Stress recognition with EEG Signals Using Neural Network Rule Extractions and a Genetic Algorithm for Feature Selection

Eric Pan

Research School of Computer Science, Australian National University
u6409977@anu.edu.au

Abstract. Stress is a natural human response to external conditions which have been studied for a long time. Since prolonged periods of stress can cause health deterioration, it is important for researchers to understand and improve at detecting it. One method for stress recognition is through the use of the electroencephalogram (EEG). Neural networks are not easily interpretable due to the complex nature of their model architectures. Moreover, pattern recognition tasks benefit greatly from feature selection where the dataset involved is of high dimensionality. Therefore, aims of this paper are threefold – to examine the effectiveness of simple Artificial Neural Networks in predicting stress, interpret the behavior of the neural network using rule extraction, and finally explore the viability of a genetic algorithm for feature selection. Using summary statistics from the EEG, a three-layer ANN is able to achieve 93.8% accuracy and using a sensitivity analysis approach along with characteristic patterns for rule extraction, it is able to explain the ANN to a good degree and thus improve interpretability. Adding feature selection with a genetic algorithm improves average accuracy achieved by the ANN to 95.4%.

Keywords: Stress Detection, Artificial Neural Network, EEG, Rule Extraction, Sensitivity Analysis, Neural Network Explainability, Evolutionary Algorithm, Genetic Algorithm, Feature Selection

1 Introduction

1.1 Background

Stress exists for humans in all domains, whether it is work, study, or otherwise situations with external pressures. The most commonly discussed source of stress is time pressure, when the work demanded of us needs to be completed within an urgent timeframe. However, there are many other forms of stress, all of which depend on psychological factors and induce physiological responses [3]. The tolerance threshold for feeling stress and the reactions triggered vary for each individual, therefore it is imperative to have some method of measurement that can objectively quantify important symptoms or indicators of stress. This is especially the case where specialist psychologist or mental health facilities are not available, where expert judgement can be exercised to identify stress [4]. One tool for objective measure is the electroencephalogram (EEG). The EEG is a wearable headset device with multiple sensors that detect electrical brain activity using electrodes (Fig 1). The combinations of EEG signals maybe used to detect core indicators of stress.

To effectively use EEG signal data for stress detection, an Artificial Neural Network (ANN) is employed. ANNs are good function approximators that also excel at simple classification tasks. Despite being able to achieve high performance and good results in terms of predictions and classifications, many domain experts are skeptical to use them to make highly crucial decisions that have significant ramifications if done wrong [7]. This is because the knowledge represented in the numerical weights of ANNs are difficult to make sense of. Although the predictions it makes are very good, unless a human can logically interpret its actions in the context of the domain, many experts cannot justify using it to make decisions. This is particularly prevalent in domains with high ethical stakes or where explanations must be given to key stakeholders. Examples include creditworthiness checks and medical decisions. This disadvantage of ANNs is why many algorithms have been developed to extract rules and behavior patterns from neural networks, which are easy for humans to understand [6]. If this can be done effectively, it will mean that we can make full use of the power of neural networks and at the same time be able to sufficiently justify its decisions, overcoming the interpretability issue.

Another prominent issue in the world of data mining and machine learning is dealing with high dimensional data. As the number of predictors, or features, increases, the total volume of the input space increases incredibly fast. This gives rise to the problem of data sparsity, and it becomes difficult for machine learning models to generalize and learn useful patterns. Neural networks are no exception. This is known as the curse of dimensionality, and is prevalent in tasks involving EEG data since there are so many channels and sources of information [9]. Many of the features may also be non-informative and negatively impact model performance. Feature selection therefore becomes extremely important when working with high dimensional data.

In this paper, a neural network will be built using EEG signal data, in order to predict whether a participant is feeling stressed or calm. This paper will combine the three critical issues above together. We will be exploring the effectiveness

of the EEG headset as a detection for stress, leveraging the predictive ability of neural networks, while deducing meaningful rules that compress the neural network's behavior into a digestible, explainable series of decisions. To select only the useful features, genetic algorithms (GA) are among the methods that can be configured freely with parameters to improve efficacy [10]. Therefore, we will utilize a GA to remove redundant attributes and complement the ANN. The goals of the paper are to demonstrate the merits of using wearable devices to learn about stress, to provide confidence that neural networks can be explained intuitively, and to show how the right kind of input processing can dramatically yield better results.

