.

Furthermore, certain variables are nearly perfectly inversely correlated, depicted by a black color in the above heatmap. To remove unnecessary information that do not provide any added value and increase the complexity of the classification task, all redundant features were removed. Only the "high" values were kept for temperature; the mean was taken for all readings over the differing time intervals. The "high" values were kept as they were strongly correlated with "mod" values and strongly inversely correlated with "slight", therefore a substantial amount of information is retained in this one feature. Similar reasoning in terms of correlation were applied for blood pressure, abdominal pain, and nausea. Therefore, in the final dataset to be inserted into the neural network, only four features remained from the initial 23. This reduction in dimensionality, due to lost information, should theoretically have made the classification task substantially harder. However, this did not greatly impact classification results.

### 2.4 Genetic Algorithm based Feature Selection

Using correlation-based feature selection, we were able to identify correlated features and aggregate them into four main features, this will improve run-time, however, a large amount of fuzzy information is lost. Such a feature selection technique may also result in the issue of overfitting to training data. Upon further investigation, we discovered that using genetic algorithm-based feature selection would be able to minimize these issues and improve classification accuracy, and by extension the accuracy of rules that are generated.

Genetic algorithms are an optimization technique that involve moving from one population of "chromosomes" to a new population through a process of "natural selection" in conjunction with genetics-inspired operators of crossover, mutation, and inversion (Mitchell, 1999 p.3). In our scenario, the idea of "chromosomes" represents a string of 1's and 0's that correspond with our optimal subset of features, where 1 represents inclusion of a feature and 0 represents its absence. The most important aspects of implementing a genetic algorithm involve selection of chromosome encoding, fitness function evaluation, selection mechanisms, genetic operators and stopping criterion (Babatunde et al, 2014).

To implement this feature selection technique, we used the DEAP evolutionary computation framework (DEAP, 2021). As mentioned above, the selection mask or binary vector which indicates included and absent features can be regarded as the "chromosome". To begin, we first initialize the population of potential solutions randomly. During each "generation" the best solutions in terms of the fitness function are kept. The fitness function here refers to the performance of a logistic regression model where we look at classification accuracy as the fitness value. Therefore, feature subsets that provide the highest classification accuracy in terms of this model will be retained past each subsequent generation. As a reproduction method for the populations, we used a tournament selection style approach (Yang, 1997) but with the DEAP "Hall of Fame" module that keeps track of the optimal subset ordered in terms of fitness value that has existed throughout the evolutionary process. We also track the test and validation accuracy of all subset variations in the "Hall of Fame". To increase diversity, each generated offspring feature subset will also go through a mutation process with a probability of 0.05. Differing mutation rates were experimented with, however, due to perfect classification accuracy under the logistic regression model, we were unable to vary accuracy significantly.

As noted in figure 2 below, multiple "optimal" feature subsets were discovered with perfect classification accuracy, which is not ideal as we can only use a single subset for subsequent modelling.

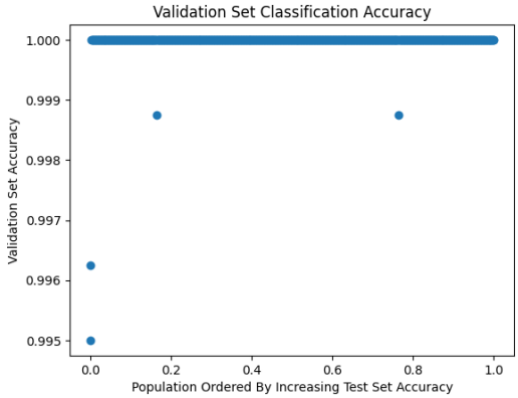


**Fig. 2.** Validation Set Classification accuracy for differing feature subsets

To identify a single subset, 15 feature subsets were randomly chosen from the population and the neural network model was re-trained with a dropout layer and the chosen subsets as inputs. Classification accuracy for each iteration was recorded and the feature subset with the highest accuracy was chosen. The chosen feature subset can be found in table 2 and corresponding accuracies for each feature selection technique in table 3.

