Bidirectional Long Short-Term Memory with Bimodal Distribution Removal as an Anger-Veracity Estimator

Weijie Tu

Australian National University, Canberra, Australia
u6261186@anu.edu.au

Abstract. Facial expressions play an essential role in human interaction. Among all emotions, anger is the most intense one carrying hostile response to provocation and behaviours. Hence determining the veracity of an individual's displayed anger is important for human-machine interaction. In this context, automatic emotion recognition becomes an active research topic in early eras. In comparison to other techniques, bidirectional Long Short-Term Memory (Bi-LSTM) yields satisfactory results on processing time series data. This paper describes experiments on implementing Bi-LSTM with Bimodal Distribution Removal (BDR) as an anger-veracity estimator. The result shows the effectiveness of Bi-LSTM and suggests BDR stabilises the model and shows its potential on larger dataset. Lastly, a proposed variant version of BDR is more generalised and maintains the ability of original version.

Keywords: Bidirectional Long Short-Term Memory · Bimodal distribution removal · K-fold cross-validation · Time Series Classification · Pupillary Response

1 Introduction

Interpersonal human communication not only includes verbal information but also contains gesture, intonation, tone of voice and emotions. Those nonverbal expressions convey feelings and reflect one's mental states. Many researchers, especially psychologists, intend to understand human emotions and develop principle concepts in the area of human-computer interaction. Ekman and Freisn are pioneers identifying six basic emotions, including anger, fear, disgust, joy, surprise and sadness, as universal emotions among humanity [11].

In this case, a number of projects aim to design algorithms to improve the accuracy of emotion recognition. With regard to recognition accuracy, correctly determining the veracity is a significant factor. In early stages, researchers payed more attention on the models built on verbal responses of participant. The accuracy is approximately 60%, which is moderately better than gambling with accuracy of 50%. Recently, a model trained by participant's pupillary responses achieves estimation accuracy up to 95% [8]. In this case, classifiers trained by pupillary response attracts increasing attention.

The are a number of effective classification techniques. Logistic regression, a traditional machine learning algorithms, yields encouraging results when classifying objects such as breast cancer diagnosis [24]. However, when it comes to dealing with complex data, logistic regression is unreliable. It assumes that the linear relationship between inputs and target, and this constraint makes this model inapplicable in many cases [22]. Artificial neural network tackles the dilemma of Logistic regression. It is able to deal with non-linearity of data and target, but it fails to handle time-series data and variable length data. Therefore, inspired by the model mentioned in [8], this paper constructs a Bi-LSTM network to estimate the genuineness of displayed anger. LSTM is a Recurrent Neural Network (RNN) with memory units to deal with the vanishing gradient problem of traditional RNN [7]. Bi-LSTM further takes both the input sequence and its reverse to fully understand question. It is well-suited for problems including classification, processing and prediction based on time series data.

In addition to the choice of classifiers, some techniques are introduced to improve accuracy. Bimodal distribution removal is an algorithm capable of detecting outliers and preventing overfitting by early stopping [18]. K-fold cross-validation is an unbiased technique famous for increasing the utilisation ratio of data, this paper uses it to evaluate model performance.

In following sections, we will depict the architecture of neural network and introduce techniques to implemented during experiments. We then explain the evaluation method and compare it with other evaluation standards. After analysing the evaluation method, we experiment about the effects of data processing methods, k-fold cross-validation and BDR on model performance. The results provide evidence that the proposed data processing technique is beneficial, k-fold cross-validation indeed measures the performance of neural network unbiasedly and BDR increases the stability of model.

2 Methodologies

This paper formulates the anger-veracity estimation problem as a classification task. We firstly design a neural network and implement BDR to improve performance. The idea behind this paper is to investigate the performance of neural network, the utility of data preprocess method, BDR and k-fold cross-validation. The flow chart of overall classification tasks is shown in Figure 1. The design of this section follows the whole procedure drawn in figure. The whole procedure of classification are mainly divided into three parts, which are preprocessing, using 5-fold cross-validation to evaluate and select models and final classification. We then use BDR to denoise data to seek further improvements.