1.2 EEG signals

The EEG signals used in this paper come from a 14-channel headset. Each channel detects activity from a different part of the brain. These channels are: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. These 14 channels are named as power spectral density of all 14 channels (PSD14). Figure 1 is a visualisation of this device, along with the location on the scalp each channel attaches to. Such contact-based devices are common amongst studies involving physiological reactions, such as classifying emotions [1].

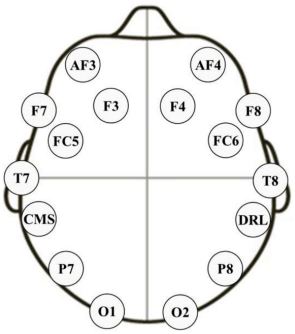

Fig. 1. A 14-channel EEG headset.

1.3 Proposed task

The goal is to classify the participants as either stressed or calm using an ANN, given the EEG signals. The goal of the experiment is the same as *Thermal Super-Pixels for Bimodal Stress Recognition* [2]. The difference is we use a traditional contact-based detection system.

We also propose a rule extraction method for the ANN, which locates decision boundaries for the EEG input parameters. Through a sensitivity analysis approach, rules are extracted based on decision boundaries found in the continuous input space [7]. But since the search of potential decision boundaries for each input parameter can be quite computationally expensive, characteristic input patterns are utilized to compress the training set and represent the unique observations [5] during rule extraction. These rules are then used for stress recognition on the original dataset and evaluated with comparison to the original ANN.

Lastly, we also incorporate feature selection to remove redundant attributes using a genetic algorithm.

The experiment will be conducted in two phases. The first phase includes building the optimal ANN architecture for this EEG dataset, manually selecting features qualitatively, and then implementing the sensitivity analysis-based rule extraction for the network. The second part will be identical to the first except the optimal features are selected by the genetic algorithm. The resulting ANN using those features are also converted into interpretable rules. Both approaches are evaluated and their respective performances compared.

2 Methodology

2.1 Data exploratory analysis and preprocessing

The data provided represents summary statistics of raw EEG signals. Therefore, the original signals have already been preprocessed. That was done as follows: For each of the 14 channels (described in section 1.2), raw measurements are taken over time for each participant. For each observation in the given data, each channel's sensor produces many separate measurements. The following features are produced:

Feature	Description
subject_no	ID of the participant
mean	Average signal over time
min	Minimum signal over time
max	Maximum signal over time
std	Standard deviation of signal over time
var	Variance of signal over time
iqr	Interquartile range of signal over time
skw	Skewness of signal over time: symmetry of the distribution
rms	Root mean square of signal over time
sum	Sum of signal over time
hjorth	Mobility parameter of the signal
hurst	Hurst exponent: rate of change in the autocorrelation of signal over time
mean_first_diff	Average difference between two consecutive signal amplitudes
mean_second_diff	Average second difference (difference between consecutive first_diff values)
apen	Approximate entropy: measures how unpredictable the signal fluctuations are
fuzzy	Fuzzy entropy: another measurement for fluctuation randomness in signals
label	Calm (1) or stressful (2)

Table 1. Features in the provided dataset

The choice of summary statistics are identical to those in *towards effective music therapy* [1].

Essentially, the 15 input features (excluding subject_no and label) form a summary of an underlying time series dataset. All of these features are numeric and can be used in an ANN as is. Since there are 14 channels, the total number of features is 210. The last column is the ground truth label of whether the participant feels stressed.

The dataset contains 144 observations. The distribution of the target variable is perfectly even (72 observations of calm and stressed, each).

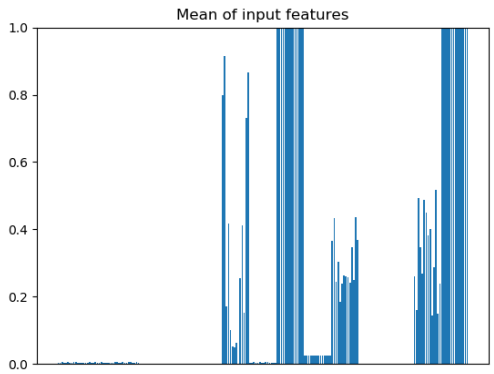

Fig. 2. Average values of the 210 input features, display capped at 1. The values are disparate.

