You Are What You Input: Using Feature Selection Techniques to Improve Performance of a Binary Classification Neural Network

Atigun Treekutpan

Research School of Computer Science, ANU College of Engineering and Computer Science,
Australian National University
Canberra 2601, Australia
u6837660@anu.edu.au

Abstract. Research conducted by Chen et al [1], demonstrates that the pupillary response of observers can be used to train networks to achieve high accuracy in classifying between genuine and posed anger. This paper develops a similar simple neural network for this classification task, using a version of the anger dataset. Input Feature Analysis and Feature Selection techniques adapted from Gedeon's paper [3], to determine any irrelevant features, which could be removed to improve the performance of the trained neural network. The paper first conducts an analysis of the proportional contribution of inputs based on their weight values. Then sensitivity analysis is performed using perturbation of features values to identify the most and least significant features. A leave-one-feature-out or brute force approach is then used, removing one feature or pair of features at a time and assessing the change in testing accuracy. Lastly, a genetic algorithm was utilized to determine the best set of input features to be included in the training of the neural network. After evaluating the changes to performance of our neural network based on each method's recommendation, the removal of Diff1/Diff2 resulted in the best improvement to the neural network's performance, demonstrating the usefulness of feature selection on classification problems based on dataset with as few as 6 features.

Keywords: Feature Selection, Input Significance, Input Pruning, Feature Significance, Leave-One-Out, Brue Force, Sensitivity Analysis, Genetic Algorithm, Input Contribution, Pupillary Response, Genuine Posed Anger

1 Introduction

Gaining insight into a person's true intentions or state of emotion, despite contradicting appearances, is an interesting area with many potential applications. Related to this concept is how humans respond to facial expressions of other humans in terms of gauging the veracity of their emotions. As Chen, Gedeon, Hossain, and Caldwell present in their paper [1], an observer's physiological responses are significantly different when observing genuine vs posed anger, and can be used by machines to obtain good accuracy for distinguishing between the two. This interaction between observing anger and an observer's physiological responses can be used as input for a classification tool where distinguishing between genuine or posed anger could be useful in interview situations, for example, law enforcement interrogation. In a related paper by Hossain and Gedeon [2], a similar measure was also found to be utilized in detecting between posed and real smiles, and as mentioned by Chen et al. [1], physiological data could be used to assess the genuineness of several human emotions.

Given the literature in establishing the validity of using the physiological response of observers to gauge emotions of other people, this paper will attempt to adapt and improve upon the results presented by Chen et al. [1], achieving a 95% accuracy genuine vs. acted anger detection classifier. The main technique this paper will explore for potential improvement is feature analysis and selection, adapted from the techniques conducted by Gedeon [3], in which several methods are used to analyze the contributions of the input features on the outputs, as well as determining the relative significance of the available features to determine if removal of the least significant features can result in improvements of our trained classifier. The main premise of Gedeon's paper [3] is that applying feature analysis and feature selection can help avoid using irrelevant features when training neural networks.

In addition to some of the methods utilized by Gedeon [3], this paper will in addition, utilize a genetic algorithm approach on the same dataset, to determine the best set of features to use for the best accuracy, which will be compared with the feature selection techniques in this paper that are adapted from Gedeon's paper [3].

2 Overview of the Anger Dataset

The dataset which is used to train the neural network consists of 400 entries, with 7 features or attributes, 6 numerical one categoric, and a label of either genuine or posed anger for the video being watched for that input entry. All the features are descriptive statistics of the pupillary response, the main physiological response measured in the paper by Chen et al [1]. The pupillary response captures the average pupil diameter of the observer, which changes each frame of the video being watched, depending on what is on each frame as displayed in Figure 1.

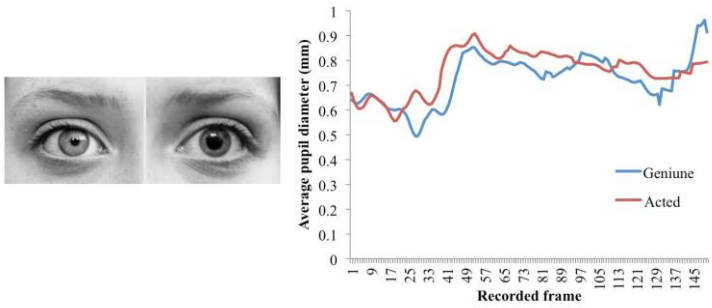

Fig. 1. Shown (Left), change in the pupillary response of a human observer to a visual stimulus. Also shown (Right) demonstration of pupillary response data measurement, measuring diameter over frames of a video. Figures directly referenced from Chen et al. paper, Are you really angry? Detecting Emotion veracity as a proposed tool for interaction [1].

The numerical features of the dataset include the mean (Mean), standard deviation (Std), first difference (Diff1), second difference (Diff2), principal component analyzed first difference (PCAd1), and the principal component analyzed second difference (PCAd2). The categoric feature, is Video name, which is not used for building this neural network because it directly identifies whether the video being watched is of a genuine or posed anger. In addition, there is another column which is identifier information, corresponding to individual observes labeled O1 to O20. For the purposes of developing the neural network, we consider only the 6 numerical features, with the target being the label of genuine or posed anger.

3 Developing a Binary Classification Neural Network for the Anger Dataset

For this binary classification problem, a simple single hidden layer feed-forward network with back propagation was used as a starting point. Data was normalized using the min-max normalization. K-fold cross-validation was utilized, with k = 5 to tune the hyper-parameters, number of hidden neurons, learning rate, number of epochs, as well as choice of loss function and Pytorch optimizer, by looking at the average testing accuracy of the 5 folds. Since we have 400 data entries, for each fold we had 320 entries in the training set and 80 entries in the testing set. Accuracy and loss after each epoch of training was considered, to see if the network had learned as much as it could or if there was oscillation of the accuracy.

After the process of trying hyperparameter turning, the final binary classification neural network that is developed for the anger dataset has 6 input neurons corresponding to the 6 numerical features, one hidden layer, consisting of 15 hidden neurons, and 2 output neurons, corresponding to the two classes, genuine or posed anger. This is output from a sigmoid activation function in the output neurons. The learning rate is 5% (0.05) with number of training epochs at 2,000. The training utilizes Cross Entropy Loss for back propagation, and the Adam algorithm for its optimizer. Running this network 10 separate times on an 80/20 training/test split resulted in an average testing accuracy of 83.89%.

The main justification for this model selection is that it achieved sufficiently good results, while being very simple in architecture. Having a single hidden layer, as well as less than three times the amount of hidden neuron compared to input neurons, reduces the time and computational requirement to tune our parameters, as well as makes it less likely that overfitting will occur as a result of our model parameters.

4 Applying Feature Analysis Techniques to Determine Least Significant Features

In Gedeon's paper on analyzing the magnitudes functional measures of input data [3], the major pitfalls of sub-optimal treatment of input data is discussed. Data which encodes the underlying data structure, and irrelevant features may mislead the model during training. Class imbalance is also a detriment to training if allowed to exist in training data. The anger dataset, fortunately, has a perfect 50/50 balance between its two classes due to the experimental design of Chen, Hossain, et al. [1].

This paper's objective to identify potentially irrelevant features of the anger dataset will follow a similar method as Gedeon's paper on feature magnitude and functional measures [3]. The objective of each technique to determine the significance of each feature relative to the others. As a result, the least significant features as determined by each technique will be considered for removal and then evaluated if any improvement in the neural network performance has been achieved. A paper by Satizábal and Pérez-Uribe [4] similarly aims to determine some input relevance measures to use for input dimension reduction, and utilizes methods involving network weights and sensitivity analysis through perturbation, used in a similar fashion as Gedeon [3].

4.1 Analyzing Neuron Weights to Determine Input Contribution

The first technique to assess the relative significance of the available input features is to analyze the weight matrix of the trained neural network, similarly to Gedeon's approach [3], using contribution metrics based on Garson's measure of proportional contribution. The Q-values as detailed in Gedeon's approach [3], essentially provide information on the magnitude of an input's contribution to the output based on the weights of the neural network.

$$Q_{ik} = \sum_{r=1}^{nh} \left(P_{ir} \times P_{rk} \right) \qquad (1)$$

When the neural network described in section 3 is trained and evaluated, the weights are extracted for analysis. Using the expression shown above, the average proportional weight contribution of each input across all hidden layer neurons is found, and then the average contribution of the hidden neurons to the output neurons. This now provides a relative order of significance for the features, under the concept that the most significant input features or input neurons will contribute the highest proportion to the total weights of the hidden neurons in the hidden layer. Table 1 displays the relative order of significance determined by extracting, calculating, and comparing the contribution values.

Table 1. Relative Ranking of Input Features Based on Proportional Weight Contribution Values.

Proportional Contribution Analysis	Ranking of Relative Significane of Features					
	Most Significant					Least Significant
Input Ranking	PCAd1	PCAd2	Diff1	Std	Diff2	Mean

Based on the analysis of the neural network's weight matrix and the input neurons contributions, the Mean is the least significant input, and is the first candidate feature to be removed for evaluation.

4.2 Sensitivity Analysis of Input Features

Sensitivity analysis is a common method used to determine the relative significance of input features to the output. If an input value is slightly changed, the most significant features will result in more drastic changes to the output. In a paper on sensitivity analysis for reducing input data dimension [5], Zurada et al. develop sensitivity measures to capture numerically a representation of the output's 'sensitivity' to each input change. As the paper mentions it is quite useful when dealing with large amounts of redundant data. However, the approach of Gedeon in analyzing magnitude and functional measures [3], which provides a relative ranking of the inputs will be more relevant.

Values for a specific input feature will be altered slightly, and the change in the output will be observed, and a relative ranking will be established for our features based on how drastic the changes in the output are. Several perturbation methods will be used much like Gedeon's [3] implementation of sensitivity analysis in order to reduce any variance or misleading results that could occur if only one perturbation method was conducted. The following perturbation methods were used:

1. Perturbation of single inputs on a single pattern, by +5% of its numerical value.
2. Perturbation of single inputs on a single pattern, by +10% of its numerical value.
3. Perturbation of single inputs on all training patterns, by +5% of its numerical value.
4. Perturbation of single inputs on all training patterns, by +10% of its numerical value.

The methodology of sensitvty analysis is shown in Figure 2. The data was separated into a training and testing set (80/20), then training set was then copied, and had perturbation applied to it. Two separate neural networks are then trained, one with the unchanged training set, the other with the training set with the perturbation method applied. The neural networks are then evaluated using the same testing set, then performances are compared.

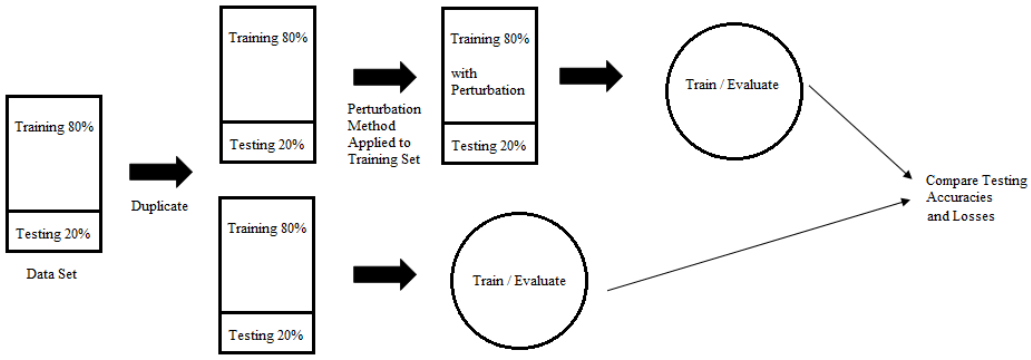

Fig. 2. Process diagram of conducting sensitivity analysis. The training set has perturbation applied in one case, and both are evaluated on the same testing set.

To account for issues such as unrepresentative testing sets, the two neural networks were run 10 times each and the average change in testing accuracy was used to evaluate. The direction of change is not relevant, so the differences are absolute. For each perturbation method, the features are then ranked by relative significance depending on the magnitude of the change in testing accuracy. Table 2 shows the features ranking for each perturbation method

Table 2. Ranking of relative significance of features based on the results of Table 1, for each perturbation method applied.

Perturbation Method	Ranking of Relative Significane of Features					
	Most Significant					Least Significant
Δ5% Single Pattern	Std	PCAd1	Diff2	Diff1	PCAd2	Mean
Δ10% Single Pattern	PCAd1	Diff1	Diff2	PCAd2	Std	Mean
Δ5% All Patterns	Std	Mean	PCAd1	PCAd2	Diff2	Diff1
Δ10% All Patterns	PCAd1	Mean	Diff1	Std	PCAd2	Diff2
Averaged Ranking	PCAd1	Std	Diff1	Mean	Diff2	PCAd2

The perturbation on PCAd1 overall had the most drastic effect on testing accuracies, while perturbation on PCAd2 overall had the least drastic effect on testing accuracy. An interesting observation is that the features that capture the spread of how much pupillary diameter is varying over the course of the video seem to be the most significant. PCAd2 is the candidate feature to be removed from our model as a result of sensitivity analysis.

4.3 Leave-One-Feature-Out (LOFO) / Brute Force Approach

Similar to Gedeon's [3] approach, a brute force approach will be used to compare results with the other methods used in this paper. The purpose of feature selection is to potentially remove any irrelevant features, so a method which removing features or pairs of features to see the impact on the neural network's performance will produce very transferrable results. The small number of features in the anger dataset means the usual disadvantages of using this method on datasets with a large number of features can be avoided, such as high computational cost or inconsistent results when removing one feature at a time.

A study of literature reveals that many papers which aim to develop techniques related to feature selection will often use the Leave-One-Feature-Out (LOFO) approach as one the methods for comparison. de Sá in his Variance based approach to feature selection [6], uses LOFO as a baseline to compare his approach by analyzing the variance of the input weights after each batch training, by removing features and observing the effect on the losses of the training. Fen, Chen, and Xu, also experiment with a leave-one-out feature selection approach and claim that if the network is properly tuned, can perform better than many feature selection methods, when considering criteria such as classification accuracy, especially on real-world datasets [7].

In addition to single inputs being removed, Gedeon's methodology when performing brute force approach [3] will also be adapted, which removes features in pairs. Only 4 pairs of inputs were considered for this approach. Second Differences are based off first differences, PCAd1 is based off Diff1, PCAd2 is based Diff2, and therefore PCAd1 and PCAd2 are related. For these reasons, these were the features selected to additionally be removed in pairs.

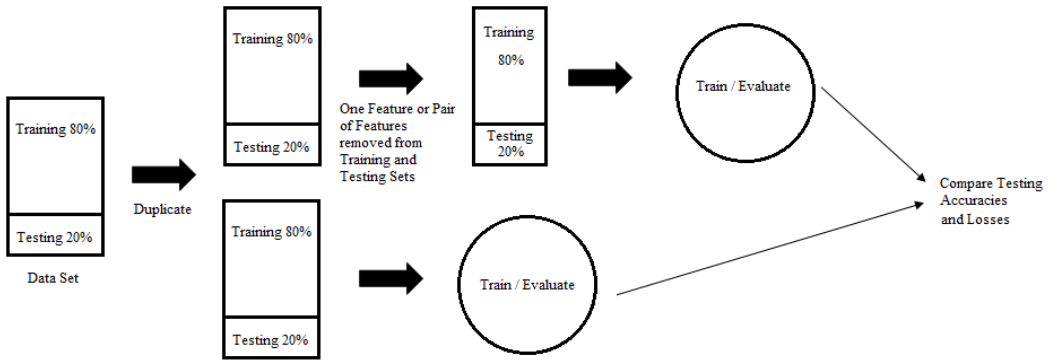

Fig. 3. Process diagram of conducting Brute Force Approach. The training and testing set has feature or features removed in one case, and are evaluated on test sets of the same entries, but with different number of present features.

The methodology is similar to before. The training and test set, split 80/20, are duplicated, and on the duplicate, a single feature or single pair of features is removed from both the training and test sets. The original training set is used to train the model, and then evaluated using the test set with all features. The training set with feature/features removed is correspondingly evaluated using the test set with the same feature/features removed and the average testing accuracies are compared between the two networks. Additionally, following the methodology of Gedeon [3] and de Sá [6], the total losses are also compared as an alternative measure of network performance.

Unlike sensitivity analysis, we are interested in whether the accuracies have increased or decreased, so the signs are considered. When interpreting the results, the most important features will cause the accuracies to decrease and the losses to increase when removed. The least important features will either see no change in accuracies or an increase in accuracies, while seeing either no change in losses or a decrease in losses. Tables 3 and 4 display the results of the Brute Force Approach.

Table 3. Average change in neural network performance metric for each feature/feature pair removed over 10 network runs.

Performance Metric	Average Change in Performance Metric for Each Feature/Feature Pair Removal									
	Mean	Std	Diff1	Diff2	PCAd1	PCAd2	Diff1/PCAd1	Diff2/PCAd2	Diff1/Diff2	PCAd1/PCAd2
Testing Accuracy (%)	2.520	-2.790	1.388	4.036	-28.646	-3.342	-22.630	-2.560	-0.556	-32.924
Total Loss during Training	-135.514	95.004	-15.966	0.754	515.822	149.972	617.602	87.038	-27.106	734.754

Table 4. Ranking of relative significance of features based on the results of Table 3, for each feature/feature pair removed.

Performance Metric	Ranking of Relative Significane of Features/Feature Pairs									
	Most Significant									Least Significant
Testing Accuracy (%)	PCAd1/PCAd2	PCAd1	Diff1/PCAd1	PCAd2	Std	Diff2/PCAd2	Diff1/Diff2	Diff1	Mean	Diff2
Total Loss during Training	PCAd1/PCAd2	Diff1/PCAd1	PCAd1	PCAd2	Std	Diff2/PCAd2	Diff2	Diff1	Diff1/Diff2	Mean

Table 4 shows that Mean and Diff2 are the least significant features according to the brute force approach. Diff1/Diff2 also appears to rank low, which may have something to do with PCAd1/PCAd2 including similar information since that pair is ranked highest based on both accuracy and loss metrics. Mean and Diff2 are the candidate features and Diff1/Diff2 is the candidate feature pair to be removed from our model as a result of LOFO/Brute force approach.

5 Using a Genetic Algorithm to Determine Best Combination of Features

In order to compare with the results of the feature selection methods used in the previous section similar to Gedeon's approaches [3], a genetic algorithm will also be utilized to select the best combination of features. In a paper by Sharma and Gedeon [8], using genetic algorithms for the task of feature selection for classification problems show the ability to greatly improve the classification rates of neural networks, similar to the objective of this paper. In facto further research of literature has proven that using genetic algorithms for feature selection specifically for the purpose of removing irrelevant features is common practice, such a paper by Hussein, Kharma, and Rabab [9] in which genetic algorithms selecting features improved the classification performance of their pattern recognition neural network, and similarly by Gamarra and Quintero [10], where features selected using error rate to determine fitness for their genetic algorithm also improved the performance of their digital image recognition classification.

For the genetic algorithm used in this approach, the chromosomes of each member of the population was expressed as a six-digit binary chromosome, representing the six available features, with a 1 meaning the feature was included and 0 meaning the feature was not included. Fitness of members will simply be based on test accuracy, similar to Gedeon's approach [8], and Gamarra and Quintero [10], who use error rate. When deciding on the selection, crossover, and mutation methods, some key points were considered. With only 64 available combinations for a binary six-digit chromosome, a relatively high selective pressure was preferred. Variation in test accuracy could occur due to the test set, therefore, including several parents from each generation was also preferred, as generation overlap could allow poor performers based on one test to be able to redeem themselves in the next generation's evaluation. Therefore, a rank-based selection was utilized, as absolute fitness is not as important as relative fitness in this situation. Lastly, the children should not be an even 50/50 split of the parents, but should have slightly more genetic material from the higher-ranking parent. Again, due the quite limited potential combinations, exploration was not a high priority, so only one gene per child was put through a mutation probability.

Table 5. Parameters of Genetic Algorithm Used for Anger Dataset Feature Selection

GA Parameter	Setting or Value
Population Size	10
Number of Generations	6
Crossover Split	0.6/0.4 , 0.6 from higher ranked parent
Mutaion Probability	20%
Crossover Type	One point Cross over [XXXX|XX]
Mutation Type	Random Mutation on One Gene
Selection Type	Rank-Based Selection
Generational Overlap	50%

Table 5 displays the properties of the genetic algorithm used for this paper. The initial population is randomized. A generation is evaluated by classification accuracy, with the top 5 members in terms of fitness being selected as parents. For each child created by crossover, one-point crossover was used, with the first 4 genes coming from the higher ranked parent, and the last 2 gens coming from the lower ranked parent. 5 children are created repopulate to 10, with 2 children from rank 1 and rank 2, one child from rank 2 and rank 3, one child from rank 3 and rank 4, and one child from rank 4 and rank 5. Each child has a mutation probability applied on their first gene. The new generation is created, and evaluated again, and put through this process again depending on the set hyper-parameter of generation limit.

Changing the mutation probability did not seem to have a large effect on our genetic algorithm. Again, exploration is not quite necessary for this case, and experimenting from 10% to 40% mutation probability yielded similar results. Table 6 displays some of the results which were found from using different generation limits, and its effect on the average fitness of the final generation, and the number of unique chromosomes in the final generation.

Table 6. Results of Experimentation on Generation Limit for Genetic Algorithm Feature Selection

Results of Genetic Algorithm Experimentation			
Generation Limit	Number of Unique Chromosomes	Highest Fitness Chromosome	Average Fitness of Final Generation (Test Accuracy)
2	7	[110110]	74.99%
3	4	[100111]	83.03%
4	4	[111011]	85.34%
5	3	[110011]	87.09%
6	2	[011011]	85.54%

As Table 6 displays, there is significant improvement after the third generation, and after that the improvement of allowing addition generation plateaus off. At around 6 generations the population converges to around 2 or 3 chromosomes. While we have achieved decent testing accuracies with the genetic algorithm, if in future work using larger populations and more complex selection and replacement methods, efficiency and computational requirement problems may occur similar as to the methodology of Hussein et al [9].

After the process of running the genetic algorithm with a generation limit of 6, several times, the observation is made that for the final generations with the higher average accuracies, the highest performing chromosomes are either [110011] or [011011]. The same chromosomes being converged on by the algorithm follows the results of Sharma and Gedeon's approach [8], where the GA excels at identifying features with redundant information for the classification task. This means that the genetic algorithm approach has determined these feature combinations to do the best on test accuracies. These chromosomes respond to dropping out Diff1/Diff2, and Mean/Diff2 from training and testing. These will be the recommended feature pairs to be removed as a result of the genetic algorithm approach.

6 Results of Feature Selection: Removing Least Significant Features

The least significant features or feature pairs recommended by each method will now be compared by evaluating how the removal of each recommended candidate feature or feature pair impacts the performance of the neural network. In addition, the limitations of the results in the context of the working dataset and will be discussed.

6.1 Impact of Removing Candidate Features on Performance of Neural Network

For evaluation, neural networks with hyperparameters as discussed in section 3 will be run again with the same 80/20 split for training and testing, but with the recommended features or feature pairs removed. For each feature removed, the testing accuracies will be averaged across 10 network runs, again to help protect against overly easy or hard test sets. Table 7 below summarizes the features and feature pairs recommended to removed from training testing based on each method utilized in this paper.

Table 7. Recommended Features or Feature Pairs to be Removed from Each Feature Selection Method

Method	Least Significant Inputs Determined By Approach
Proportional Weight Contribution	Mean
Sensitivity Analysis	PCAd2
Brute-Force Approach	Mean, Diff2, Diff1/Diff2
Genetic Algorithm Approach	Diff1/Diff2 , Mean/Diff2

In order to evaluate the recommendations produced by the applied feature selection techniques, the neural networks with hyperparameters as discussed in section 3 will be run again with the same 80/20 split for training and testing, but with a feature removed. For each feature removed, the testing accuracies will be averaged across 10 network runs, again to help protect against overly easy or hard test sets.

Improvement in network performance will be evaluated using testing accuracy, as the objective is to achieve results as close to or better than the results reported by Chen et al. [1], where trained machine classifiers were able to correctly distinguish between genuine and posed anger with 95% accuracy. The following table and figure show the results of removing the candidate features produced from feature selection techniques on the average testing accuracy of the neural network over 10 network runs.

Table 8. Average testing accuracies of neural network with candidate features/feature pair removed. Recommendations based off leave-one-feature-out or brute force all resulted in improved performance in classification.

	Features Removed from Training/Testing					
	None	PCAd2	Mean	Diff2	Diff1/Diff2	Mean/Diff2
Average Testing Accuracy Over 10 Network Runs (%)	83.89	76.62	84.77	87.46	88.93	87.38

Removing the features and feature pair that were recommended by the brute force approach all improved the average testing accuracy of the neural network. This is fully expected, since we are evaluating its performance through testing accuracy, the same way we did when determining the relative significance of the features. The most significant improvement, seen in Table 8, was achieved by removing the least significant feature pair, Diff1/Diff2 which was recommended by both brute force approach and the genetic algorithm, as seen in Table 7. This is quite interesting, as it demonstrates that even when dealing with a dataset with as few as 6 features, removing 2 features can improve the performance significantly, in this case by a little over 5%. This indicates that Diff1/Diff2 may be unimportant for classifying between genuine or posed anger, or more likely, that its information is encapsulated in one of the other features.

Feature selection techniques have managed to improve the average testing accuracy of the neural network from 83.89% to 88.93%. While this falls short of the 95% accuracy reported by Chen et al. [1], an average improvement of 5.04% was achieved for the neural network developed for this paper from 83.89%, demonstrating the usefulness of performing feature selection techniques and considering potentially irrelevant features. For this dataset, given the available features, the brute force approach and the genetic algorithm approach was the best approach.

Unfortunately, the candidate feature to be removed recommended by sensitivity analysis did not achieved desired results. The model performs worse without the PCAd2 feature in its training and testing. As sensitivity analysis has been demonstrated to be useful feature selection tool, in papers for example by both Gedeon [3], and Zurada [5], the results observed for this paper's sensitivity analysis indicates sub-optimal execution. More refined sensitivity analysis should be considered.

6.2 Discussion of Limitations of Anger Dataset

Based on the results of the sensitivity analysis, it is useful to discuss the limitations that are created by the characteristics of the anger dataset when performing feature selection techniques. The features of the anger dataset are actual statistical measures of the pupillary response. Statistical descriptive measures are often related or derived from one another. This can result in overlapping information, and result in issues similar to multicollinearity in linear regression. Compound with the fact that only 6 features are available for this analysis, it is understandable why a feature selection technique such as perturbation may create inconsistent results. When an input value of a feature for a pattern was changed, the resulting change overserved in testing accuracy may have been less of a reflection of the sensitivity of the output to that input, but that distorting a dataset with a limited number of features that may be dependent on one another creates inconsistent results.

As mentioned by Feng et al. [7] in their implementation of leave-one-feature-out strategy on their chosen datasets, using a brute force approach, not only is a good method to achieve better classification performance directly, but also helps to provide some insight into dependence of features on one another, which has an effect on techniques such as sensitivity analysis. The takeaway from the results of this paper should then, not be that brute force or LOFO approach is better than sensitivity analysis, but that for this particular dataset given the available features, brute force approach was better suited to determine the best way to improve classification accuracy through feature selection. The fact that several feature pair removals yielded improved results also suggest overlap in formation in the available features.

7 Conclusions and Future Work

Feature selection techniques were used to identify potential features to be removed from the input data to improve the performance of a binary classification neural network. For the anger dataset produced from the paper of Chen et al. [1], a Leave-One-Feature-Out or brute force approach provided candidate features to be removed that all resulted in an improvement in the average testing accuracy of the neural network. The candidate feature of sensitivity analysis did not result in an improved performance. The network weight matrix approach as conducted by Gedeon [3] resulted in improvement as it recommended the same feature to be removed as the brute force approach. Lastly, one of the combination of features recommended by the Genetic Algorithm approach was the same as the brute force approach. The best result was achieved when the feature pair of Diff1/Diff2, recommended by brute force approach and genetic algorithm approach, was removed from the training and testing of the neural network, which had an average testing accuracy of 88.93%.

Future work for improving this model can be separated into three main areas. The first is improvement to the methodology of this paper. A refined sensitivity analysis approach, so that we may achieve results that agree with the brute force approach, should be pursued, by considering other metrics to assess changes in the networks behavior other than testing accuracy, such as loss as conducted by Gedeon [3], or the variance of input weights per batch approach by de Sá [6]. A different cross-validation method could also be explored to see if better hyperparameters for the network can be found. Further experimentation of the genetic algorithm should also be conducted, with larger population sizes and perhaps different selection, crossover, mutation, and replacement methods.

The second area for future work is redo this process but with more features available for this dataset. Chen et al. [1] collected more than just pupillary response information, and other physiological responses or other features could be introduced to see if feature selection results will be similar. While improvement in accuracy was achieved, there is a limit to the usefulness of feature selection for a dataset with 6 features. More features will allow more insight to be gained about the task of classifying between genuine and posed anger, and allow for comparison with other feature selection techniques in a situation unlike this one where a brute-force approach simply made the most sense.

Lastly, the genetic algorithm approach could also be applied to feature selection, but also the hyper-parameter tuning of the network in section 3. While feature selection has proven to improve the network performance, the exploratory aspect of genetic algorithms through mutations could reveal an even better network architecture which would provide a better baseline network.

References

1. Chen, L., Gedeon, T., Hossain M., Caldwell, S.: Are you really angry? Detecting emotion veracity as a proposed tool for interaction. 29th Australian Conference on Computer-Human Interaction (OzCHI 2017, 412-416. (2017)
2. Hossain, M., Gedeon, T.: Classifying Posed and Real Smiles from Observers' Peripheral Physiology. 11th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '17) (2017)
3. Gedeon, T.: Data Mining of Inputs: Analysing Magnitude and Functional Measures. International Journal of Neural Systems Vol.8 No.2 , 209-218 (1997)
4. Satizábal M. H.F., Pérez-Uribe A.: Relevance Metrics to Reduce Input Dimension in Artificial Networks. Artificial Neural Networks – ICANN 2007. Lecture Notes in Computer Science, vol 4668 (2007)
5. Zurada, J., Malinowski, A., Cloete, I.: Sensitivity Analysis for minimization of input data dimension for feedforward neural network. IEEE International Symposium on Circuits and Systems (ISCAS '94). London. Vol. 6, 447-450 (1994)
6. de Sá, C.R.: Variance-Based Feature Importance in Neural Networks. International Conference on Discovery Science (DS 2019). Discovery Science 306-315. (2019)
7. Feng, D., Chen, F., Xu., W.: Efficient Leave-One-Out Strategy for Supervised Feature Selection. Tsinghua Science and Technology. Vol. 18 No. 6, 629-635 (2013)
8. Sharma, N., Gedeon, T.: Hybrid Genetic Algorithms for Stress Recognition in Reading. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO 2013). (2013)
9. Hussein, F., Kharma, N., Ward, R.: Genetic Algorithms for Feature Selection and Weighting, a Review and Study. Proceedings of Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA. 1240-1244 (2001)
10. Gamarra, M.R., Quintero, C.G.: Using Genetic Algorithm Feature Selection in Neural Classification Systems for Image Pattern Recognition. Ingeniería e Investigación Vol. 33 No. 1. 52-58 (2013)