Using Bimodal Distribution Removal and Genetic-based Feature Selection on Anger Veracity Recognition

Chaoxing Huang

Research School of Computer Science,
Australian National University
ACT 2601 AUSTRALIA
u6441859@anu.edu.au

Abstract. It is important for a machine to know the anger veracity in human-machine interaction. This paper looks into the problem of applying Bimodal Distribution Removal (BDR) to remove outliers in anger dataset and Genetic-based feature selection (GFS) to select time-series features in training a neural network for anger veracity recognition. Study shows that though BDR method with proper choice of error function does remove outliers, it can negatively affect the classification performance of the baseline model due to the its overkill effect in removing "slow coaches" and cause overfitting. It also shows that Genetic algorithm can effectively improve the classification accuracy. We proposed a two-stream network architecture in this paper to handle the pupillary information from both eyes.

Keywords: Anger dataset · Bimodal distribution removal · Neural network · Genetic algorithm

1 Introduction

The veracity of emotions plays an essential role in human interaction. It influences people's view towards others after observing a certain emotion[10]. One of the important things in human-machine interaction is to let the machine know whether a human's emotion is disguised or genuine. There has been works that look into the problems of using the physiological data of the emotion observer to interpret the emotion of the stimuli. In [7] and [1], a classifier is trained to identify if a person's smile and anger is genuine or posed. Meanwhile, a human thermal data based algorithm is proposed to analyse human's stress in [8].

Neural Network(NN) is able to learn the parameters automatically in back-propagation way and it is used to map the physiological data to the emotion observer's feeling. However, since human beings interact with the environment, it is likely that the collected physiological data in the data-set can be noisy and contains outlier which do not reflect the observer's feeling properly but reflect how other factor in the environment affect the observer. On one hand, outlier can dampen the learning process of the NN on the dataset, since the model need to learn the underlying pattern of the outlier. On the other hand, the model can overfit on the dataset when the training time has to be escalated.Therefore, it is crucial to look into the problem of moving out outlier from the physiological dataset. Outlier rejections has a profound studied history. One of the most classic method is the generative model learning approach[16]. However, this method requires a cumbersome learning process and relies heavily on distribution assumptions. Other works have also been done to detect outlier[6, 4], and Bimodal Dsitribution Removal(BDR) is proposed in [14] to remove the outlier during training process without much human intervention in an adaptive way[19]. Since the physiological data outliers are ususally not obvious to non-expert human, BDR may become a potential choice to tackle this problem. In the original work of [1], it is shown that using pupillary data for anger veracity recognition can significantly improve the accuracy(95%), compared with that of using verbal data(60%). However, the data that it collected may contains environment-affected outliers and they did not consider this kind of scenario. Also, when we take the time-series information into consideration, not all the recorded data from the sensor plays essential roles in classification due to the noisiness and redundancy, which requires feature selection, and the Genetic algorithm provide a way to achieve this . Therefore, we study the effect of applying BDR and Genetic-based feature selection (GFS) [17] on anger veracity recognition in this paper. As for similar application, BDR is applied to the binary diagnosis of the breast cancer in [12].

In this paper, we first tune a baseline NN with one hidden layer by using the compressed version of anger data-set, and then compare the BDR method with the baseline under different settings. Analysis on the comparison of the results are also conducted to provide insight of the effect. Then we apply GFS to the time-series version of the ataset and verify our proposed two-stream model to handle the binocular pupilary information. The rest of this paper is organised as follows: Section 2 introduces the NN architecture, BDR pipeline and the GFS pipeline. Section 3 is about the experiments and its related result. Discussions are also provided in this section. Section 4 includes future work and concludes this paper.

2 Method

2.1 Dataset

In this paper, we use Gedeon's anger dataset[1]. The dataset was collected by displaying 20 video segments to 22 different persons(observers). A sample in the dataset means a video watched by an observer and each of them is labelled with "Genuine" or "Posed". For each of the sample, the observer watch the stimuli's anger expression in the video and the pupillary response of the observers are collected by eye-tribe sensor.The compressed version of the dataset (version-1) has 6 features while the time-series version (version-2) contains the pupillary response from each observers two eye in different time-step as well as the mean statistics .