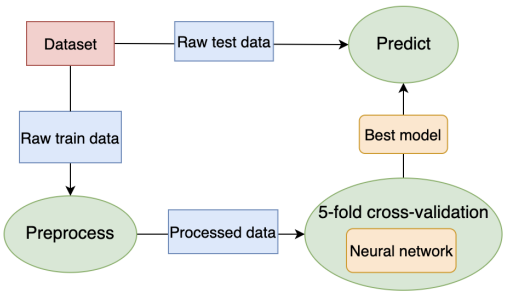

Fig. 1: Flow chart of all tasks

2.1 Dataset

This dataset is data derived from the one in the experiment conducted in [3]. In their experiment and other similar research, models based on these statistics yield satisfactory results. In this case, we construct a new model as a further study to investigate the significance of pupillary response on anger-veracity estimator.

This dataset contains time series data recording the diameters of 20 participants' pupils when they were watching 20 videos with time step of $\frac{1}{60}$ second. Samples distributes evenly, which means there are equal number of samples labelled as genuine and posed. Balance of two classes helps us to build a less biased model. Besides, 20 videos are of different lengths and the 0 occurs if the participant was blinking.

In addition, we further construct a dataset which contains processed data from this dataset to build an orthodox Artificial Neural Network for comparison with this model. This dataset consists of 400 pieces of data with 8 features. These features include participant ID, video ID, and some real-value statistics from pupillary response.

2.2 Data Preprocessing

Since each column of dataframe represents the diameter of participant's pupils, we consider each column as a time-serial data. In terms of zero, there are a few methods to implement. Firstly, we replace the zero values by the data from the other eye. For instance, if the participant P_1 for video F_1 for 10^{th} second is 0, then it will be filled with the corresponding value from right eye. When both two data are 0, we implement a self designed algorithm. The formula is given by:

$$\frac{X_i}{X_i + \sum_{n\neq i}^{N} X_n} = \frac{mean(X_i)}{\sum_{n}^{N} mean(X_n)} \tag{1}$$

$$\Rightarrow X_i = \frac{mean(X_i)}{\sum_{n}^{N} mean(X_n) - mean(X_i)} \cdot \sum_{n\neq i}^{N} X_n \tag{2}$$

X_i represents the value to replace 0 and $mean(X_i)$ is the mean value for this column. The idea behind the method is that we recover the current state of data by reading other participants' states. To be noticed, we will solely count the zero-free rows so as to reduce the influence of zero to minimum. Since reactions from participants are similar for each video, we implement this method dataframe-wise. After processing zero values, it is generally followed by normalisation. Sola and Sevilla [19] demonstrated the significance of normalisation on convergence speed and result quality. Jayalakshmi and Santhakumaran [9] supported that normalisation helps to increase the reliability of backpropagation in a gradient-based algorithm. Additionally, we find the pupil size of each participants are significantly different. The main idea of the research is to change of pupil size with respect to

time. We therefore conduct standardisation before training model. There are two methods can be utilised before training, which are normalisation and standardisation. The equation is shown as:

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

$$Y = \frac{X - X_{mean}}{X_{std}} \tag{4}$$

In formula, Y is the normalised data and X is the original one. In this paper, standardisation is chosen. The reason is that normalisation locates data within $[0, 1]$ while standardisation can further form data points to be normal distributed. At the same time, standardisation maintains the nature of each statistic except magnitude. Neural network is essentially a set of weights reflecting relationship of between attributes of statistics and targets. This implies attributes with greater magnitude tend to exert stronger influence on network. Neural network is supposed to learn the significance of each attribute according to its designed algorithms other than preassigned magnitudes. Thus, restricting attributes within a ranges is desirable and beneficial.

2.3 Bidirectional Long Short-Term Memory (Bi-LSTM) Architecture

Human brain can be depicted as a complex system or network [1], it can understand cognitive and non-linear variables [2]. Vaguely inspired by biological neural network constituting animal brain [4], neural network is introduced and constructed to perform tasks. Such system is capable of handling complex examples by understanding and learning features. It is able to deal with data regardless of quantity and dimensionality. In addition to this, it generally learns the features without studying task-specific rules and prior knowledge.

As the data are time series and each video is variable length, we construct deep learning model to improve result quality. In this case, Bi-LSTM is the optimal choice. The basic structure of Bi-LSTM is shown in Figure 2:

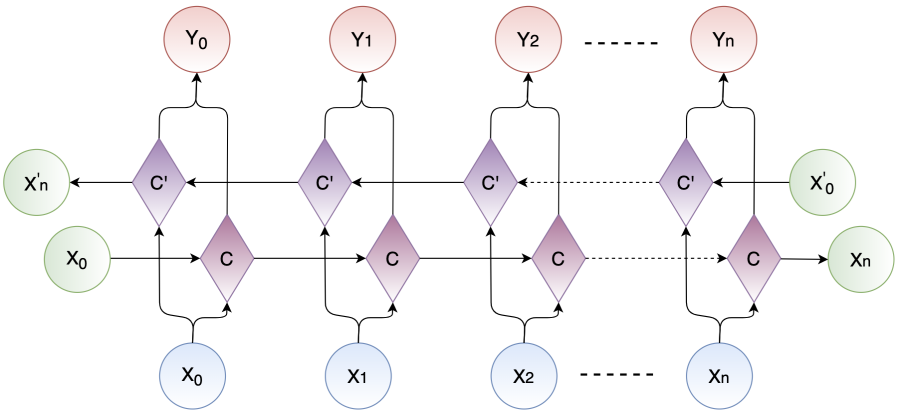

Fig. 2: Implicit Structure of Bidirectional LSTM

This figure shows the basic structure of one layer Bi-LSTM. The final model in experiment is two-layers Bi-LSTM. The reason to utilise a bidirectional model is that the bidirectional model can better understand the feature of sequence as it knows the information before and after the current state [17]. Within Bi-LSTM, dropout is also used to prevent overfitting and improve time efficiency [20]. Lastly, after processed by Bi-LSTM, data then flows into a linear layer to generate labels.

2.4 Loss function and Optimiser

Loss function is a method to calculate dissimilarity between predictions and real labels. In this model, we employ cross-entropy loss function, as it generally performs better results than other methods on classification [21]. The calculation is formulated below.In the function, y_n is the output of neurons and $\hat{y_n}$ is a value returned by sigmoid function.

$$L(w) = -\frac{1}{N}\sum_{n=1}^{N}[y_n log(\hat{y_n}) + (1 - y_n) log(1 - \hat{y_n})] \tag{5}$$

Adam, an algorithm optimising first-order gradient of stochastic objective function. It is straightforward, memory-efficient and well-suited for problems [10]. Unlike other optimisers, Adam generates more intuitive interpretation

of hyperparameter and requires less tuning, which is time-saving for adjusting parameters. However, it is shown that Adma fails to converges to optima when it comes to a simple convex optimisation problem [15]. This is caused by the exponential moving average in algorithm setting, so we employ the variant version 'amsgrad' proposed in [15]. In addition, we further use adamW [13] to repair weight decay regularization in original Adam.

2.5 Mini-batch Stochastic Gradient Decent and Pack Padded Sequence

Mini-batch Stochastic Gradient Decent is a method updating gradients in training. Generally, algorithms update weights in neural network after training all data. This method splits a cluster of samples into small batches. It then updates weights according to loss of each batch. This step accelerates the convergence of training and tackles issues arisen by local minimal [12]. In addition, Bi-LSTM generally takes a large amount of time to train on sequences, Mini-batch is capable of drastically shortening this procedure.

Since we have variable length sequences, mini-batch with variable length of data is not supported by pytorch. Besides, Bi-LSTM uses the reverse of sequences, so it is unreasonable to fit padded data into training. We therefore introduce zero padding for each sequence to ensure sequences in batch with same length, and use 'pack padded sequence' to omit padded zeros when feeding into model.

2.6 Bimodal Distribution Removal

BDR is an algorithm removing outliers introduced in [18]. It firstly collects the dissimilarity of predictions and real labels and computes the variance. Then, we launch the process when $variance < 0.1$. as low variance means neural network has partially learnt data. It relies on mean loss to identify outlier candidate. The threshold condition of deletion is formulated as below.

$$e_i > Mean(L) \tag{6}$$

$$e_i > Mean(L_c) + \alpha \cdot Std(Loss_c) \tag{7}$$

$Mean(L_c)$ is the mean loss of all candidates, and $Std(L_c)$ is standard deviation. α is a constant whose range is $(0, 1]$. This algorithm basically uses statistical features to navigate outliers and carry on deletion. Bimodal distribution of error frequency implies noises are stopping models from learning the general representation of data. We therefore need to remove them so as to reach convergence and redirect model onto the right path. In general, real life information follow normal distribution. A smaller amount of data can be identified as noise if they lead distribution to be bimodal. This phenomenon further reasons the rationality of BDR.

2.7 Hyperparamters Setting

Hyperparameters are parameters that neural network is unable to learn as training proceeds. Capability of model varies with respect to different configurations of hyperparameters [23]. In this paper, we have to choose the settings of both Bi-LSTM and BDR. We follow One-factor-at-a-time Method by setting random seed to change one independent variable at a time. After optimisation, we attain a series of optimal or suboptimal answers. Bi-LSTM has two layers with 32 hidden cells and dropout rate of 0.3, and there are 64 instances in each batch. Training epoch is 100 and the constant α in BDR is 0.2. As for evaluation, fold number is set as 5.

2.8 Evaluation method

We execute two methods, k-fold cross-validation and Matthews correlation coefficient (MCC), to score models. K-fold cross-validation generates multiple models, and we select the best version whose score is computed according to the formulae of MCC.

5-fold Cross-Validation K-fold cross-validation is a statistical technique to assess model performance. It deals with the issues arisen by randomisation. It is generally hard to assess the performance, because randomly splitting dataset causes trained model erratic. The classifier can be biased to one class, because training set can be imbalanced between two classes due to randomisation. Cross-validation addresses it by dividing samples into folds thus generating k models. Each data point fully engages in the process of train-test loop without being 'wasted' in test set. All models are assessed by evaluation method, and we then select the model with highest score. In this experiment, we employ 5-fold cross-validation. 5-fold, as illustrated in [16], is reasonably less unbiased and time-saving. It, meanwhile, maintains the ratio between train-test set as 0.8, which is empirically preferred.

Table 1: Confusion Matrix

	Class: Genuine (Actual)	Class: Posed (Actual)
Class: Genuine (Predicted)	True Positive (TP)	False Positive (FP)
Class: Posed (Predicted)	False Negative (FN)	True Negative (TN)

Matthews correlation coefficient It is important to evaluate model capability. We take both MCC score to measure functionality of models. To begin with, we introduce the following confusion matrix:

In predictive analytics, a confusion matrix is a 2 by 2 table counting the number of TP, TN, FP and FN.

(1). True Positive: Data classified as Genuine with originally labelled with Genuine.

(2). False Negative: Data actually belonging to Genuine while being misclassified.

(3). False Positive: Instances meant be Posed but mis-identified.

(4). True Negative: Instances with prediction aligned with actual class of Posed.

In early age studies, we use $F1$ to evaluate performance. It is computed as following. Precision is the fraction of True Positive cases among all instances predicted as positive class. Recall is the fraction of True Positive instances among total number of relevant samples. We then further proposed F1 score. This score is the harmonic mean of precision and recall. However, David Hand and others criticised F1 as it gives assign precision and recall with equal weights.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{10}$$

In addition, F1 and Accuracy may mislead researchers by providing over-optimistic results. When class sizes are imbalanced, neural network is likely to attach a larger bias to the majority class. The classifier labels each test instance by the dominant class regardless actual features of them. This causes an illusion that accuracy is effective and intelligent as most of instances are identified as the dominant class. Furthermore, by conducting 5-fold cross-validation, groups are randomly generated. It is not necessarily guaranteed that instances are uniformly distribution in each fold.

In order to deal with above situations, Matthews correlation coefficient (MCC) is introduced. It is generally regarded as a balanced measure which maintains functionality even if classes are of significantly imbalanced sizes [14]. The function is given as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

As a correlation coefficient, MCC calculates the geometric mean of regression coefficients to avoid accuracy-illusion [5]. It comprehensively takes both classes into consideration. As both classes matter in MCC, it avoids tyranny of the majority.

3 Results and Discussion

In this part, we mainly rely on One-factor-at-a-time method. We will discuss three topics: 1) Investigate the effectiveness of self-defined zero-value processing method 2) Test if 5 cross-validation evaluates the results better 2) Is BDR effacious? Does it successfully remove outliers? 3) Evaluate if Bi-LSTM preforms better than original LSTM and compare it with other models including logistic regression and orthodox neural network.

3.1 Effectiveness of Self-defined zero-value processor

Since the final model in this paper has accuracy around 95%, it is hard to detect the difference between each method. In order to better illustrate effects of each method, we further construct a less accurate model. This model is a basic LSMT without any other techniques. Three methods include: keep zero values (Keep), Linear Interpolate (LI) and self-designed one (SD). The performance is measured by MCC. According to the table, it can be seen that the SD performs moderately better than LI and Keep method for advanced LSTM model, and

basic model attains more obvious improvements by SD. The reason why PD performs better might be that this research focuses on the change of pupillary response. LI sometimes introduces additionally increase or decrease on data change. In terms of stability, standard deviation of five times of training is computed. SD is less stable as this method leads data more artificial, so that further studies about variant form of SD can be conducted.

Table 2: Comparison between with/without cross-validation

	Basic LSTM			Advanced LSTM		
Trial	Keep	LI	SD	Keep	LI	SD
1	0.84	0.86	0.90	0.68	0.70	0.71
2	0.86	0.87	0.87	0.63	0.67	0.64
3	0.85	0.85	0.91	0.57	0.63	0.82
4	0.84	0.85	0.89	0.67	0.65	0.65
5	0.84	0.88	0.91	0.63	0.66	0.71
6	0.83	0.88	0.87	0.64	0.68	0.67
7	0.85	0.87	0.88	0.67	0.67	0.65
Mean	0.844	0.865	0.89	0.64	0.67	0.70
Std	0.009	0.010	0.016	0.035	0.020	0.058

3.2 Benefits of 5-fold Cross-Validation (5-CV)

To start with, we reason about the necessity of 5-fold cross-validation. In advance to testing, we dismiss this algorithm and compare results of the version equipped with K-fold. As for other factors, we assign all parameters with values stated in subsection 2.7 and implement without BDR. Table 3 illustrates the ability of evaluation method.

Table 3: Comparison between with/without cross-validation

	With 5-CV		Without CV	
Trial	MCC	Acc	MCC	Acc
1	0.90	95%	0.89	92%
2	0.91	95%	0.86	93%
3	0.89	95%	0.81	90%
4	0.87	94%	0.85	92%
5	0.92	96%	0.88	94%
6	0.90	95%	0.90	95%
7	0.90	95%	0.89	94%
mean	0.90	95%	0.87	93%
Std	0.015	0.53	0.029	1.552

From Table 3, the version without 5-CV reveals the instability of randomisation. It is potential to produce a high-quality estimator but it is of low probability. The standard deviation indicates that, without CV, evaluation method is biased and less stable. In contrast, k-fold cross-validation generally provides an optimal or suboptimal evaluation results. It does not guarantee the minimum influence of group splitting of solutions but it does provide satisfactory answers. Thus, we conclude k-fold cross-validation comprehensively evaluates the performance.

In terms of fold number, we execute programs with different fold numbers to check if 5-fold is optimal. Before fitting data into process, a test set is split from dataset. Among k candidates, we uses the best model to test on test set to check if the reliability of this method. The following table shows the average results of 5 different random seeds. According to Table 4, we can find the 5-fold is optimal. 5-CV has both satisfactory mean and standard deviation of MCC. It is because it has a empirically ideal train-test ratio of 4 (i.e. 80% training data), while other fold number either has 'waste' a large amount of data on training which leads underfitting or allocate a small number of data for testing which makes results sensitive to data-splitting. Larger K number is more applicable than 5 folds if a larger dataset is provided, but 5-CV is optimal in this paper.

3.3 Effectiveness of Bimodal Distribution Removal

BDR is built based on statistics. It regards the occurrence of bimodal distribution as a signal that outliers are trying to mislead models. Hence, we start to prune them to improve model quality. To examine if BDR is

Table 4: MCC on fold number

K	mean	std
2	0.86	0.048
3	0.87	0.025
4	0.89	0.019
5	**0.89**	**0.017**
6	0.85	0.065
7	0.87	0.041
8	0.88	0.054
9	0.89	0.064
10	0.89	0.060

functional, we design two modes. The difference is whether deploy BDR, while other settings remain the same. After contrasting results, we conclude if BDR is efficacious and explain in details.

This subsection plots two figures which are 1^{st} and 51^{th} error distribution with BDR. , According to Figure 3, it illustrates that error distribution of data is bimodal distribution, which supports the idea in [18]. After trained for 50 epoch and once execution of BDR, it shows that the model better understands data as error of data decrease and data become less scattered. In this case, this indicates BDR indeed denoises daatset.

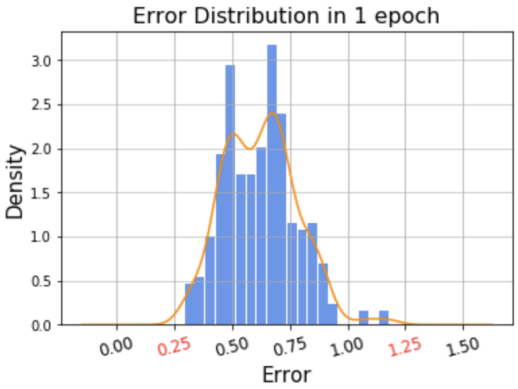

Fig. 3: 1^{st} Error Distribution with BDR

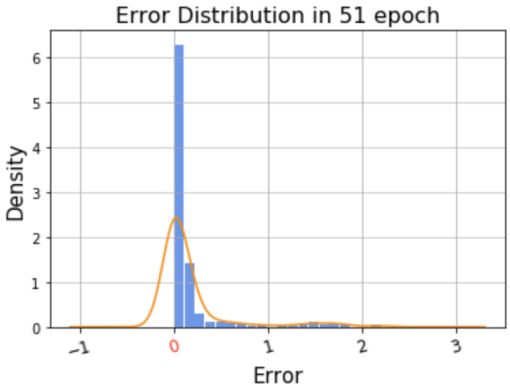

Fig. 4: 51^{th} Epoch Error Distribution without BDR

There exists solely one constant α in Equation 6, so it is a key value affecting performance. We then plot the MCC with respect to α. The results are shown in Figure 5. From figure, α ranges in $[0.0, 0.6]$ performs the same, and BDR becomes worse as α approaches to 0.7. The reason might be that BDR converges for $\alpha \leq 0.6$ in this epoch which means all possible outliers are detected, yet larger α keeps more data to maintains high MCC. However, we maintain $\alpha = 0.2$, because it is more generalised to remove more noise for each time of data splitting.

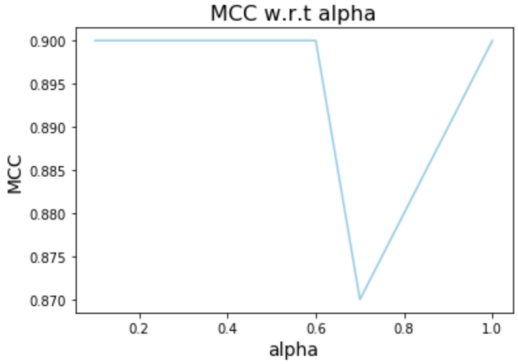

Fig. 5: Constant alpha and MCC

It is advised that BDR breaks down when the training set is 'clean' [18]. We therefore investigate if training examples are clean. The approach is to plot distribution of errors with first round of training. As shown in Figure 3, it is not an typical bimodal distribution but it is similar to normal distribution, which is preferable as it validates further inference on dataset [6]. Since training set is relatively clean, small in size and evenly containing either label, BDR is less applicable. As a variant version, instead of directly commencing BDR by $variance \leq 0.01$, we add an extra condition in advance to the start of algorithm. The condition is to lock the function if $variance < 0.1$ in first epoch. In the previous edition, $variance \leq 0.1$ signals that outlier appears. This validates inputs are clean if the signal occurs in first epoch.

In addition to calculating variance, we further try to replace the Equation 6 and Equation 7 by below formula. In Equation 12, x means data points to be removed. μ_{err} and σ_{err} represent the mean and standard deviation of errors accordingly. This idea is inpired by the features of normal distribution. In Equation 13, Q_1 is the first quantile and Q_3 is the third one. IQR is the interquartile range of all errors. We than remove intersection between candidates of two formula. We have a normal distribution of errors after the first epoch of training in Figure 5. In this case, we remove noise instances after the first epoch of training instead of executing through iterations.

$$(x > \mu_{err} + 3 * \sigma_{err}) \cup (x < \mu_{err} - 3 * \sigma_{err}) \tag{12}$$

$$(x < Q_1 - 1.5 \cdot IQR) \cup (x > Q_3 + 1.5 \cdot IQR) \tag{13}$$

Then, we compares the results of the original version and update version. The results are shown in Table 6. Updated BDR is applicable and yields accurate predictions, and it is moderately better than the original version. Outliers are detected and removed given this clean and small dataset. It might be less generalised compared to the original version, but it also yields satisfactory results.

Table 5: Test on update of BDR

	Original BDR		Variant BDR	
Round	MCC	Accuracy	MCC	Accuracy
1	0.86	93%	0.91	95%
2	0.89	94%	0.88	94%
3	0.88	94%	0.89	94%
4	0.90	95%	0.89	94%
5	0.87	93%	0.92	95%

3.4 Comparison to Logistic Regression (LR)

In previous sections, we examine the ability of techniques within the structure of Bi-LSTM. We do not know if it, as an estimator, is more effective than other models. In this section, we compare neural network with Logistic Regression and orthodox feedfoward neural network. We remain all dependent variables the same as what we derive from previous sections to maintain the accuracy of experiment.
Due to the limit of Logistic Regression and Orthodox neural network, data are process to fit into models. The dataset consists of 400 pieces of instances with 8 features. Samples are distributed evenly, which means there are equal number of sample labelled as genuine and posed. First two columns are labels with participants ID and video ID. The other features are real-value statistical data extracted from pupilliary response.

Table 6: Compare ANN, Logistic Regression and Bi-LSTM

	Logistic regression		Basic Neural network		Bi-LSTM	
Trial	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
1	0.083	54.11%	0.542	76.62%	0.913	95%
2	0.145	53.25%	0.710	85.29%	0.894	94%
3	0.096	54.79%	0.665	82.19%	0.907	95%
4	0.094	53.85%	0.637	80.77%	0.919	96%
5	0.054	50.96%	0.723	86.53%	0.881	93%

Looking at MCC, we find Logistic regression behaves as a gambler. The malfunction of logistic regression means the relationship between features and targets is non-linear. The orthodox neural network performs better than Logistic regression but still not accurate enough. It partially learns the data extracted from original dataset. Bi-LSTM performs significantly better than the other two, which means it fully understands data and is able to accurately classify sequence.