**Table 2.** "Optimal" feature subset chosen under Genetic algorithm-based Feature Selection

| | | | | | |
|---|---|---|---|---|---|
| temp_8am_high | temp_12pm_slight | temp_4pm_mod | temp_8pm_slight | temp_8pm_mod | bp_systolic_slight |
| bp_systolic_med | bp_systolic_high | bp_diastolic_med | bp_diastolic_high | nausea_slight | nausea_high |
| ab_pain_yes | | | | | |

This feature subset, which contains 13 features, is substantially reduced from the original dataset which contained 23 features. Based on figure 1, it is possible to determine that under this subset, quite a few of the highly correlated features have been removed. Since this subset contains more information than selected under correlation-based feature selection, it may be more robust towards reducing the probability of overfitting, hence we will use this feature subset throughout this paper to determine rule-extraction. The following results have been recorded under each of the feature selection techniques that were utilised:

**Table 3.** Average classification accuracy (3 iterations) under each Feature Selection method (% values)

| **Feature Selection (FS) Method** | **Decision Tree (10-fold avg.)** | **Neural Network (Train/Test)** |
|---|---|---|
| No FS (23 Features) | 100 | 90.97/91.23 |
| Correlation-based FS (4 Features) | 91.09 | 90.92/91.06 |

| | | |
|---|---|---|
| Genetic-based FS (13 features) | 91.98 | 92.62/93.07 |

It is important to note that the above values occur with optimal hyperparameter settings for each respective model. These hyperparameters will be further discussed in section 3 and section 4 of this paper. The neural network has a lower score than the decision tree when there is no feature selection as a dropout layer has been applied to reduce the probability of overfitting as described in section 4.1. This is also the reason for comparatively lower accuracy under each feature selection process.

### 2.5 Baseline Neural Network Architecture

In terms of a baseline architecture for the classification task, a simple fully connected three-layer neural network architecture was used with an input layer, hidden layer, and output layer. The input layer consists of 13 neurons, representing the input features. A single hidden layer with 20 hidden neurons was used, as well as an output layer with 4 neurons, representing the target variables. A tanh activation function was used, other activation functions such as SoftMax were also trialed, however, these led to poorer results than expected. This is detailed further in section 4.1. Holdout validation with an 80-20 train/test split was used. Model training and testing as well as hyperparameters will be discussed further in section 3 and 4 of this paper.

### 2.6 Performance Metrics

To evaluate performance of the neural network model, we use test and training accuracy conducted over multiple iterations specified in section 4.3. For the decision tree, we use K-fold cross validation and use the average accuracy of all iterations. A confusion matrix was used to monitor the ratio of false positives, true positives, and the like. However, since the classification results constantly exceeded 90+ % with few misclassifications, it did not make sense to report on and use precision, recall or other evaluation metrics as the main metric to determine performance in the model. Furthermore, the rules that are extracted from the neural network with the decision tree will be analyzed to evaluate whether they are consistent with the literature.

### 2.7 Characteristic Input and Decision Tree Intuition

The proposed network architecture can be used in conjunction with the pre-processing and preparation methods to create a classifier that is able to adequately differentiate between the various class variables. However, it does not answer a key question: specifically, how the network has made the prediction. A set of rules that can express the knowledge learnt by the neural network would be very beneficial, especially in the medical domain (Dancey et al, 2010). One method widely described in the literature is to prune the network to a minimal size (Siestma and Dow, 1991), however, extreme pruning may reduce the robustness of the neural network solution (Gedeon and Turner, 1993).

Using a causal index and characteristic input method is another alternative to creating rules (Gedeon and Turner, 1993). This paper proposes an alternative to the explanation procedure and mechanisms introduced by Gedeon and Turner (1993), with certain modifications to enhance the generation of accurate rules for this specific dataset. The methodology employed by Gedeon and Turner uses a feedforward neural network and employs four main steps. Firstly, comparing the input pattern to certain so called “characteristic” patterns, which are a set of patterns that can trigger the target variable, corresponding to “characteristic ON” and vice versa for “characteristic OFF” patterns, which do not turn the output on. Secondly, providing input patterns considered ‘important’ for the current network label, as well as their corresponding values in the characteristic patterns. Thirdly, producing a set of rules which are subsequently evaluated. Finally, in step 4, the neural network is given the next most likely output.

