A Comparison of Model Architectures for Classifying Genuine vs Posed Anger

Marcus King

Research School of Computer Science
Australian National University
u6978252@anu.edu.au

Abstract. Recognising facial expressions has long played a role in human-human interactions. With the emergence of virtual assistants and other human-computer interactive technologies, recognising genuine and posed facial expressions has utility for improving these interactions. Machine classifiers, such as neural networks, have shown promise in classifying genuine and posed emotion based on physiological indicators but suffer from the common problems of determining the correct model architecture and model size. Therefore, we compare two different types of model architectures, feed-forward neural networks (FNNs) and recurrent neural networks (RNNs), specifically long short-term memory (LSTM). A comparison between the two different model architectures, FNNs using significance pruning and LSTMs, indicates that LSTMs perform slightly better than FNNs in its ability to correctly identify posed anger emotion from individuals pupillary responses.

Keywords: Neural Network · LSTM · Feed-Forward · Deep · Pruning · Significance · Emotion.

1 Introduction

In social interactions humans use facial expressions, among various other modalities, to convey emotion. Different emotions elicit different responses, both conscious and unconscious from the person observing the expression [4]. Furthermore, humans have also learned to use facial expressions and other modalities during social interactions to manipulate and deceive observers, which in some contexts, can be difficult for the observers to consciously detect. Effective and convincing posed facial expressions though, have utility for human-computer interactions, and these interactions could be improved by recognising if a genuine emotional response has been elicited in the observer to virtual stimuli [3]. Therefore, there is value in classifying stimuli as genuine or posed emotional expressions based on an observer's physiological response.

There have been promising results of machine classification that use physiological characteristics to classify whether the facial expressions being observed are genuine or posed [3, 7]. [3] used artificial neural networks (hereafter referred to simply as neural networks) to achieve high accuracy in classifying genuine or posed anger, although, implementing neural networks is not always straightforward and there are many challenges, particularly with selecting appropriate model architectures and associated hyperparameter values. Different types of model architectures such as feed forward neural network (FNN) and recurrent neural networks (RNN), specifically long short-term memory (LSTM) networks, are both applicable to many problems but go about the task in very distinct ways. In particular, RNNs leverage sequential data where input vectors can be of different sizes; whereas, FNNs only work if every input vector is of the same size. To overcome this problem for comparison we use an aggregated summary of the sequential data as input into FNNs. Additionally, hyperparameter values for each architecture is another issue. These values, such as the number of hidden layers and number of hidden neurons, cannot be easily determined beforehand [5] and optimal values for these hyperparameters are likely unique to the problem, although, knowing the optimal values can be a useful starting point for related problems and therefore an approach is required to determine the number of hidden neurons each hidden layer, such as pruning.

Neural network pruning techniques can assist in finding a minimal network size that is still able to solve the problem. Neural network pruning refers to a variety of techniques that remove (prune) neurons or weights from a network that are not deemed important, thus reducing the size and complexity of the network while conserving its performance. Pruning networks are also useful to assist in generalising the model to perform better on unseen data, i.e. avoid overfitting on data that it was trained on [5]. Significance pruning is used to optimise FNNs as it has been shown this method outperforms distinctiveness pruning [9].

The objective of this report is to compare different model architectures to determine whether sequencing effects within raw data outperform FNNs to classify genuine and posed emotion.

2 Method

2.1 Dataset Description

The data was sourced from [3] and was collected to determine whether humans could consciously determine genuine vs posed anger as accurately as their unconscious physiological characteristics. The data records physiological responses for 20 individuals who watched 20 short videos (10 with genuine anger, 10 with posed anger). Genuine and posed anger videos were sourced from YouTube and were differentiated by the video source. The genuine anger videos were sourced from documentaries and live news while posed anger videos were sourced from movies (videos used unknown actors to avoid participants knowing the anger being expressed was posed). Additionally, the videos were selected to control for other factors that could affect the final results, see [3] for additional information on the data collection procedures. The raw data contains left and right pupil measurements for each individual for each video, therefore as each video is of varying lengths the data for each video is of different lengths. Aggregated data was also sourced from [3] and is used to train and evaluate FNNs. The aggregated data contains the following attributes.