Upon inspection of the data, it is discovered that some features tend to be much higher than others. The mean can be as high as 8.7 (sum_F3) and as low as -0.01 (skw_P8). In order for the ANN to not be dominated by these high-magnitude features, normalization is needed. Data analysis shows that most feature columns have a high proportion of outliers (up to 20%), therefore Min-Max normalization is not appropriate. z-score normalization is performed instead to scale all values to a similar range. Each of the 210 features are transformed with the formula $z = \frac{x-\bar{x}}{\sigma}$.

In the first phase of the experiment, features selection is done manually by identifying variables which are intuitively redundant. The standard deviation features are dropped from the data since variance is already used. Variance gives enough information about the spread of data, and the standard deviation is simply a function of variance. The final number of input features is 196 using this method.

In the second phase of the experiment, we implement a genetic algorithm to stochastically search for the optimal subset of features to be used by the ANN. Hence, data processing is done systematically with a more sophisticated algorithm. We discuss this method in detail in section 2.3, as well as its results in section 3.3.

2.2 Neural network design

In the first phase of the experiment, a number of combinations of hidden layers and neurons are experimented with. The optimal model is found to be a three-layer fully connected neural network, with 196 input features (after manual preprocessing) corresponding to the EEG signals in the input layer, a first hidden layer with 3 neurons, a second hidden layer with 1 neuron, and finally two output neurons (calm/stressed) for binary classification. The weighted sum of each hidden neuron goes through a sigmoid activation function before being fed to the next layer. Since this network is relatively shallow, the vanishing gradient problem associated with the sigmoid activation is not a concern.

The model is trained with the Adam optimizer and cross entropy loss function. The cross entropy is used to penalize more heavily for misclassifying training examples and helps the neural network learn better and faster. In order to not overfit the training data, 300 epochs of training are run, as any more will result in the model attempting to memorise the training patterns. The learning rate is 0.01.

The same neural network architecture will be used to evaluate the performance of different feature selections such that the input processing methods can be compared fairly.

2.3 Feature Selection with GA

To extend and improve on the approach in our first phase, a more sophisticated feature selection process is utilised. Inspired by natural selection, the genetic algorithm iteratively and stochastically evolves a population of feature subsets to find an optimal subset containing only informative variables. In the context of features selection, each individual in the population represents one subset of the available EEG features. Figure 3 shows how representing the inclusion/exclusion of the features as bit strings translates into a fully connected neural network.

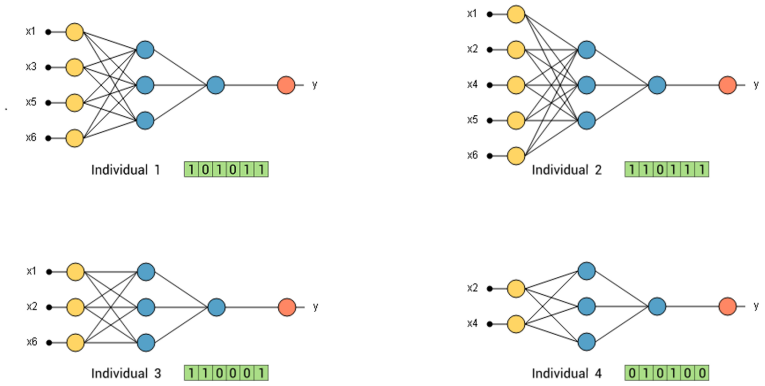

Fig. 3. ANN representation after feature selection with GA [11]

The population size of 20 is chosen and initialised randomly. This population size stays fixed at each generation, of which there are 20. These are parameters tried and tested that have yielded sufficiently good results while keeping a reasonable run time. The mutation rate of each feature is 0.02, which gives an expected number mutations of about 4 (out of 210 total features). This ensures a sufficiently large change that would make an observable difference to the fitness of an individual. We also incorporate elitism which ensures that the population's best fitness is never decreasing and enables the retention of good solutions. The proportion of elites to retain at each generation is 0.1, which means the top 2 individuals are always directly put into the next generation, without mutation.

The GA algorithm consists of the following operations: Fitness evaluation, selection, crossover (breeding) and mutation. The workflow of the algorithm is shown in Figure 4.