There is a strong possibility that employing steps 1 and 2 may not be effective for this specific dataset as there is little differentiation between the high blood pressure and SARS outputs in terms of their corresponding inputs. In other words, there may be overlap in the characteristic input patterns for SARS and high blood pressure as discovered during exploratory data analysis and accentuated further by the decision tree classifier. It may be easier to rapidly approximate these rules by instead using a decision tree classifier, as we have done. Therefore, we do not follow Gedeon and Turner’s explanation procedure (1993) exactly; instead, we forego step 1 and 2 due to issues with this dataset that we have documented throughout this paper. Our method is perhaps more apt for simplistic datasets such as SARS which can achieve high classification accuracy with most models.

The decision tree classifier uses the CART algorithm to segment the data into binary sub-trees based on splitting criteria such as Gini index which is the impurity measure that is used in this paper. The aim of the decision tree is to continuously select the “best” split based on the impurity measure starting from the root node and is subsequently repeated until stopping rules are satisfied (Pitombo et al, 2017). An advantage of using decision trees is that they may

be less influenced by issues with the dataset, primarily regarding distribution and encoding of the variables. In addition to more rapidly being able to generate accurate rules, a decision tree classifier may be more suitable to the objective of this dataset due to these reasons.

To accommodate this objective, we first train a three-layer neural network and provide its predictions to the decision tree as an input, instead of using the ground truth labels. The decision tree classifier will then be trained and used to extract rules, which will subsequently be evaluated, which is step 3 of the above framework. Step 4 of the above process is to produce the network's most likely output, which can be done by counting the number of rules that have been satisfied for a specific label. To clarify the efficacy of the rules that have been extracted, we will remove any features which do not appear to have any effect on the neural network and re-train both the neural network and the decision tree classifier without the specific feature(s), to confirm that there is no effect of the feature(s) on classification accuracy.

## 3.0 Implementation

### 3.1 Experiment settings and hyperparameters – Neural Network

PyTorch version 1.8.1 was used to implement the neural network, the hardware basis for this paper was an Intel i9-9980Hk Processor, running on MacOs Version 11.2.3. The model was trained with Stochastic Gradient Descent (SGD) optimizer and a learning rate of 0.03. We use Stochastic Gradient Descent (SGD) over alternative optimizers like Adam due to two main reasons. Firstly, optimizers that use adaptive learning rates generalize more poorly compared to SGD (Wang et al, 2018), which allows SGD to get past local minima more effectively. Secondly, the dataset is quite simple and has certain problems which are detailed in section 2, which may allow Adam to achieve higher classification scores, but also be more prone to overfitting.

The learning rate was trialled in a naïve manner, in several increments once the rest of the hyperparameters were finalized. The iteration number for the training set is 1000 epochs. Furthermore, we use dropout regularization and set the p value to 0.25. Dropout was introduced to reduce the risk of overfitting as the train/test accuracy is quite high for this dataset. The dataset was initially randomly shuffled. Subsequently, holdout validation was used, where the data was split into 80% training data for the model and 20% testing data. The model architecture is given above in section 2.4

### 3.2 Experiment settings and Hyperparameters – Decision Tree

Sci-kit learn version 0.24.1 was used to implement the decision tree model. The main hyperparameter of note in the Decision Tree model is the splitting criteria which is crucial in classifying objects (Raileanu and Stoffel, 2004). Both Gini index and information gain were compared, however, there was no significant difference found. Therefore, we use Gini index as it is less computationally expensive, since it does not have a logarithmic component like information gain does (Raileanu and Stoffel, 2004). Furthermore, the max depth set for each tree was 3, which is a standard amount to generate comprehensible rules. Additionally, we set a seed of 1 to prevent the random generation issue discussed in section 3.4. Finally, K-fold cross validation is used in the model, where K is set to 10.