2.2 Network Architecture

Baseline Architecture In the study of BDR, we adopt a simple fully-connected neural network architecture with one hidden layer with n hidden neurons, which is shown in Fig. 1. Since every data point has 6 features and the problem itself is a binary classification, the input node number is 6 and the output node number is 2. There are three potential choices of activation function in our NN, which are Sigmoid, Tanh and ReLU. We will look into the effect of different choice of n and activation function type in the experiment part. Since this is a binary classification, we choose cross-entropy loss as the loss function, which is expressed in Equation (1).

$$L = \frac{\sum_{i=1}^{N} -(y_i log(p_i) + (1 - y_i) log(1 - p_i))}{N} \tag{1}$$

where p_i is the output probability of the i^{th} sample and y_i is the ground-truth label of either 0 or 1. N is the total number of patterns.

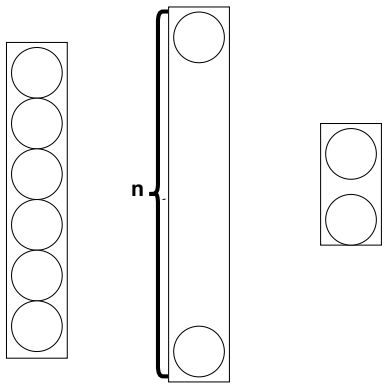

Fig. 1. Network architecture, with 6 input nodes and 2 output nodes. All the neurons are fully connected.

2.3 Two-stream architecture

Inspired by the two-stream architecture in video recognition [3, 13], we adopt a two-stream fully connect architecture in our classification task, which is shown in Figure 2. For every stream, the sub-stream network is the baseline model and the feature vector from the two streams are fused together to a one-layer fully connected layer for final prediction. There are two potential kind of input to the network. The first scenario is, the first stream takes the pupilary temporal data from the left eye and the second stream takes the pupilary data from the right eye.The second scenario is, the first stream takes the pupilary temporal data from the left (right) eye and the second streams takes the pupliary differences data from the left (right) eye. The pupilary difference for each time step is just the data at current time step minus the data at the previous step.

2.4 Data Pre-procsseing and Feature Selection

Data pre-processing When dealing with label "Genuine" and "Posed", we follow the expression of cross enthropy loss and denote "Genuine" as 0 and "Posed" as 1. As for the features in version-1 dataset, the vector video number and the ID are removed since they are irrelevant to our task. The rest 6 features are at different scale, and it is important for the network to treat every feature equally and does not think some of the features

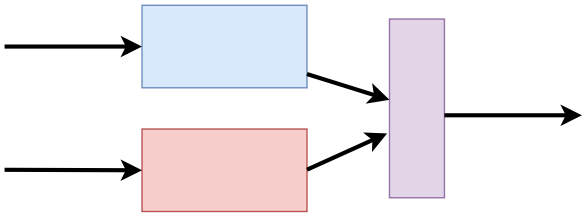

Fig. 2. Two-stream network architecture

with larger range are more important. Therefore, we apply the normalization technique to those features and re-scale them into the range from 0 to 1. It has been studied that normalization is crucial for producing promising result in NN related system[15]. For every feature, we apply the normalization technique in Equation 2.

$$x = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

where x are the feature values, and x_{min} and x_{max} are the maximum value and minimum value of the features respectively.
When it comes to the version-2 dataset, we regard every time-series data of each sample as an input vector to the neural network. To deal with the length varying issue, we use zero padding to pad every feature vector to the same length (186×1).

Genetic-based feature selection The feature selection mask is indicated by a binary vector with the length of the feature-vector (0 for omitting a feature and 1 for keeping a feature). In genetic algorithm, the selection mask is regarded as the chromosome. We first initialize the population as $n+1$, and we adopt a neural network to compute the validation classification accuracy as the fitness value. Note that for different chromosome, the input size of the neural network is different, and thus we are not only doing a feature selection but also conducting a network architecture selection. We adopt a tournament-based reproduction[20], in which we create $n/2$ set of tournament-group, and we randomly choose a fix size of member from the current generation to form the tournament-groups as the population pool for generating off-spring. Note that we actually repeat $\frac{n}{2}$ times of tournament group creation, which means one chromosome can appear in different tournament group. In each tournament-group, two parents are selected by using the selection probability which is obtained by its normalized fitness value in the population (proportional selection). The crossover generates two off-springs by a one-point crossing. Therefore, the tournament-reproduction can generates n off-springs, while the rest one is the chromosome with the highest fitness value in the current generation. Every generated off-spring go through a mutation process to increase the gene diversity. To sum up, the population of each generation retains at $n+1$ while the parents selection in every generation's reproduction need to go through a fierce tournament competition. The pipeline is shown in Figure 3.