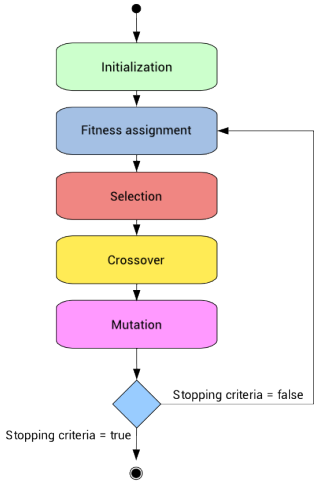

Fig. 4. Workflow of a genetic algorithm for feature selection [11]

We use an individual's average 5-fold testing accuracy as the measure of fitness, consistent with the performance measure in section 2.7. The selection of the surviving individuals into the next generation is done with the proportional method. Each member of the next generation is selected, with replacement, according to their relative fitness in the population. The fitter an individual is, the more likely they will survive. Following selection, a new population is bred through crossover. Uniform crossover is used, where a child of two randomly picked parents will have each bit (feature inclusion) randomly come from one of the parents. This crossover method is used because with so many available features, it's not feasible to manually pick crossover points that make intuitive sense. The better approach is to just enforce random selection of behaviour from the parents. For additional exploration, the GA mutates the bit strings in the new generation with a small probability of 0.02 per attribute. This allows the algorithm to discover good subsets of features by chance.

After 20 generations/iterations, the algorithm terminates and the fittest individual out of that population is picked. The corresponding subset of features determines the optimal input dimensionality of the ANN in phase two.

2.4 Sensitivity Analysis

Given a neural network, one of our main objectives is to reasonably explain its behaviour using rule extraction. The main idea in this paper is to detect decision boundaries along input feature space, by identifying where there is a large rate of change in the output nodes with respect to the input nodes. Consider an input attribute x. A decision boundary is a value z where all input patterns with $x_i < z$ belong to class 0 and all inputs with $x_i > z$ belong to class 1. It is theorized that where a small change in the value of x causes the output to change from 0 to 1 or vice versa, it is likely to be the location of a decision boundary [7]. By searching for such decision boundaries in all input parameters, rules can be produced which predict the output of the underlying neural network.

Since the magnitude of change in the output nodes with respect to the inputs is the determining factor for finding decision boundaries, the fully-trained neural network is differentiated to find the gradients of the input features. For each output node (one for each class out of k), the gradient of input x for observation i is:

$$\frac{\partial C_k}{\partial x_i}$$

For each of the input features, a value z is found is where the absolute value of this gradient is largest. In Engelbrecht *et al* 1999 [7], the decision boundary for each input feature is found by computing the gradient for every input pattern in the dataset, plotting a graph of the gradients and using a curve fitting algorithm to locate the peak gradient. The corresponding input value will then become the decision boundary. However, this is computationally expensive and thus, characteristic patterns are used to compress and represent the training set, as described in T. D. Gedeon *et al* 1993 [5]. Essentially, this reduces the number of gradients calculated to $No.of\ classes \times No.of\ input\ features$. Characteristic patterns are explained further in 2.4.

2.5 Characteristic patterns

Rather than computing the gradient of each input for every pattern in the dataset, it is only done over the characteristic patterns which represent the "typical" input pattern for each output class. Using the fully-trained ANN described in 2.2, each input is classified as calm or stressful. For each of these classes, a characteristic pattern (for example class k_1) is

calculated as the arithmetical mean vector of all the relevant input patterns (all that the ANN classified as k_1). For the set of input vectors classified by the ANN into class k_1, $\{[x_{11}, x_{21}, \dots, x_{j1}], [x_{12}, x_{22}, \dots, x_{j2}], \dots, [x_{1p}, x_{2p}, \dots, x_{jp}]\}$, the characteristic pattern is $[\overline{x_1}, \overline{x_2}, \dots, \overline{x_j}]$. For this EEG dataset, there are two characteristic patterns – one for calm and one for stressful.