## 4.0 Results and Discussion

### 4.1 Baseline Neural Network Model results

During the first iteration of the model, a train/test score of 100% accuracy was recorded. Hence the reason why correlation-based feature selection was used to reduce the number of features, as it was discovered that there were several features which were highly correlated with one another and could be considered redundant. However, reducing the number of features alone may not be sufficient to prevent the risk of overfitting. Due to this issue, dropout regularization was used with a p value of 0.25. With a dropout layer, it was possible to reduce the accuracy score to less than 100% and prevent risk of overfitting to a certain extent. As stated above, SGD was also chosen over other well-suited optimizers such as Adam (Kingma and Ba, 2014) to reduce the risk of overfitting and generalize more efficiently.

To decide the activation function and hidden neuron settings, we run the model 3 times and take the average test accuracy. Several activation functions and hidden neuron settings have been investigated; they are detailed below, along with the hyperparameter settings discussed in section 3.1 above. Note: *n* refers to the number of hidden neurons

**Table 4.** Test Accuracy of the Baseline Neural Network model (% values)

| *n* | Sigmoid | Tanh | ReLu | Softmax |
|---|---|---|---|---|
| 10 | 75.71 | 89.51 | 78.20 | 59.32 |
| 20 | 79.61 | **92.80** | 92.10 | 57.87 |
| 30 | 80.36 | 92.06 | 91.46 | 53.36 |
| 40 | 86.10 | 90.76 | 89.34 | 55.71 |
| 50 | 86.03 | 89.52 | 91.21 | 50.33 |

As can be seen above, a Tanh activation function with 20 hidden neurons yielded the highest test accuracy result. This may be due to the notion that an output using Tanh center around 0 rather than 0.5, which is the case with activation functions like sigmoid. This is also consistent with findings from LeCun et al (2012) who determined that convergence occurs at a faster pace when the average of each input variable over the training set is closer to 0. Since increasing the number of hidden neurons from 20 does not increase, but rather decreases performance due to overfitting, we keep the number of hidden neurons at 20. Cross-entropy loss was used as the loss function as it is a widely accepted criterion for multiclass classification (Ghiasi-Shirazi, 2019). The diagrams below highlight two key metrics of the created neural network, steadily decreasing loss and increasing accuracy which stabilizes at roughly the 1000$^{th}$ epoch, during convergence.

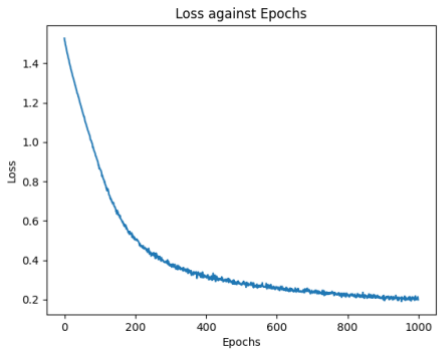


**Fig 3.** Measuring loss against epochs

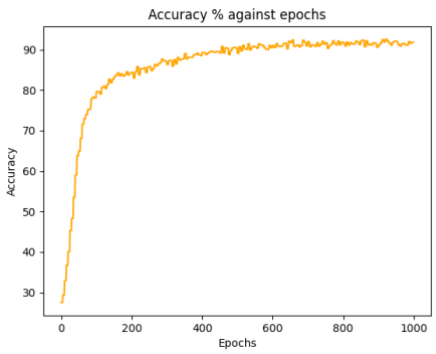


**Fig 4.** Measuring accuracy against epochs

### 4.2 Decision Tree Rule Extraction

Now that the neural network has been trained, we forego steps 1 and 2 of the Gedeon and Turner explanation procedure (Gedeon and Turner, 1993) and produce a set of rules using the decision tree classifier. This is done by firstly allocating the same training data that is used by the neural network to the decision tree. Subsequently, the predicted output is provided from the neural network to the decision tree classifier as labels, which is an important distinction to note.

As stated in 3.2, we use K-fold cross validation for the decision tree with K set to 10, the following results were received during each iteration with an average classification accuracy of 0.918, they have been rounded to 3 d.p.

**Table 5.** Accuracy of each "fold" of the Decision Tree Classifier (% values)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.909 | 0.931 | 0.921 | 0.931 | 0.905 | 0.927 | 0.905 | 0.890 | 0.921 | 0.937 |

This average classification accuracy is lower than the result achieved by Gedeon and Turner (1993) which was 94%, this may be due to several reasons such as differing datasets, differing models and possibly the introduction of the dropout layer in the neural network. Unfortunately, it was not possible to compare against other papers as the literature on this dataset is scarce. Mendis et al (2005), which this paper was based on do not provide their classification results to benchmark against.

A limitation of note while using a decision tree is the inherent use of a random number generator to generate rules, so each iteration of the model will produce slightly different rules, even though they share similar characteristics. The random state of the decision tree model was kept fixed to prevent this from being a significant issue. The rules that were generated by the iteration with the highest classification accuracy are provided below:

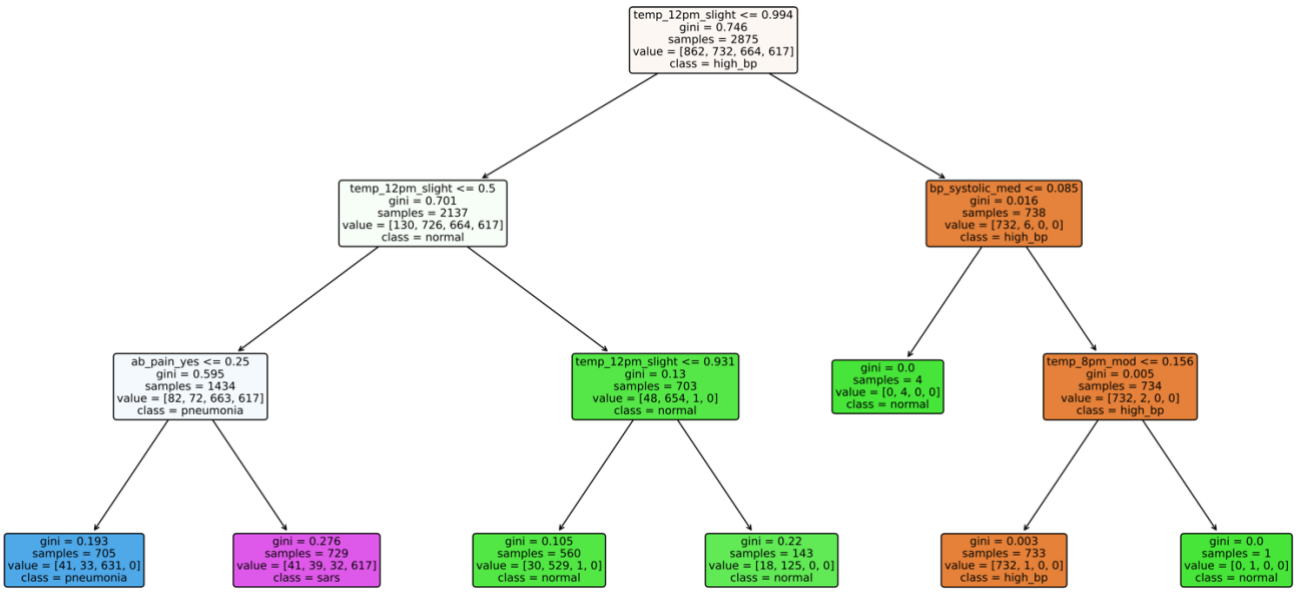


**Fig 5** Decision Tree visualization with 13 features chosen by GAFS (Iteration with 93.7% Accuracy)

Note: In the above "value" section, the list of numbers is indexed as follows: position 1 refers to the quantity of "high blood pressure" labels, position 2 refers to quantity of "normal" labels, position 3 refers to quantity of "pneumonia" labels and position 4 refers to quantity of "sars" labels. An arrow to the left of a decision node represents a TRUE response to the condition in the decision node, for example the left child of the root decision node with "temp_12_pm_slight<=0.994" is "temp12_pm_slight<=0.5" where "temp_12pm_slight<=0.994" is true. The opposite is true for arrows to the right of a decision node where it will represent a FALSE response. Gini index essentially calculates the "impurity" of the node, and whether all training instances belong to the same class (Raileanu and Stoffel, 2004).

### 4.3 Explaining the generated rules

We provide a compressed view of the rules that were extracted using the decision tree below, in a format similar to the rule format presented by Gedeon and Turner (1993), substituting on for TRUE and off for FALSE.

| Pattern | Rule Set |
|---|---|
| TRUE BP | (temp_12pm_slight>0.994) AND (bp_systolic_med>0.085) AND (temp_8pm_mod<=0.156) |
| TRUE SARS | (temp_12pm_slight<=0.5) AND (ab_pain_yes>0.25) |
| TRUE Pneumonia | (temp_12pm_slight<=0.5) AND (ab_pain_yes<=0.25) |
| TRUE Normal | (temp_12pm_slight>0.5) AND (temp_12pm_slight<=0.931) |

IF (temp_12pm_slight<=0.5) AND (ab_pain_yes>0.25) THEN SARS