2.5 Bimodal Distribution Removal

Algorithm According to [14], as the network starts to learn the underlying features of the patterns, the majority of the patterns in the training set will gain a very small error while those outliers will get much larger error. Therefore, the error of the patterns should form two peaks, with a higher peak representing the majority of the patterns condensing at the low error range and a lower peak representing the outliers condensing at the high range of error. Ideally, the variance of the error should tend to be zero since all the patterns will eventually condense at the higher peak with low error. Therefore, a threshold is computed to determine the omission of the outliers.

When the variance of error v is lower than 0.1, it means the network starts to learn the dataset and BDR can start. It first computes the mean error m as the first threshold, and patterns with error higher than m are grouped into a subset. As for the subset, the subset error mean m_s is computed and the final threshold T is determined by Equation 3:

$$T = m_s + \alpha\sigma_s \tag{3}$$

where σ_s is the standard deviation of the errors in the subset and α is a hyper-parameter ranging from 0 to 1. Patterns with error higher than the threshold will be considered as the outliers and will be removed from the training set. To avoid deleting all the training samples and cause overfitting, the BDR is only applied for every 50 epoch and the training is terminated if v is lower than a threshold v_T. Note the errors are normalized to the range of 0 to 1 for fair comparison.

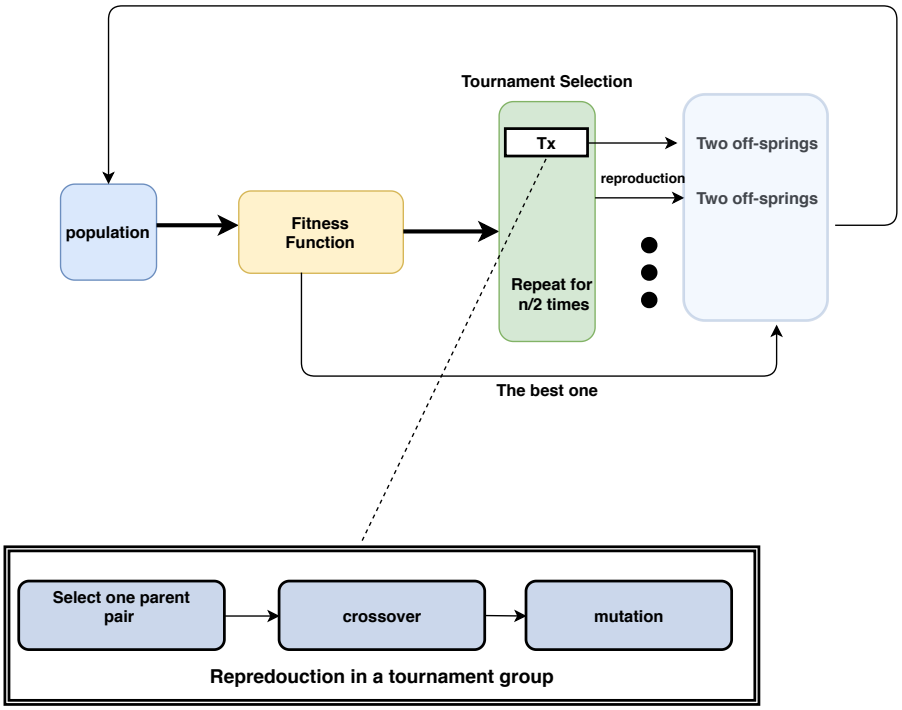

Fig. 3. Genetic-algorithm pipeline

Error function The choice of the error function is essential since it need to simultaneously maintain the bimodel assumption and reflect the True/False intuition of every prediction of the training patterns. One might think of directly using the cross-entropy loss as the error. But since neural network tends to overfit the training data, the numerical value of the errors can all be very small and floating point error may occur if we further apply normalization. Instead, we consider using the highest prediction probability to form the error function. Here we present two potential way of constructing the error function, which are shown in Equation 4 and Equation 5.

$$error = \begin{cases} 1 - P, & \text{if } True \\ P, & \text{otherwise} \end{cases} \tag{4}$$

$$error = \begin{cases} -P, & \text{if } True \\ P, & \text{otherwise} \end{cases} \tag{5}$$

Since the range of the output of Equation 5 is from -1 to 1, we re-scale it to the range from 0 to 1 for convenience. Comparison experiment on these two errors will be shown in section 3.

2.6 Performance metric

Since the BDR has an early stop mechanism, we adopt the test prediction accuracy of the model after the last training epoch as the performance metric, which is shown in Equation 6.

$$Accuracy = \frac{\sum_{i=1}^{N_{test}} \delta(M_t(x_i), t_i)}{N_{test}} \tag{6}$$

where M_t is the model after the last epoch of training and t_i is the ground-truth of the test data. The higher the test accuracy, the better the model.

3 Experiments and Discussions

3.1 Experiment settings

We first shuffle the dataset and randomly split out 80% of the data as training patterns. The rest of the data are for testing. We use Pytorch 1.0[11] to implement the experiment and the environment is on Windows 10. Since the training set is small, we only use an i7-8750H CPU for computation and we adopt batch gradient descent with an Adam[9] optimizer. For the optimizer, the hyper-parameters are: $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The

learning rate is set to be 5e-3 and a weight-decay of 1e-5 is used to prevent overfitting. The iteration number of training is set to be 1000 epochs.

Our experiments are mainly in two folds. All the experiments on the study of BDR are based on the version-1 data-set. While all the experiments of using temporal data and applying GFS are based on the version-2 data-set.

3.2 Baseline model

We train the model under different settings of hidden neuron number n and different activation function type. The result are shown in Table 1.

Table 1. Test accuracy of the baseline model (%)

n	**ReLU**	**Tanh**	**Sigmoid**
10	72.50	62.50	50.00
20	67.50	66.25	65.00
30	72.50	65.00	66.25
40	73.75	66.25	68.75
50	**81.25**	66.25	70
60	**81.25**	67.5	61.25

As is clearly shown, the performance of using ReLU activation function shows significant advantage over those of Tanh and Sigmoid. This is aligned with the knowledge that Tanh and Sigmoid suffer from gradient vanishing problem during training. The results of ReLU and Tanh also reveal that adding more hidden units can genreally improve the performance. Meanwhile, the result of using Sigmoid with 60 hidden units shows that adding more neurons can also cause over-fitting. Noticeably, the model using ReLU with 50 and 60 hidden units achieve the highest accuracy at 81.25%, but they cannot be compared with the result of 95% in [1] since the train split and test split in this paper is different from that of [1]. We also try adding more neurons and layers by using the ReLU activation function, but it turns out the performance are degraded due to overfitting. In the rest of the experiments, we will use the model using ReLU with 60 hidden units as the baseline.

3.3 Bimodal distribution removal

We first compare the outlier removal effect of the two potential error function settings mentioned in section 2.4. After choosing the appropriate error settings, we conduct experiment to see the performance of the network by using BDR with different standard deviation distance α and different training termination condition v_T, which are also mentioned in section 2.4. We also compare the result with the baseline result.

Comparison of different error functions Let us denote the error function in Equation 4 as e_1 and the error function in Equation 5 as e_2. In this group of experiment, α is set to be 0.5 and the training termination condition v_T is set to be 0.01. The results are shown from Fig 4 to Fig 7.

At the first glance of Fig .4 and Fig .5, the error distribution of using e_2 is more close to the bimodal assumption in BDR. All patterns are condensed at the two very far end of the histogram in Fig. 4. In contrast, the distribution in Fig .5 does not fully comply to the bimodal assumption. Many of the patterns are distributed in the middle part of the histogram and the left peak are mowed down as training goes by. We can further visualise the effect in Fig. 6and Fig. 7. When using the e_2, the BDR does trigger the variance to reduce sharply and it makes the training to terminate early. According to [14], the BDR usually takes place between 200 epoch to 500 epoch and the curve in Fig.6 indeed reflects this attribute. On the other hand, the curve in Fig. 7 does not show a downward trend and the early stop is not triggered. Therefore, e_2 can be more suitable for being the error function. We will use e_2 in the rest of the experiments.

Performance of NN by using BDR under different standard deviation distance and early stop condition In this part, we present the performance of BDR by choosing different α and v_T. The result are shown from Table. 2 to Table. 3

It seems the result in Table 2 reject the statement that BDR can improve the network performance by removing outliers. None of the settings achieve an accuracy of higher than that of the baseline result(81.25%). One possible reason is that those "slow coaches"[14] are also removed by BDR and thus cause overfitting. This can be further explained in Table. 3 and Fig. 8. In Table 3, the lower the v_T, the more patterns are removed. And

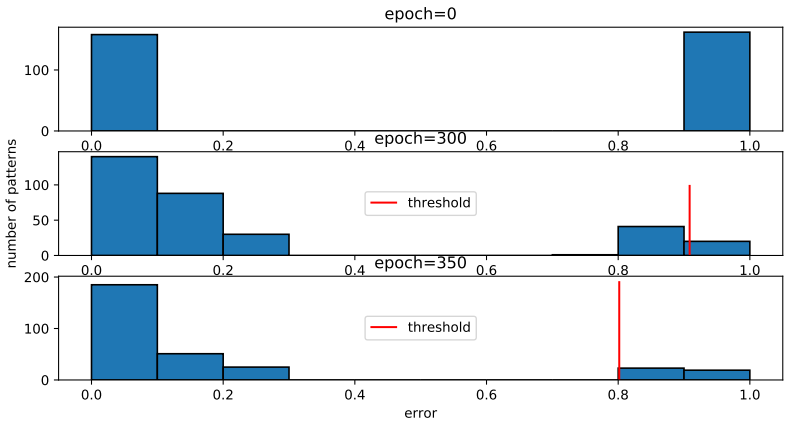

Fig. 4. e_2 distribution at different epoch. The top figure is the distribution at epoch 0.

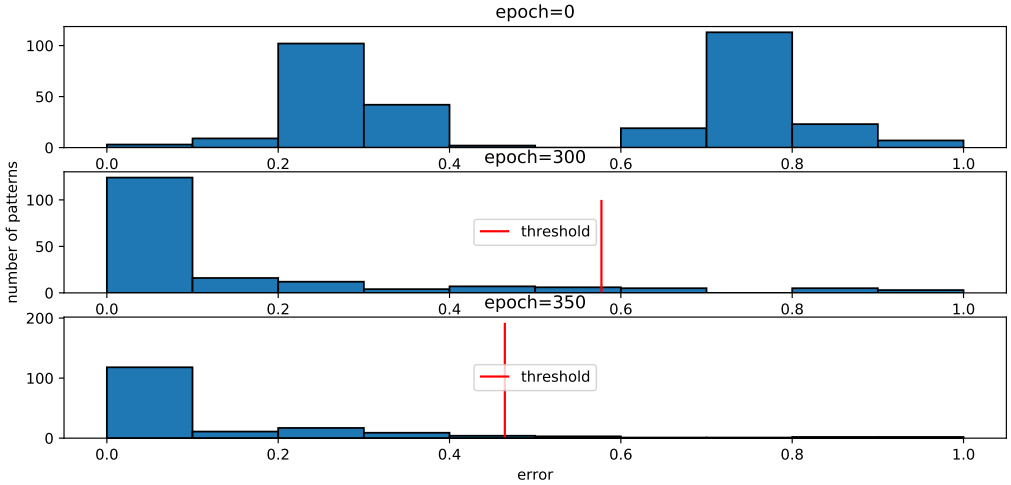

Fig. 5. e_1 distribution at different epoch. The top figure is the distribution at epoch 0.

if we look at the case of $v_T = 0.01$, it has the least number of left patterns and it achieves the lowest accuracy, which is aligned with the intuition that a small training set can cause overfitting. It should be noted that though the case of $v_T = 0.08$ has more patterns left than that of $v_T = 0.05$, it actually has a lower performance. This can be explained by a possible fact that those real outliers are removed when $v_T = 0.05$ while not removed when $v_T = 0.08$. In Fig. 8, there is a sharp increase in training accuracy curve while that of test does not change to much. Such an enlarged gap between the train curve and the test curve is a sign of overfitting.

Slow coaches removal of BDR To further investigate the overkill effect of removing "slow coaches" of BDR, we conduct an experiment of artificially constructing outliers. Among the 320 patterns, we manually set the feature entries of some of the patterns as -100. Since the feature values of all the other patterns are at the range from 0 to 1, those features with negative hundred values are absolutely wrong patterns. We apply BDR with $\alpha = 0.1$ and $v_T = 0.05$under different number of outliers and the removal effect can be seen in Table 4.

As we can see, the removal of the outliers is not satisfying, since only a minority of artificial outliers are removed while the majority of the removed patterns are not the outliers with negative hundred values but those

Table 2. Performance of BDR by choosing different α and $v_T(\%)$.

v_T \ α	0.1	0.4	0.6	0.9
0.01	72.50	70.00	70.00	72.50
0.05	73.75	70.00	71.25	76.25
0.08	72.5	70.00	71.25	75.00

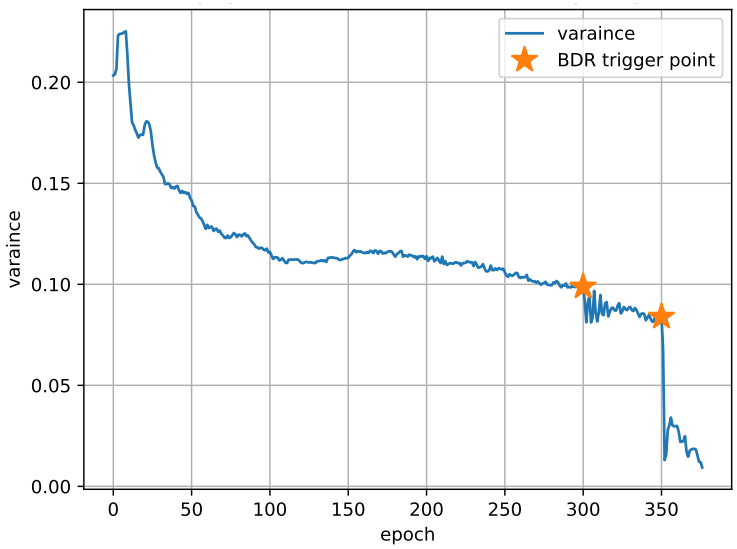

Fig. 6. e_2 variance change according to epoch

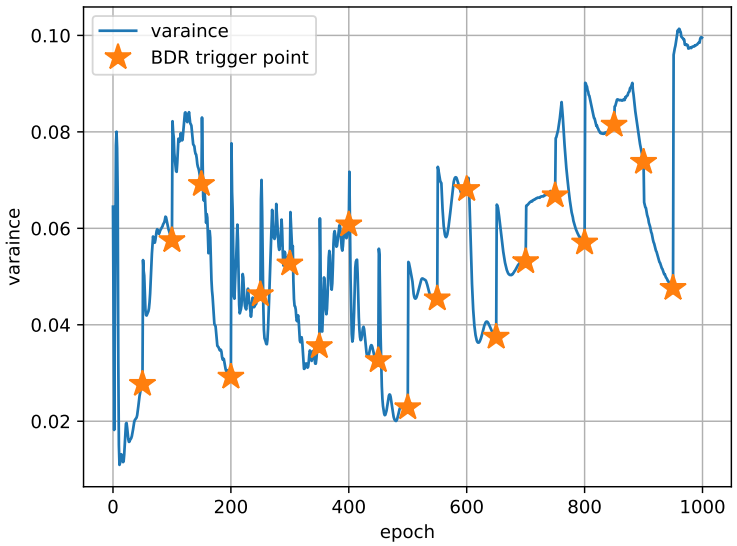

Fig. 7. e_1 variance change according to epoch

original patterns. Therefore, we can say the what cause the bad performance of BDR is its overkilled attribute of deleting "slow coaches".

3.4 Experiments on version-2 dataset

Classification without feature-selection We first conduct experiment on training the fully-connected classifier without applying GFS on the temporal features. We compare the results of different input settings of both single-stream and two-stream model, and the results are shown in Table 5. The hidden layer number is still 6. Note that in the version-2 data-set, several samples are left blank and they are removed from our experiment. For fair comparison, we also omit those samples in the version-1 data-set and re-train our baseline model. The train-test split of version-2 data-set is the same as that of version-1.

As is clearly shown, using detailed temporal data can significantly improve the prediction performance than that of using version-1 data. This is because our neural network has more parameters in the input layer and the fitting ability is improved. Besides, we can also notice that using information from two eyes achieve a better performance that purely using information from one eye, and this is aligned with our human's daily intuition. In fact, the two-stream double-eye model gets the highest performance of 92.31% accuracy. It should also be noted that taking the diameter differences into consideration does not promise an improvement, this may due to the fact that the diameter change in every time-step is very small and does not provide much useful information.

Table 3. Number of patterns left by choosing different α and v_T in BDR.

v_T \ α	0.1	0.4	0.6	0.9
0.01	258	261	261	269
0.05	278	261	277	274
0.08	278	298	277	302

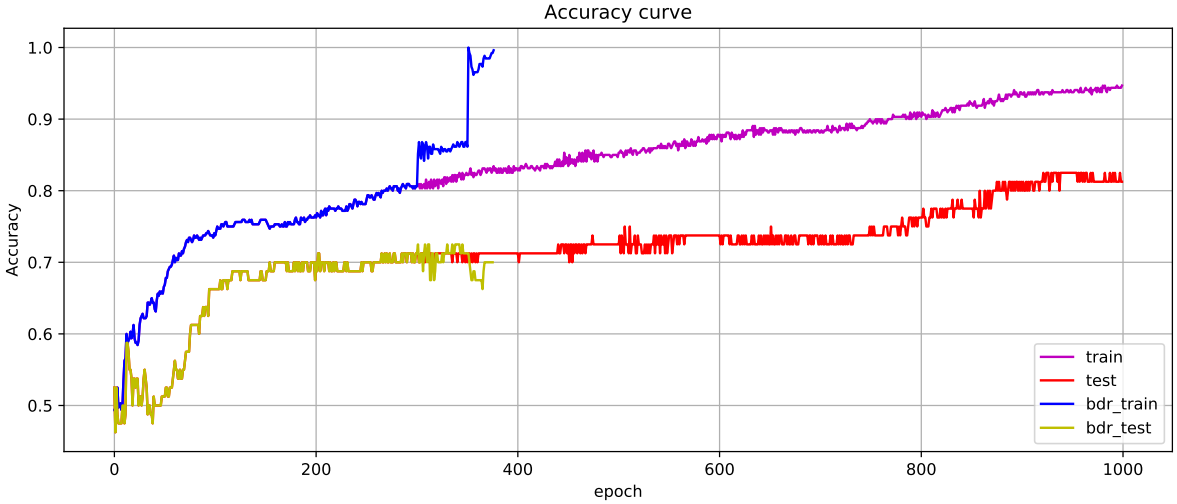

Fig. 8. Accuracy curve comparison

Table 4. Results of constructing outliers

Number of outliers	BDR Accuracy(%)/ Patterns removed/ Outlier removed	Baseline Accuracy(%)
10	61.25/73/3	67.5
20	66.25/69/3	65
12	61.25/85/3	76.25
5	58.75/86/1	71.25

Effectiveness of GFS We study the effect of applying Genetic feature selection to the model. During the feature selection stage, 80% of the training data is used for training while 20% of the training data is used for validation to compute the fitness value. After selecting the best chromosome, we train our model on the entire training set. For every tournament group, the member number is set to be 9. The generation number is set to be 10. We compare the result of using different population size and mutation rate, and they are shown in Table 6.Note the model in this experiment only takes the left-eye pupilary temporal data as input to a single stream fully-connected layer.

As we can see,by using the GFS method, the single-stream model with left eye input can achieve a better performance than that of not applying feature selection. Moreover, we can see that the Genetic algorithm actually abandon a vast number of features in the time-series vector.It reveals an underlying drawback that using zero padding to fill up the length of those "short" vector does create feature redundancy. Lastly, it can also be seen that a small mutation rate can get a better result than that of large mutation rate. This is because a large mutation result will result in over-explore in the solution space. We can also see that increasing the population number does not necessarily enhance the model training since the tournament selection is very competitive and our population number is fixed during each generation.