2.6 Rule extraction

For each characteristic pattern (2 in this classification task), a gradient is calculated for each pair of output and input node. Only a few input variables with high gradients (in absolute value) are picked for each characteristic pattern to become rules, in order to generalize well and be compact enough to explain. Since these will be the highest gradients, they are the most significant in terms of determining the output and are most likely to be close to decision boundaries. For example, for a characteristic input $[x_{1k,}x_{2k}, \dots, x_{jk}]$, gradients $\frac{\partial O_1}{\partial x_{1k}}, \frac{\partial O_1}{\partial x_{2k}}, \dots, \frac{\partial O_1}{\partial x_{jk}}, \frac{\partial O_2}{\partial x_{1k}}, \frac{\partial O_2}{\partial x_{2k}}, \dots, \frac{\partial O_2}{\partial x_{jk}}$ are all calculated. Once a predetermined number of the highest gradients are selected, the corresponding inputs have proposed decision boundaries. A value on either side of each boundary is sampled to determine which class the ANN will predict if the input attribute were smaller/larger than the boundary value [7]. Thus, each characteristic pattern will have its own set of rules generated. In this paper, the number of rules extracted, which is equivalent to the number of high-gradient input features picked, is exogenously set to 5. This will allow the rules to have relatively high predictive power, while being interpretable. Too many rules will be hard for humans to make sense of.

When classifying a new input, it is first grouped into one of the characteristic patterns by Euclidean distance. This essentially puts them into the category of "being most similar to the typical pattern X". Then the rules belonging to that characteristic pattern is run against the new input – giving a classification of calm or stressful.

For each characteristic pattern, the 5 rules are formed as the condition for an unseen input to be predicted as the class that isn't the characteristic class. We call these 5 rules the "rules_against" the canonical class. To illustrate the "rules_against" concept, consider a simpler case with 2 rules. If the set of rules extracted for the characteristic class "calm" is $mean_{AF3} \geq 0.006\ AND\ mean_{F3} \leq 0.003$, and a new input is closer to the "calm" characteristic pattern than the "stressful" characteristic pattern, then its prediction will be "stressful" (the opposite class) if and only if both the rules are met: $mean_{AF3} \geq 0.006\ AND\ mean_{F3} \leq 0.003$. If at least one of these conditions is not met, the prediction will default back to the input's characteristic class, which is "calm" in this case.

The extracted rules are based on a neural network fully trained with 80% of the provided data.

Since there are only two possible classes for this stress detection task, outputting the next most likely prediction is not meaningful and hence this step is omitted.

2.7 Performance measure

The performance metric is highly dependent on how balanced the classes are within the dataset. Since the provided EEG dataset is perfectly balanced (72 instances of each label) as shown in Figure 3, accuracy can be reliably used [8].

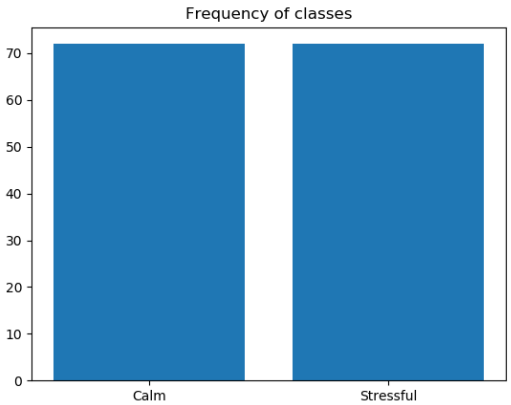

Fig. 3. Distribution of output classes

A train-test-validation split of the dataset is used to tune hyper parameters as well as give an unbiased evaluation of the chosen model. 20% of the data is used for the final evaluation – this portion of data is completely unseen until evaluation stage.

The remaining 80% of the data is first used to tune hyper parameters. Each neural network setting is evaluated with the average testing accuracy from the hold-out test sets during a 5-fold cross validation. Cross validation makes the most use of available data and does not subject the model selection to mere chance when splitting train/test sets. Each input pattern available out of the 80% of the data is used for testing exactly once.

The performance of the ANNs are simply evaluated with accuracy compared to the ground truth labels provided.

Performance of the extracted rules is also evaluated with accuracy. Using rules extracted from the optimal ANN trained with all training data (no cross validation), predictions from these rules are compared to both the ANN outputs and the ground truth labels. If the rules demonstrate high accuracy, it would suggest that a meaningful abstraction of the ANN behavior has been found.

3 Results and discussion

	Manual feature selection (dropping standard deviation fields)	Features selected by genetic algorithm
ANN against ground truth	93.8%	95.4%
Rules against ANN output	85.0%	84.9%
Rules against ground truth	86.2%	86.3%

Table 2. Final evaluation results – average final testing accuracy over 100 runs.