Note, if we increase the max depth of the decision tree, we can extract more granular detail regarding the separation of values. However, this creates the drawback of being much more difficult to understand in a diagrammatic form as the number of rules that are generated are exponential in nature. For example, if we were to increase max depth to 4, we could generate the following set of rules for TRUE SARS:

IF (temp_12pm_slight<=0.99) AND (temp_12pm_slights<=0.50) AND (ab_pain_yes>0.25) AND (by_systolic_high>0.54) THEN SARS

The symptoms showcased in the rule set appear to be consistent with symptoms for each specific medical condition. In the case of SARS, a value less than 0.50 for temp_12pm_slight indicates that the temperature will be high due to the inverse nature of the data for slight and high values. Additionally, abdominal pain and high blood pressure values are high, these symptoms are consistent with SARS-COV1. However, it is important to verify these results with a domain expert to determine true accuracy.

While evaluating these rules, we found that the features “nausea_slight” and “nausea_high” did not prominently feature among the rules that were generated with a max depth of 3 in the decision tree. As mentioned above, to evaluate that rules that are being generated are accurate, we set all features that do not contribute to the classification to 0. Therefore, a new dataset was created without these features. After re-training the neural network and the decision tree classifier on the new dataset, it was discovered that average accuracy of the model did not fall substantially (less than 0.5%). This confirms that “nausea_high” and “nausea_slight” were negligible factors and that the rules generated are somewhat precise as they were able to point out this flaw in the model.

### 4.4 Limitations of this paper

A flaw of this approach is that the decision tree classifier, despite having an average classification accuracy of 0.918, which is comparable numerically to the neural networks highest result of 0.928, is proportionately lower. This is because we feed in the labels of the predicted output from the neural network, which are inaccurate to a certain degree and contain misclassifications, directly into the decision tree classifier as opposed to using the ground truth labels. The accuracy from the decision tree is then calculated based on both the correct and incorrect predicted labels from the neural network.

Additionally, the decision tree, will not know the internal representations of the neural network, and can only approximate the output of the neural network rather than its internal structure. This results in our method not being entirely deterministic. As such it is possible to argue that our method cannot completely explain the decision making of a neural network. However, we believe that for simplistic datasets such as the SARS dataset that we have used, where a limited quantity of features can directly predict class output; our method is suitable to enable rapid and accurate rules extraction. For a more complicated dataset, where there is clear differentiation in the characteristic inputs, Gedeon and Turner’s explanation procedure (1993) as well as other pruning methods may have been more relevant and potentially could have provided better results.

Furthermore, the dataset is too simplistic to generate a vast quantity of meaningful rules from them. As explained in section 2.2, a substantial portion of the data is redundant and does not contribute greatly to the efficacy of the model, which led to the reduction in features from 23 to 4. Evaluation of the rules generated in 3.5 further led to the discovery that an additional feature “nausea_high” was again not required. As this is the case, it is possible to conclude that the insights generated from the rules-generation were not as detailed and profound as they could have been, with a more varied and detailed dataset.

Finally, due to the random nature of decision tree classifiers, the decision trees can generate several rules. These rules are largely the same, however, if run with enough iterations, there could be a possibility of generating drastically different rules. It may be important to find a way to merge all these rules into the results of a single model.

## 5.0 Conclusion and Future work

This paper commenced by discussing a classification problem, specifically the differentiation of SARS-COV1 from pneumonia, high blood pressure, and healthy individuals. A three-layer neural network was proposed to solve this problem. We first conducted an exploratory data analysis which involved comparing both correlation-based and genetic algorithm-based feature selection methods. It was found that genetic algorithm-based feature selection could provide an optimal feature subset that provided approximately 2% greater accuracy than when no feature selection or correlation-based feature selection was applied.

We then discussed the characteristic input method by Gedeon and Turner (1993) for rule extraction, proposing an alternative method that uses a decision tree classifier to move directly to step 3 of their explanation procedure. It was found that the neural network could classify the labels with approximately 92% accuracy. Once the neural network predicted labels had been fed to the classifier, it was possible to train the decision tree classifier with close to 91% accuracy (albeit the ground truth comparison issue discussed in 3.6). The rules that were generated seemed largely accurate, although they need to be inspected by a domain expert for confirmation.

Unfortunately, we are not able to compare our models and results against those documented by Mendis et al (2005), as they do not detail classification accuracy. They also have a different goal than rule-extraction, which is the goal of this paper, so it is not possible to compare extracted rules for this specific dataset across other papers in the literature.

Ultimately, we conclude by stating that our method was able to achieve rapid approximation of rules for this specific dataset, due to its simplicity, however, it has several limitations that prevent it from being used in more complex

datasets. These limitations mean that for more complex datasets, alternative frameworks like Gedeon and Turner's explanation procedure (1993) and other pruning methods could prove to be more beneficial.

Upon conducting the methodology outlined in this paper, several potential areas of future work were identified. An important area which could potentially be useful is the investigation of merging rulesets from multiple iterations of the decision tree classifier to identify the most effective and accurate ruleset. Currently, research into this area is lacking and it was not possible to immediately find any paper proposing an effective method to solve this problem. Investigating and utilizing such a method will allow for an extension to the results of this paper and create even more accurate rules. These rules may have a greater probability of uncovering previously unknown insights into the data.

## References


1. Asmaa, A., Abdelsamea, M.M. & Gaber, M.M. 2021, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network", *Applied Intelligence,* vol. 51, no. 2, pp. 854-864.
2. Babatunde, O.H., Armstrong, L., Leng, J. and Diepeveen, D., 2014. *A genetic algorithm-based feature selection.*
3. Chakraborty, M., Biswas, S.K. & Purkayastha, B. Rule Extraction from Neural Network Using Input Data Ranges Recursively.*New Gener. Comput.* ***37,*** *67–96 (2019)*
4. Cira Souza Pitombo, Andreza Dornelas de Souza, Anabele Lindner, *Comparing decision tree algorithms to estimate intercity trip distribution*, Transportation Research Part C: Emerging Technologies, Volume 77,2017,Pages 16-32,ISSN 0968-090X,https://doi.org/10.1016/j.trc.2017.01.009.
5. D. Dancey, Z. A. Bandar and D. McLean, "Rule extraction from neural networks for medical domains," *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1-8, doi: 10.1109/IJCNN.2010.5596693.
6. Deap.readthedocs.io. 2021. *DEAP documentation — DEAP 1.3.1 documentation*. [online] Available at: <https://deap.readthedocs.io/en/master/> [Accessed 30 May 2021].
7. Engelbrecht A., Viktor H. (1999) "Rule improvement through decision boundary detection using sensitivity analysis". In: Mira J., Sánchez-Andrés J.V. (eds) *Engineering Applications of Bio-Inspired Artificial Neural Networks*. IWANN 1999. Lecture Notes in Computer Science, vol 1607. Springer, Berlin, Heidelberg
8. Ghiasi-Shirazi, K. Competitive Cross-Entropy Loss: A study on training single-layer neural networks for solving nonlinearly separable classification problems *Neural Processing Letters* 50, 1115-1122 2019
9. Hall, M. (1999). Correlation based feature selection for machine learning. Doctoral dissertation, University of Waikato, Dept. of Computer Science.
10. K. Hara and K. Nakayamma, "Comparison of activation functions in multilayer neural network for pattern classification," *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, 1994, pp. 2997-3002 vol.5, doi: 10.1109/ICNN.1994.374710.
11. Kim, W., Choi, B.J., Hong, E.K., Kim, S.K. and Lee, D., 2003. A taxonomy of dirty data. *Data mining and knowledge discovery*, *7*(1), pp.81-99.
12. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
13. LeCun, Y.A., Bottou, L., Orr, G.B. and Müller, K.R., 2012. Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9-48). Springer, Berlin, Heidelberg.
14. Lin, C., Chen, H. & Wu, Y. 2014, "Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection*", Expert systems with applications, vol. 41, no. 15,* pp. 6611-6621.
15. M. R. A. Iqbal, "Eclectic rule extraction from Neural Networks using aggregated Decision Trees," *2012 7th International Conference on Electrical and Computer Engineering*, 2012, pp. 129-132, doi: 10.1109/ICECE.2012.6471502.
16. Melanie M. An Introduction to Genetic Algorithms A Bradford Book The MIT Press, 1999.
17. Mendis, B. S., Gedeon, T. D., & Koczy, L. T. (2005). "Investigation of aggregation in fuzzy signatures," in Proceedings, *3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore*.
18. N. Shahadat, B. Rahman, F. Ahmed and F. Anwar, "Dropout effect on probabilistic neural network," *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017, pp. 217-222, doi: 10.1109/ECACE.2017.7912908.
19. Raileanu, L.E., Stoffel, K. Theoretical Comparison between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence* **41,** 77–93 (2004). https://doiorg.virtual.anu.edu.au/10.1023/B:AMAI.0000018580.96245.c6
20. Sietsma, J & Dow, RF**,** "Creating Artificial Neural Networks That Generalize," *Neural Networks,* vol. 4, pp. 67-79, 1991.
21. S. Ernawati, E. R. Yulia, Frieyadie and Samudi, "Implementation of The Naïve Bayes Algorithm with Feature Selection using Genetic Algorithm for Sentiment Review Analysis of Fashion Online Companies," *2018 6th International Conference on Cyber and IT Service Management (CITSM),* 2018, pp. 1-5, doi: 10.1109/CITSM.2018.8674286.
22. T Hailesilassie (2016) "Rule Extraction Algorithm for Deep Neural Networks: A review" (IJCSIS) *International Journal of Computer Science and Information Security, Vol. 14, No. 7, July 2016*
23. T. D. Gedeon and H. S. Turner, "Explaining student grades predicted by a neural network," Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), 1993, pp. 609-612 vol.1, doi: 10.1109/IJCNN.1993.713989.
24. Wang, Y., Zhou, P., & Zhong, W. (2018). *An optimization strategy based on hybrid algorithm of adam and SGD*. Les Ulis: EDP Sciences. doi:http://dx.doi.org.virtual.anu.edu.au/10.1051/matecconf/201823203007
25. Yang, J. and Soh, C.K., 1997. Structural optimization by genetic algorithms with tournament selection. *Journal of Computing in Civil Engineering*, *11*(3), pp.195-200.