4 Conclusion and Future work

In summary, this paper demonstrates a bidirectional Long Short-Term Memory functioning as an anger-veracity estimator. This paper firstly designs a stable and accurate neural network by assembling an appropriate loss function and optimiser. We then derive a comprehensive evaluation method, by adopting MCC, which effectively avoids high-accuracy illusion. The next section further analyses the function of BDR statistically, and we propose an upgrade of this algorithm from the results. The outcome reflects that the new version successfully detects the cleanness of the dataset and yields accurate results. Lastly, this paper provides a comparison between the performance of logistic regression, basic feedfoward neural network and Bi-LSTM. The results imply that Bi-LSTM is more stable and accurate.

If we are to conduct a further research in depth, we should consider the effects of hyperparamters more carefully. Even though the exploration of the optimal configurations is tedious, tuning parameters is an indispensable step in designing a neural network. It would be preferable if more experiments can be conducted to optimise parameters. Furthermore, the design of experiments needs to be more rigorous. A pre-assigned random seed does not necessarily entail whether the influence of randomisation is present. We frequently evaluate performance by averaging scores in previous experiments, and relying on the mean scores of five results might be insufficient to draw a conclusion. Hence, a more rigorous experiment needs to be designed to cope with randomisation.

References

1. Alahmadi, A.A., Samson, R.S., Gasston, D., Pardini, M., Friston, K.J., D'Angelo, E., Toosy, A.T., Wheeler-Kingshott, C.A.: Complex motor task associated with non-linear bold responses in cerebro-cortical areas and cerebellum. Brain Structure and Function **221**(5), 2443–2458 (2016)
2. Bassett, D.S., Gazzaniga, M.S.: Understanding complexity in the human brain. Trends in cognitive sciences **15**(5), 200–209 (2011)
3. 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)
4. Chen, Y.Y., Lin, Y.H., Kung, C.C., Chung, M.H., Yen, I., et al.: Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. Sensors **19**(9), 2047 (2019)
5. Chicco, D.: Ten quick tips for machine learning in computational biology. BioData mining **10**(1), 35 (2017)
6. Croarkin, C., Tobias, P., Filliben, J., Hembree, B., Guthrie, W., et al.: Nist/sematech e-handbook of statistical methods. NIST/SEMATECH, July. Available online: http://www. itl. nist. gov/div898/handbook (2006)
7. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6**(02), 107–116 (1998)
8. 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)
9. Jayalakshmi, T., Santhakumaran, A.: Statistical normalization and back propagation for classification. International Journal of Computer Theory and Engineering **3**(1), 1793–8201 (2011)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kobayashi, H., Hara, F.: Recognition of six basic facial expression and their strength by neural network. In: [1992] Proceedings IEEE International Workshop on Robot and Human Communication. pp. 381–386. IEEE (1992)
12. Li, M., Zhang, T., Chen, Y., Smola, A.J.: Efficient mini-batch training for stochastic optimization. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 661–670 (2014)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
14. Matthews, B.W.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure **405**(2), 442–451 (1975)
15. Reddi, S.J., Kale, S., Kumar, S.: On the convergence of adam and beyond. arXiv preprint arXiv:1904.09237 (2019)
16. Rodriguez, J.D., Perez, A., Lozano, J.A.: Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE transactions on pattern analysis and machine intelligence **32**(3), 569–575 (2009)
17. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing **45**(11), 2673–2681 (1997)
18. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249–254. Springer (1993)
19. 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)
20. Srivastava, N.: Improving neural networks with dropout. University of Toronto **182**(566), 7 (2013)
21. Suresh, S., Sundararajan, N., Saratchandran, P.: Risk-sensitive loss functions for sparse multi-category classification problems. Information Sciences **178**(12), 2621–2638 (2008)
22. Tu, J.V.: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology **49**(11), 1225–1231 (1996)
23. Wang, B., Gong, N.Z.: Stealing hyperparameters in machine learning. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 36–52. IEEE (2018)
24. Wolberg, W.H., Street, W.N., Heisey, D.M., Mangasarian, O.L.: Computer-derived nuclear features distinguish malignant from benign breast cytology. Human Pathology **26**(7), 792–796 (1995)