3.1 Neural network performance

Using manual feature selection, the chosen 196-3-1-2 architecture with sigmoid activations showed the best average cross-validation accuracy of 94.2%, and a final testing accuracy of 93.8% averaged over 100 runs. Other configurations were experimented with, such as those with one hidden layer, like 196-3-2 and 196-2-2, all of which averaged around 90-93%. An accuracy higher than what is obtained is very difficult as ultimately, neural networks are tools of generalization, therefore if there are inputs in the testing set which differ from what it has seen as the norm, the prediction will be incorrect. This is a fundamental feature of machine learning in general.

The evaluation result shows the high predictive power a simple ANNs has for this particular dataset. The summarised version of time series EEG data seems to be very effective indicators of stress in humans. This is not extremely surprising as contact-based headsets with a multitude of highly relevant sensors whose statistics are processed and summarised should give good insights into physiological reactions.

Although physical sensors may not always be practical [2], the results show that when they happen to be available for use, they can be very effective for stress detection.

3.2 Extracted rules performance

This is an example of some rules extracted in one of the ANN runs:

1. For an input whose nearest characteristic pattern is “calm”:
    a. Manual feature selection: $Rules\_Against = sum\_FC6 < 6.7\ AND\ sum\_P7 < 8.6\ AND\ sum\_F4 < 8.0\ AND\ skw\_O1 < 0.02\ AND\ hjorth\_P7 > 0.02$
    b. GA feature selection: $Rules_{Against} = sum\_O1 < 7.8\ AND\ hjorth\_P7 > 0.02\ AND$ max _P7 $< 0.005\ AND\ sum\_O2 < 8.1\ AND\ sum\_P7 < 8.6$
2. For an input whose nearest characteristic pattern is “stressed”:
    a. Manual feature selection: $Rules\_Against = sum\_FC6 > 6.4\ AND\ sum\_P7 > 8.2\ AND\ sum\_F4 > 7.7\ AND\ skw\_O1 > -0.03\ AND\ hjorth\_P7 < 0.02$
    b. GA feature selection: $Rules_{Against} = sum\_O1 > 7.4\ AND\ hjorth\_P7 < 0.02\ AND$ max _P7 $> 0.005\ AND\ sum\_O2 > 7.7\ AND\ sum\_F8 > 7.4$

One interesting feature of the rules is that most of the time, both classes use the same input variables to form the rules, although the directions of the decision are flipped. This is due to the fact that this is a binary classification task. The opposite of one class’ decision condition will be the decision condition for the other class.

The average testing accuracy of the rules when compared to the output of the first ANN is 85.0%. After adding GA feature selection, the rule accuracy vs ANN prediction is 84.9%. This is a good level of accuracy considering the simplification from a continuous function-approximating ANN to a discrete and much reduced set of simple rules. The high accuracy is in large part due to the effectiveness of the use of characteristic patterns. The characteristic patterns themselves are an encapsulation of the ANN's behavior. By averaging the inputs that give each respective class as predictions, we obtain a "typical" input that the ANN would recognize as calm or stressed.

Because the rules formed for prediction in this experiment are not unconditional (i.e. they are not applied to a raw input straight away) and instead, inputs are first likened to a characteristic pattern, they rely on the characteristic patterns a lot to make the final prediction. The rules are useful mostly for classifying outliers, as exceptional conditions will grant it the power to predict against its typical class. This is a possible explanation for the testing accuracies against ground truth labels of 86.2% (manual) and 86.3% (GA), which are higher than the rules' accuracies against the ANN output. The additional accuracy obtained may be due to the extra predictive power the rules have when it comes to classifying outliers, for which the ANN will sometimes struggle with given its weights are general.

The predictive power the generated rules are almost identical whether a GA is used for feature selection.

3.3 GA feature selection performance

The fittest subset of features after 20 generations of the genetic algorithm contains 98 attributes out of the 210 available. This is a very significant reduction in dimensionality of 53%. Using these features to train an ANN on all 80% of the training data, and averaging the final testing accuracies over 100 runs, we obtain 95.4% accuracy. This is a significant improvement on the ANN without GA feature selection of 1.6%. Despite only using a small population of 20 and a low number of generations, the GA is able to eliminate over half of the input features as redundancies achieve better generalization (better results with less information).