- Observer: observer unique identifier, 20 observers labeled O1 to O20.
- Video: video identifying number, labeled F1 to F10 for posed anger videos and T1 to T10 for genuine anger videos.
- Mean: the mean pupillary response value.
- Std: the standard deviation of the pupillary response.
- Diff1: change in left pupil size after watching the video
- Diff2: change in right pupil size after watching the video
- PCAd1: orthogonal linear transformation with first principal component
- PCAd2: orthogonal linear transformation with second principal component
- Label: whether the video displayed genuine or posed anger.

Before constructing a neural network to classify genuine vs posed anger, data exploration was conducted to understand if the task is suited to neural networks and a simpler solution was not present. Therefore, a brief description of the data is provided to better understand the data and determine the correctness of neural networks to the problem.

Raw Data

Interpolation was applied to the raw data to remove eye blinks (zero values) and normalised for each participant using min max normalisation across all that participants data. These values were then averaged across each video for left and right pupil responses before taking the mean difference between both pupils. Finally, these values were averaged again across genuine and posed videos to produce Figure 1. Figure 1 indicates that the mean pupillary response for genuine and posed anger are similar however the posed anger is regularly above the genuine indicating the possibility that a recurrent architecture will be able to distinguish between the two.

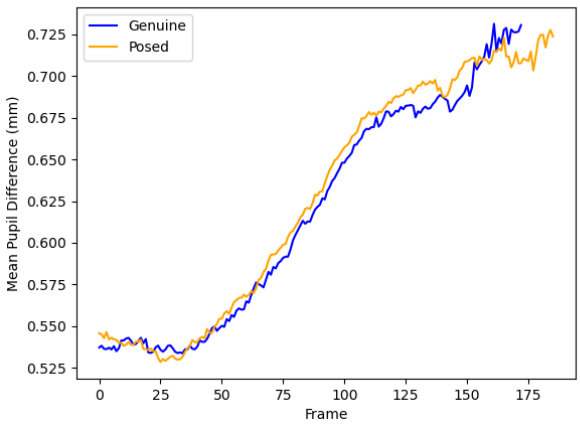


Fig. 1: Mean Pupil Difference for Genuine and Posed Anger

Aggregated Data

As can be seen in Figure 2, the distribution of values for each attribute is similar for genuine and posed anger making it hard to find an easy solution that separates the two classes. However, it is worth noting the attribute PCAd1 is possibly of more importance than other attributes in separating genuine and posed.

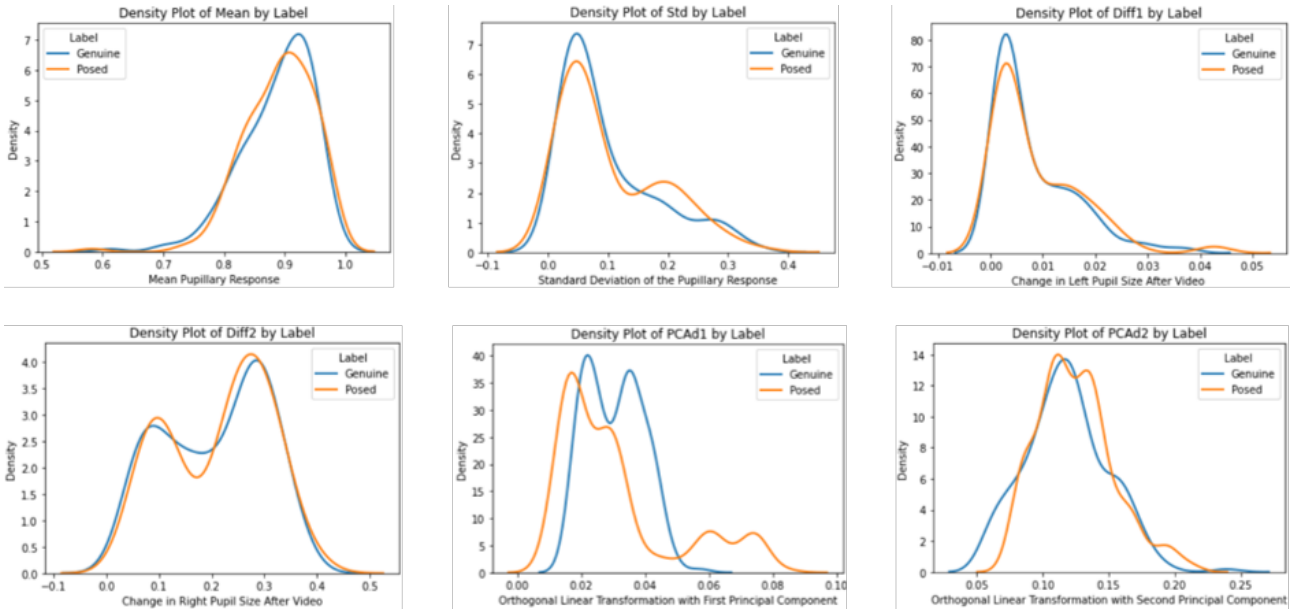


Fig. 2: Density plots of physiological attributes

Furthermore, Figure 3 indicates that the two attributes related to pupillary difference after watching the video cannot separate genuine and posed anger. Comparing these individually against PCA1d we can see a separation between genuine and posed anger further indicating the potential importance of PCAd1 to machine classifiers. This problem does not appear to have a straightforward solution and therefore is suited to using neural networks to solve the problem. Other methods such as K-nearest-neighbour (KNN), support vector machine (SVM), and ensemble methods would also be suited to the problem and were explored by [8] in comparison to neural networks. [8] found that for classifying posed vs genuine smiles, neural networks outperformed other methods in terms of model accuracy.

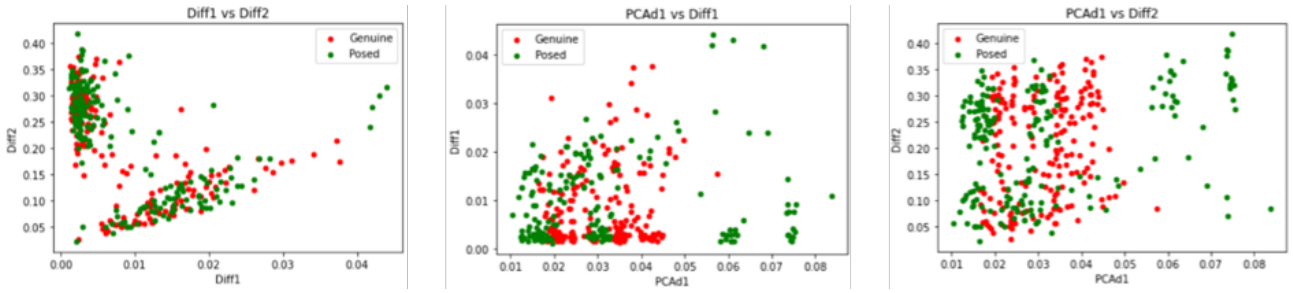


Fig. 3: Scatterplots of select attributes

2.2 Data Preprocessing

Raw Data

The raw dataset included left and right pupil responses recorded from each of 20 participants observing 20 videos, 400 sequences in total. However, in 10 cases, an individual had no data for a particular video and therefore these were removed from the overall dataset. The remaining data also contained zero values, which indicate a participant eye blink. These eye blink values were replaced with linearly interpolated values. The data for each participant was then normalised using min max normalisation across all of that participants data (Equation 1).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

After initial experimentation an additional pre-processing step was added to reduce the size of each sequence. Starting from the beginning the average was taken for every 20 data points to reduce the sequence size. Finally, the left and right pupil responses for a sequence were vectorised to form vector inputs for LSTM networks and split into training and testing sequences using an 80/20 split respectively.

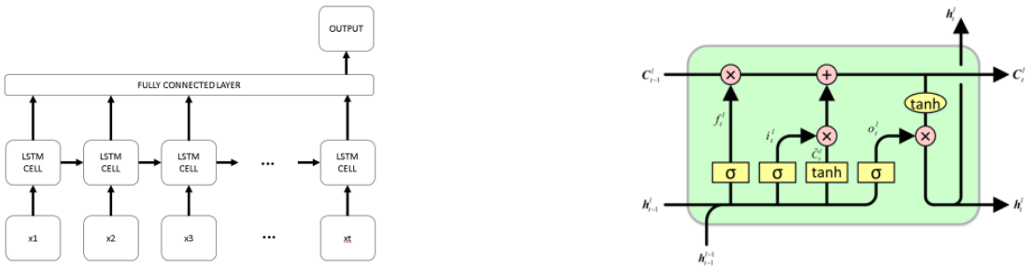
Aggregated Data

The video and observer identifier attributes were removed from the data as they are not needed for the model and will result in the network overfitting on these attributes. The remaining input attributes displayed dissimilar ranges, therefore before training the attributes were normalised using min-max normalisation (Equation 1). This normalisation technique scales the data to the range 0 to 1 and preserves the overall distribution of the attribute. Each record in the data was randomly assigned to one of two categories, training data (80% of the data), test data (20% of the data). The training data will be used to train the neural network, while the test data is used to evaluate the network.

2.3 Neural Network Architecture

LSTM

RNNs are a type of architecture that assist in working with sequential data. The main feature of RNNs is that they aren't memory-less like FNNs. What this means is that the current output of the model is not just dependent on the current input, but also previous outputs from previous inputs as well. However, a major problem for RNNs is the vanishing gradient problem. This is particularly prevalent for long sequences as the network effectively 'forgets' inputs earlier in the sequence [7]. LSTM networks are a type of RNN which solve this problem by replacing neurons in a traditional rnn hidden layer with LSTM cells. This cell determines what information from the input is relevant, what information in its memory is important to keep/forget and what information is output (Figure 4b) [7].



(a) LSTM Model

(b) LSTM Cell

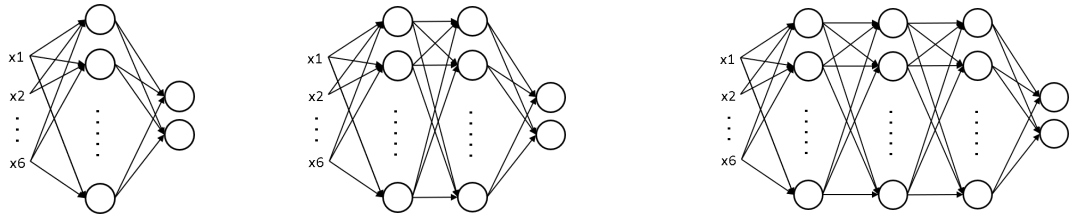
Fig. 4: LSTM Architecture

One LSTM network was implemented in pytorch (Figure 4a) using the following hyperparemeters which were chosen through experimentation. The dimension for each input vector is 2 (left pupil, right pupil) which feeds into one LSTM layer. The output of the LSTM layer feeds a fully connected linear layer with dimension 50. Finally, the model was trained for 500 epochs using a learning rate of 0.001 and optimised using the Adam optimiser and cross entropy loss function. For training, a batch size of 1 is used so that each forward pass through

the model contains one sequence where a sequence contains the left and right pupil data for one participant for one video.

FNN

Three fully connected feed-forward neural networks were implemented, all have an input layer, output layer and one, two and three hidden layers for the three networks respectively (Figure 5). The input layer consists of 6 input neurons (one for each attribute in the dataset), each hidden layer consists of 50 hidden neurons, noting here that the network will be pruned, and 2 output neurons (one for each possible classification, genuine/posed).



(a) 1 Hidden Layer (b) 2 Hidden Layers (c) 3 Hidden Layers

Fig. 5: FNN Models

Pytorch was used for the implementation of the FNNs with the following network hyperparameter settings. The activation function implemented in the hidden layer(s) is the sigmoid activation function which is appropriate for two-class classification problems, with PyTorch automatically implementing the softmax activation function in the output layer. For 2 class problems, the softmax activation function performs the same transformation as the sigmoid activation function.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

As the activation function being employed here is sigmoid, the cross-entropy loss function is used as it is commonly associated with the sigmoid activation function.

$$L(y, t) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_{nk} \tag{3}$$

For the network optimiser, resilient backpropagation (Rprop) [10] is used. Rprop is different than other methods that improve upon backpropagation in that it only uses the sign of the gradient with its main advantage being that it results in faster learning. Finally, the model was trained using 1000 epochs with a 0.0001 learning rate and were replicated 100 times in order to ascertain a better estimate of model performance.

2.4 Feed Forward Network Pruning

After training the network was pruned using significance pruning [1]. The technique determines the significance or importance of each neuron to the model. The significance of each hidden neuron is determined by the sigmoidal activation of the hidden neuron and all its outgoing weights. Hidden nodes that fall below the significance threshold are deemed insignificant and removed from the model. A summary of the pruning process is described below. Note: the algorithm outlined in [1] prunes inputs from the model; however, this part of the algorithm is not considered.

1. For each hidden neuron, h, in hidden layer, l, calculate the sum of all the inputs into the hidden neuron.

$$t_{hl} = \begin{cases} \sum_{p=1}^{P} \sum_{i=0}^{I} x_{ip} w_{ih} & \text{,when l=1} \\ \sum_{h=1}^{Hl-1} f(t_{hl-1}) v_{hl-1,hl} & \text{,when } l > 1 \end{cases}$$

where,

h = hidden node
l = hidden layer
T_{hl} = total sum of inputs in hidden node h of hidden layer l
p = training pattern
P = number of training patterns
i = input attribute
I = number of input attributes
Hl-1 = the number of hidden neurons in the previous layer
x_{ih} = the value of the i^{th} attribute of pattern p
w_{ih} = weight value between i^{th} attribute and hidden neuron h
$v_{hl-1,hl}$ = weight between neuron in previous hidden layer and neuron in current hidden layer

2. The sigmoid activation function is applied to the sum of all inputs into the hidden neuron

$$f(T_h l) = \frac{1}{1 + e^{-T_h l}} \tag{4}$$

3. The significance of a hidden neuron, h, is calculated by

$$s_{hl} = \sum_{k=1}^{K} |f(T_h l) + v_{hk}| \tag{5}$$

where,

s_{hl} = significance of hidden neuron h
k = neuron in next layer
K = number of neurons in next layer
$v_{hl,kl+1}$ = weight between hidden neuron h in current layer l and output neuron k in the next layer

4. Neurons are deemed insignificant if they fall below the average significance value of neurons in the hidden layer

$$s_h l \text{ is } \begin{cases} \text{insignificant if } s_h l < \frac{\sum_{lh=1}^{Hl} s_{hl}}{Hl} \\ \text{significant otherwise} \end{cases}$$

where,

Hl = number of neurons in hidden layer l

5. If accuracy of pruned network falls below desired threshold then cease pruning and retrain network, otherwise retrain network and repeat process of pruning while accuracy remains above desired value. Note: accuracy threshold is set high so that only 1 pass of the pruning process is required.

3 Results and Discussion

3.1 LSTM

The LSTM loss decreased initially before stabilising and failing to decrease any further below 0.5. The model achieved 79% with precision and recall vlaues of 0.87 and 0.9 respectively (Table 2). The balance between and high precision and recall provides promising results that recurrent networks can classify genuine and posed anger. The high precision indicates that when the model classifies a video as posed anger it is right approximately 87% of the time. Similarly, the high recall indicates that the model has correctly classified 90% of posed anger sequences.

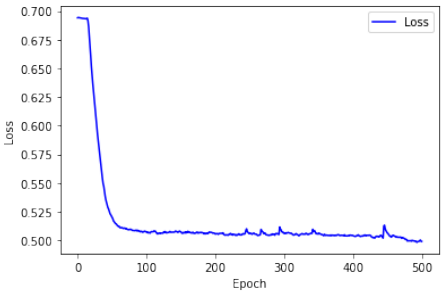
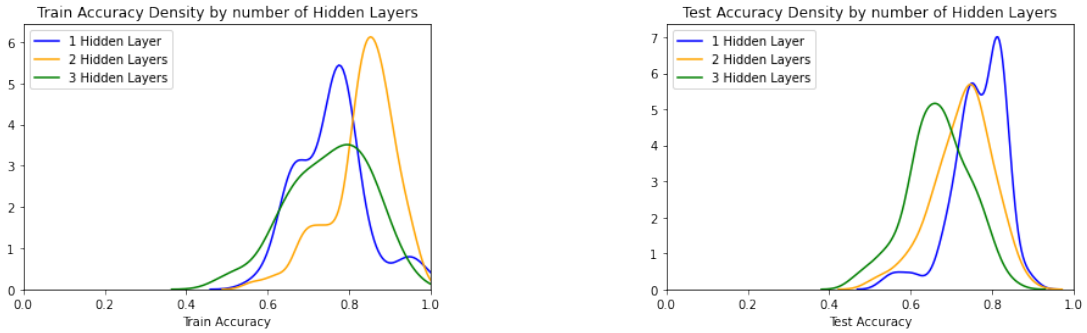


Fig. 6: LSTM Loss

3.2 FNN

Figure 7a displays the training accuracy density for each of the three FNNs. All three models performed well on the training data where the models with 1 and 3 hidden layers performed similarly achieving a training accuracy of 75%. The FNN with 2 hidden layers outperformed its counterparts with respect to training accuracy achieving an average training accuracy of 83%. Importantly, no models appear to be overfitting on the training data.



(a) Training Accuracy

(b) Test Accuracy

Fig. 7: Training and Test Accuracy for FNN Models

Figure 7b displays the testing accuracy density for each of the three FNNs. This indicates that testing accuracy becomes worse as more hidden layers are added to the model where models with 1, 2 and 3 hidden layers achieved average test accuracy of 77%, 73%, 67% respectively.

Table 1 displays the average size of the pruned network for each of the three models. Significance pruning resulted in consistent hidden layer sizes for each respective model. While the results for the FNN with one hidden layer are consistent with [9], the number of pruned neurons from the first hidden layer is not consistent as more hidden layers are added to the model.

Model	Hidden Layer 1 Neurons	Hidden Layer 2 Neurons	Hidden Layer 3 Neurons
FNN (1)	6	NA	NA
FNN (2)	14	11	NA
FNN (3)	12	12	12

Table 1: Pruned FNN Architecture

3.3 Comparison

Overall the LSTM architecture outperformed all FNN models in terms of accuracy precision and recall; however, the computation time was significantly higher for this model. While the FNN model with one hidden layer produced similarly results in terms of accuracy and precision, and in a much quicker time frame, there is an imbalance between precision and recall (Table 2). A low recall indicates this model incorrectly classifies posed anger as genuine anger at a much higher rate than the LSTM model.

Comparison of Models				
Model	Accuracy	Precision	Recall	Time (min)
LSTM	0.79	0.87	0.9	10.00
FNN (1)	0.77	0.85	0.66	0.06
FNN (2)	0.73	0.70	0.73	0.35
FNN (3)	0.67	0.65	0.67	0.74

Table 2: Model Comparison

Overall, LSTMs perform slightly better than single or multi-layer FNNs; however, the computation time is a significant variable that can influence model selection. The deciding factor is that LSTM models have shown to produce models with high precision and recall, that is, when the LSTM classifies a sequence as posed anger it is usually correct, and it correctly classifies most posed anger videos.

4 Future Work

This research compared two different model architectures and showed similar results between those selected. As the sample size for the training is small, future research should further explore the results of RNNs and FNNs on larger datasets to verify the reported results. Additionally, the training on LSTMs converged early and the loss only descreased a small amount from its initial value. Therefore, future research should utilise larger datasets to determine the effect of hyperparameter values, such as the number of LSTM layers, batch size and length of input sequences, on model performance.

References

[1] Augasta, M., Kathirvalavakumar, T.: A novel pruning algorithm for optimizing feedforward neural network of classification problems. Neural processing letters **34**(3), 241 (2011)
[2] Augasta, M., Kathirvalavakumar, T.: Pruning algorithms of neural networks—a comparative study. Open Computer Science **3**(3), 105–115 (2013)
[3] Chen, L., Gedeon, T., Hossain, M., Caldwell, S.: Are you really angry? Detecting emotion veracity as a proposed tool for interaction. In: Proceedings of the 29th Australian Conference on Computer-Human Interaction, pp. 412–416 (2017)
[4] Frith, C.: Role of facial expressions in social interactions. Philosophical Transactions of the Royal Society B: Bilogical Sciences **364**(1535), 3453–3458 (2009)
[5] Gedeon, T.: Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. In: Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, pp. 26–29. IEEE (1995)
[6] Gedeon, T., Harris, D.: Network reduction techniques. In: Proceedings International Conference on Neural Networks Methodologies and Applications, pp. 119–126 (1991)
[7] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735-1780 (1997)
[8] Hossain, M., Gedeon, T.: Classifying posed and real smiles from observers' peripheral physiology. In: Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare, pp. 460–463. (2017)
[9] King, M.: Finding the Right Size: A Comparison of Pruning Techniques to Determine Hidden Layer Size for Classifying Genuine vs Posed Anger. In: 4th ANU Bioinspired Computing Conference. (2021)
[10] Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: IEEE international conference on neural networks, pp. 586–591. IEEE (1993)