4 Conclusion and Future Work

This paper looks into the problem of applying Bimodal Distribution Removal and Genetic feature selection on the anger dataset. From the experiment results, it can be concluded that BDR can negatively affect the training of the model on the anger dataset since it cause overfitting. It can also be seen that BDR has overkill effect of moving slow coaches and it is not sensitive in identifying the true outliers and slow coaches. However, it does show outlier itself can negatively affect the performance of the model and it is necessary to remove them. As for result on version-2 data-set, we demonstrate that using two-stream model by taking the binocular

Table 5. Test accuracy on version-2 dataset (%)

Input	Test Accuracy
Double-eyes (two-stream)	**92.31**
Left-eye (single-stream)	88.46
Right-eye (single-stream)	88.46
Left-eye+Left-differences (two-stream)	87.18
Right-eye+Right-differences (two-stream)	89.74
Baseline	74.36

Table 6. Test accuracy of the baseline model (%)

Population number	Mutation rate	Features remaining number	Accuracy
21	0.2	90	89.74
21	0.001	92	91.03
51	0.2	100	88.46
51	0.001	94	91.03
101	0.2	88	91.03
101	0.001	88	91.03

information benefits the prediction, it can also concluded that applying Genetic-based feature selection can effectively improve the model performance and remove redundant features.

One of the problem of BDR is that it removes the patterns permanently from the dataset once those patterns are removed. Therefore, future work can look into the problem of adaptively adding back some of the patterns during training to avoid overfitting. Besides, the performance of BDR can be affected by the choice of the hyper-parameter α and v_T, since we need to remove patterns during training and training itself is a dynamical process, one might question if we can make those two values as learn-able so the removal of the outliers can become adaptive. Furthermore, the effect of applying BDR on large scale dataset remains unknown. Future work requires research on applying BDR on large scale dataset like those standard computer vision benchmark datasets[2].

As for the model that uses time-series data, our fully-connect model requires zero padding to deal with vary length data, which create redundancy and reduce flexibility. Therefore, it is worth looking into the method of applying RNN/LSTM [5] or Transformer model [18] in the future.

References

1. Chen, L., Gedeon, T., Hossain, M.Z., 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)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
3. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)
4. Gedeon, T.D., Wong, P.M., Harris, D.: Balancing bias and variance: Network topology and pattern set reduction techniques. In: International Workshop on Artificial Neural Networks. pp. 551–558. Springer (1995)
5. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with lstm (1999)
6. Hawkins, S., He, H., Williams, G., Baxter, R.: Outlier detection using replicator neural networks. In: International Conference on Data Warehousing and Knowledge Discovery. pp. 170–180. Springer (2002)
7. Hossain, M.Z., 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)
8. Irani, R., Nasrollahi, K., Dhall, A., Moeslund, T.B., Gedeon, T.: Thermal super-pixels for bimodal stress recognition. In: 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 1–6. IEEE (2016)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Knutson, B.: Facial expressions of emotion influence interpersonal trait inferences. journal of Nonverbal Behavior **20**(3), 165–182 (1996)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
12. Quan, K.: Bimodal distribution removal and genetic algorithm in neural network for breast cancer diagnosis. arXiv preprint arXiv:2002.08729 (2020)

13. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
14. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249–254. Springer (1993)
15. Sola, J., Sevilla, J.: Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on nuclear science **44**(3), 1464–1468 (1997)
16. Svensén, M., Bishop, C.M.: Pattern recognition and machine learning (2007)
17. Vafaie, H., De Jong, K.A.: Genetic algorithms as a tool for feature selection in machine learning. In: ICTAI. pp. 200–203 (1992)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
19. Wong, P.M., Gedeon, T.D.: A pattern adaptive technique to handle data quality variation. Neural processing letters **10**(1), 7–15 (1999)
20. Yang, J., Soh, C.K.: Structural optimization by genetic algorithms with tournament selection. Journal of Computing in Civil Engineering **11**(3), 195–200 (1997)