3.4 Comparison with related stress detection research

The dataset on which this paper is based is very new and not much research work has been performed using it. However, there is related research experimenting on the same task of stress detection. One such paper is *Thermal Super-Pixels for Bimodal Stress Recognition* [2]. For this experiment, no physical sensors were used and instead, computer vision techniques are utilized. Accuracies of 60% and 82% are obtained using RGB and thermal modalities, respectively. Compared to the higher of the two (82%), the results of this paper using EEG sensors and rule extraction does outperform it.

Contact-based sensors ultimately will perform better for classification, although contact-free methods like computer vision still seems to yield good results and may be necessary when there is no equipment availability.

4 Conclusion and future work

Stress detection using a simple ANN along with sensitivity analysis-based rule extraction performs is shown to be very effective on their own. We have shown that using EEG data, a person's stress level can be reliably recognized. Ultimately, physical sensors outperform non-traditional methods such as computer vision even using a simple model.

Through rule extraction, experts can be confident when using neural networks to solve sensitive problems, using the rules as a "sense check" to provide an additional layer of assurance. As the rules help with making the network more interpretable, they can assist in reducing the skepticism surrounding neural network techniques.

Adding a genetic algorithm to select important features further strengthens the predictive power of the ANN and subsequently, the rules generated from it. We have shown that these three techniques in conjunction with EEG-based devices can effectively detect human stress.

A possibility for future work would be extracting unconditional rules that do not rely on the characteristic pattern, since this dataset is small enough to afford this. Characteristic patterns are more important for reducing computational costs with large datasets. Unconditional rules will require the sensitivity analysis to be performed over the entire training set, without any grouping into "typical" patterns. This can potentially help with finding better decision boundaries and hence require fewer rules to achieve a similar level of accuracy.

The feature selection process can also be modified slightly. With more time permitted, the configuration of the GA can be further fine-tuned and experimented with. With 210 features, there are 1.6×10^{63} possible bit strings, so setting a population size much larger than 20 will likely help us discover even more optimal solutions that reduce redundancy in the input space. Another possible change to the GA is the crossover step. If we had domain expertise and a deeper understanding of the different EEG channels and what they represent, a multi-point crossover could prove to be effective, where the points are strategically chosen such that certain summary statistics staying together will always be beneficial due to any existing codependency between them. A non-uniform crossover function will also reduce computation time as well as explainability, as uniform crossover's random behaviour is difficult to qualitatively justify.

5 References

[1] Rahman,J.,Gedeon,T.,Caldwell,S.,Jones,R. & Jin,Z.(2020).Towards Effective Music Therapy for Mental Health Care Using Machine Learning Tools: Human Affective Reasoning and Music Genres. Journal of Artificial Intelligence and Soft Computing Research,11(1) 5-20.

[2] Irani,R.,Nasrollahi,K.,Dhall,A.,Moeslund,T. & Gedeon,T.(2016). Thermal super-pixels for bimodal stress recognition. 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), Oulu, Finland, 2016, pp. 1-6.

[3] Lupien,S.,Maheu,F.,Tu,M.,Fiocco,A. & Schramek,T.(2007). The effects of stress and stress hormones on human cognition: Implications for the field of brain and cognition. Brain and Cognition, vol. 65, no. 3, pp. 209 – 237, 2007.

[4] Saeed, S., Anwar, S. M., Khalid, H., Majid, M., & Bagci, A. U. (2020). EEG based Classification of Long-term Stress Using Psychological Labeling. Sensors (Basel, Switzerland), 20(7), 1886.

[5] Gedeon,T.,Turner,H.(1993).Explaining student grades predicted by a neural network. Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), Nagoya, Japan, 1993, pp. 609-612 vol.1.

[6] Hailesilassie T. (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, page 371-381.

[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] Chawla,N.,Japkowicz,N., & Kotcz,A.(2004). Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl. 6, 1 (June 2004), 1–6.

[9] Erguzel,T., Ozekes,S.,Tan,O. & Gultekin,S. (2014). Feature Selection and Classification of Electroencephalographic Signals: An Artificial Neural Network and Genetic Algorithm Based Approach. Clinical EEG and neuroscience. 46. 10.1177/1550059414523764.

[10] Babatunde,O., Armstrong,L., Leng,J. & Diepeveen,D. (2014). A Genetic Algorithm-Based Feature Selection. International Journal of Electronics Communication and Computer Engineering. 5. 889-905.

[11] Gomez,F., Quesada,A. & Lopez,R. (2017). Genetic algorithms for feature selection. Neural Designer Data Science and machine learning blog. https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection