Can bio-signal better discriminate liars: A Deception Recognizability Competition between Statistical Models with Human Belief

Chen Luoyu

School of Computing, The Australian National University
u6214908@anu.edu.au

Abstract. In an era with inundated fake and manipulated information, helping people to recognize deceptive information can be very beneficial. To help with deceptive recognition, this paper focuses on three tasks. Firstly, build neural network models, by using bio-signal data to detect the presence of manipulation. the second task is using Genetic algorithm as feature selection techniques to remove unnecessary features, to obtain a simpler and more effective model. However, due to complex architecture of neural network, the output is hardly interpretable. Therefore, the third task aims at extarcting rules to help understanding neural network output, for potential use in bio-signal diagnostics. A three-layer fully connected model can achieve prediction accuracy of 100%. A Multi-task learning model can achieve prediction accuracy of 100%. Gradient sensitivity rule extraction can achieve explanation accuracy 58% on Fully-connected model and 54% on Multi-task Learning model.

Keywords: Belief Prediction · Artificial Neural Network · Multi-task Learning · Genetic Algorithm · Rule Extraction· Gradient Sensitivity Analysis.

1 Introduction

1.1 Background

When facing potential deceptions, there will be two sources to distinguish liar and truth-teller: one is the cognitive recognition, it can either be an instant judgement by intuition or careful reasoning. Another is the continuous bio-signal data, which can capture subtle physiological changes while listening a speaker. When people's big-signal data are assessed, these data can generally better detect deceptions than human cognition by detecting human doubts [2]. To evaluate the research result, 'Subjective Belief' experiment was conducted, presenters were asked to deliver topics that some part are true while some part are deliberately notified as bogus content. Then listener's bio-signal data were recorded in the total duration of listening. Therefore to effectively use bio-signal data to build a liar detector, different Neural Network models are used, as they are good function approximators to describe the relation between bio-signal data and presence of

manipulation. And also, some of the bio-signal data may be useless or even disruptive in model prediction, Genetic algorithm is used to remove useless features to improve prediction performance. Finally, because Neural Network model is a multi-level nested function with vast number of weights in each nesting level, the learned knowledge are represented by weights, which are uninterruptible for researcher or experts in biomedical domain, who are trying to find exact conclusion on which kind of bio-signal data is playing a vital rule in deception detection. Thus, finding a good model output explain-er is urged in this case. This paper uses Gradient Sensitivity Analysis method to perform rule extraction because gradient is an indicator to reveal how much change of the output side will occur when the input side has a certain amount of change [1]. Overall, this paper aims at showing bio-signal data can better predict the presence of doubts to help people better discriminate liars and truth-tellers by using different Neural network models, and optimize model performance by using Genetic algorithm to omit useless features, finally use Gradient Sensitivity Analysis to extract rules for model output prediction, as an explanation assistance to the complex predictor, to help researchers discover which features are decisive in detecting the presence of doubts.

1.2 Bio-signal data

Research has shown that when listening lying speech, listeners can have lowered skin temperature on their fingers (ST), greater galvanic skin response (GSR), greater pupillary dilation (PD) and higher heart rate, and heart rate is alternatively measured by blood volume pulse (BVP) [2]. Each class of these bio-signal data came from continuous sensor record on each listener, and later the collected data were smoothed and sampled.[2]

The data consists of 23 person's bio-signal data, totally 368 records, with 16 records per person for different presenters and different topics. For each person, BVP has 34 records, GSR has 23 records, ST has 23 records, PD has 39 records, and the presence of manipulation was also recorded. Bio-signal data is in float data type and presence of manipulation is in boolean type, 0 or 1.

1.3 The proposed tasks

The first task is to train Neural Network models, using bio-signal data as input, and predict the presence of subjective belief manipulation on presenter's side. One model is a fully connected model with one hidden layer, another is a Multi-task Learning model with three layers in the shared layer, and 23 personal layers with one hidden layer.

The second task is to use Genetic algorithm as a feature space optimization, by encoding feature set as binary array and fitness value as model prediction accuracy, aim at learning a small set of feature to get a more compact model

while keeping good predictability.

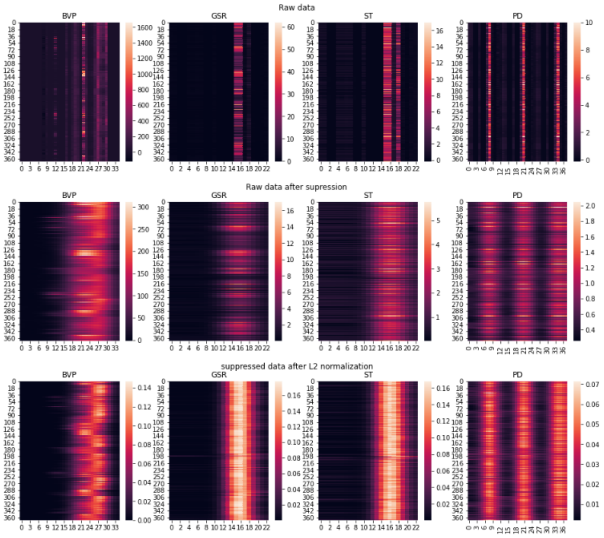
The third task is to perform rule extraction, use Gradient Sensitivity Analysis to make a model explanation model, given input, it can predict model output.

2 Method

2.1 Data Pre-processing

First observe the raw data for BVP, GSR, ST, PD (see 1st row Fig.1). At some moments of the entire listener's experiment duration, high peaks exist, and all the other moments are too dark because of the presence of theses peaks. Then the time series data will become almost binary, with peak data =1 and non-peak data =0, this may be destructive for model to extract useful information from learning the time series trajectory. So I suppress these peaks first. Here 1D

Fig 1. Data preprocessing heatmap



Gaussian filter with $\sigma = 3$ is selected to suppress these peaks while maintaining fluctuation magnitudes. Then filter each kind of bio-signal data, for each participants(see 2nd row Fig.1). And also, considering our participants are in different ages, different genders or having different physiological properties, so I re-scale the data for everyone's bio-signal fluctuation in the same range. Here I select *L2-norm* normalization to re-scale for each row data, do it four times as different type of bio-signal data have different range for everyone. What's more, *L2-norm* normalization can compensate these overly suppressed fluctuations after Gaussian smoothing at 1st step, to make signals have more manifest trajectory (see

3rd row Fig.1).

However, our neural network model cannot handle time series data directly, time series data is only a suitable input type for Recurrent Neural Network model such as LSTM [6]. So I transform time series data into statistical quantities, which can reveal the internal structure of the given time series data. There are four kinds of bio-signal data so I compute statistical quantities for each, and a summery of statistical quantities is shown below:

Table 1. Statistical quantities summary

Feature	Description
max	maximum signal over time
min	minimum signal over time
mean	average signal over time
std	standard deviation of signals over time
var	variance of signals over time
rms	root mean square of signals over time
diff1	first order difference of signals over time
diff2	second order difference of signals over time

Thus I finally obtain data has size 368 * 32. Apart from that, after data is processed and transformed, to prepare 5-fold cross-validation, I shuffle and split for each participant's data in ratio of 0.6:0.2:0.2 for train, validation, test respectively , shuffle is to avoid positive, negative samples imbalance in training, validation, test data. And split data for each participant is to make sure in Multi-task Learning model can learn task for each person in a balanced way.

2.2 3-layer Fully-connected model

Multiple settings of 3-layer fully connected structure are attempted, finally the best model I obtained is a 32-5-2 structure model with Sigmoid activation layers behind the input layer and the hidden layer, since the network is shallow, vanishing gradient issue is not a problem here. The optimizer chosen is Adam [5] and loss function is cross-entropy loss as a smooth of prediction value. To avoid over-fit, epoch is set to be 20 and learning rate is set to be 0.1.

2.3 Multi-task Learning model

The Multi-task Learning model consists of a shared layer and 23 task layers (towers), after attempting multiple settings, the structure of 32-8 for shared layer and 4-2 for task layer, Sigmoid activation layers are behind the input layer and first layer of task layer. Optimizer and loss function are all the same with the 3-layer Fully-connected model. Epoch is set as 50 and learning rate is 0.1, longer

training time is required as it is one layer deeper and more complex structure than the Fully-connected model.

2.4 Genetic Algorithm

After making Neural Network model works finely, Genetic Algorithm is applied for feature selection, in order to obtain a more compact and descent prediction accuracy model. Both fully connected model and Multi-task Learning model are optimized. I select fitness function as the $\frac{accuracy}{size(input)}$ to encourage good accuracy and discourage large size of input to achieve descent predictability with compactness. First I generate a large population with pop size = 100, set each chromosome in the form of a binary array with length =32, initialize each entry randomly with equal probability between 0, 1. In each iteration, set the mutation rate as 0.1 and cross rate as 0.8. Then after 10 generations of searching, I obtain the best model.

2.5 Rule Extraction via Gradient Sensitivity Analysis

After achieving a model with much less input (generally less than 10 features), the final step is to make model explainable. The extracted rules are in the form of

$$\text{if } X_i < T_i, \text{ then predict its class as: } C_i.$$

Here X_i is the ith attribute of pattern X, T_i is the decision boundary, C_i is the rule prediction result. However, because we only have two classes, so a single rule will be enough. However extracted rules may be predicting a pattern into different classes, so instead of believing any one of the rule prediction, I use each rule prediction as a vote, if positive class wins equal to or more than half of votes, the ensembled rule prediction predict X as positive class.

The method to determine the decision boundary and predicting class comes from analysis of gradients. Use the gradient for output w.r.t. inputs from the final epoch of neural network model training. The boundary values is are obtained from the input, so we have gradient matrix and input matrix in the form shown below: (S is the last feature from the GA obtained compact model).

Gradient matrix G

$$\mathrm{G}=\begin{bmatrix} G_{1,1} & G_{1,2} & \dots & G_{1,S} \\ G_{2,1} & G_{2,2} & \dots & G_{2,S} \\ \vdots & \vdots & \ddots & \vdots \\ G_{368,1} & G_{368,2} & \dots & G_{368,S} \end{bmatrix}$$

Input matrix A

$$\mathrm{A}=\begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,S} \\ A_{2,1} & A_{2,2} & \dots & A_{2,S} \\ \vdots & \vdots & \ddots & \vdots \\ A_{368,1} & A_{368,2} & \dots & A_{368,S} \end{bmatrix}$$

Then T_i should locate at input matrix column i row j, and j maximize the gradient in the column i, which mathematically should be:

$$j = \underset{1 \leq j \leq S}{\operatorname{argmax}} G_{ij}$$

and

$$T_i = A_{ij}$$

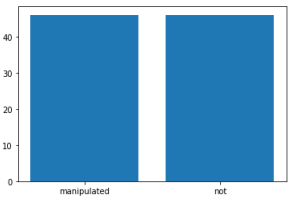
To get the prediction class label C_i, it should come from lots of sampling at the left side of decision boundary T_i, and the sampling will be counting the frequency of positive pattern label and negative pattern label (the label is coming from model output). If positive pattern has greater frequency, then the predicted class label should be 1, else 0. Also, considering model output maybe imbalanced, so instead of comparing the frequency of positive output and negative output, we compare the relative frequency, so if the at the left side of the boundary, the frequency of positive output is $F_{left}(1)$, and the population frequency (both left and right) is $F_{all}(1)$. If $F_{left}(1) > F_{all}(1)$, we will believe positive pattern is relatively more frequent at left side over the entire domain, and the rule prediction label will be 1 in this case.

3 Results and Discussion

3.1 Neural Network performance

The validity of model results are highly sensitive to pattern class balanceness, to ensure the results provided below are convincing, Fig 2. is the positive/negative pattern distribution in test set: The test data is unseen at training phase and

Fig 2. Positive Negative pattern distribution



validation phase as it had been split out at the data pre-processing stage. I also performed identical checking to ensure no duplicates intersects training, validation and test set. As we can see from Fig.2, the distribution of positive negative pattern is perfectly half-half, with totally 92 testing patterns, 46 are positive patterns and 46 are negative patterns. So the results displayed below should be trustable.

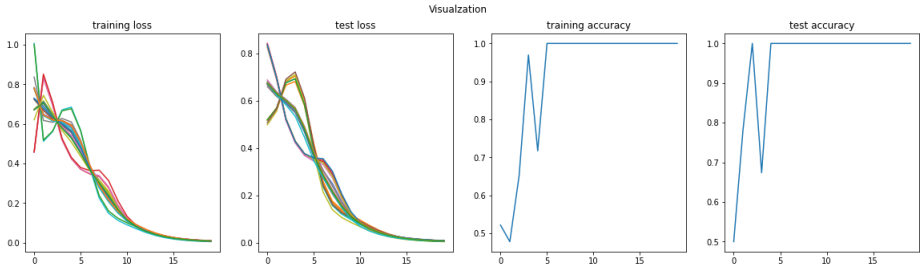
Table 2. Neural Network model evaluation

Feature	accuracy	precision	recall	f1 score
Fully-connected model	1.0	1.0	1.0	1.0
Multi-task Learning model	1.0	1.0	1.0	1.0
GA Fully-connected model	1.0	1.0	1.0	1.0
GA Multi-task Learning model	1.0	1.0	1.0	1.0

3.2 Neural Network performance discussion

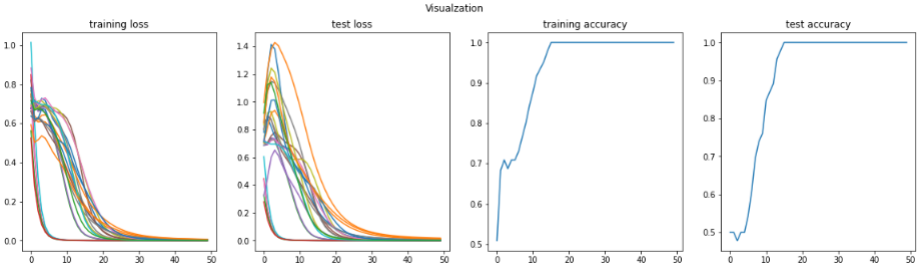
Though both the two models achieve equally great performance in every score, but w.r.t. convergence speed, Fully-connected model have the steepest convergence speed, both models performance visualization are displayed in Fig.4 and Fig.5.

Fig.3. Fully-connected-model performance



The first picture in Fig.3 shows the training loss w.r.t. each task and the second one shows the validation loss w.r.t. each task, the third one shows the training accuracy and the fourth one shows the test accuracy. Fig.5 is the same as Fig.4.

Fig 4. Multi-task Learning model performance



Thus w.r.t. the accuracy, precision, recall, f1 score, both model are perfect. Accuracy examine the prediction correctness, precision examine the predictability for positive patterns, recall examine the predictability for negative patterns, f1 score combine both precision and recall, examine the robustness of model. Hence, both two models can be believed to solve the presence of manipulation prediction problem perfectly in a quite small volume of dataset. Even after GA, the selected features generally have number less than 10. thus we can believe for manipulation detection, statistical model can highly surpass human performance with accuarcy 0.54 [2].

Compare with the 'Subjective belief' experiment result, they obtained accuracy of 0.63, Precision 0.64 Recall 0.64 F1 score 0.63 in a Fully-connected model and accuracy of 0.68, Precision 0.74 Recall 0.72 F1 score 0.72 in Multi-task Learning model. The Fully connected model architecture is 119-512-2 with Sigmoid activation layer behind input layer and hidden layer, the Multi-task Learning model architecture has shared layer as 119-350-350 with Sigmoid activation layer behind input layer and hidden layer, and 50-2 in each tower with a Sigmoid layer behind tower input layer, and totally 23 towers [2]. The reason mine model is achieving unexpected much better results perhaps comes from or this experiment used a much larger dataset with much more noise.

When it comes to convergence, Fully-connected model is slightly quicker than Multi-task learning model. There are two reasons. First the Fully-connected model is one layer shallower than Multi-task Learning model, and it has much less parameters, as Multi-task Learning model contain 23 towers behind the shared layer. As we know,in practice shallower model is easier to train because less parameters need to be tuned. In contrast, in deeper model, *dof* increases drastically, should take longer to tune to fit training data. Second reason comes from some good aspects of Multi-task learning model, to illustrate this, two concepts need to be discussed first: one is *Attention focusing* [3], as in shared layer, the model is trained to fit loss for each task, so shared layer is trying to shift attention to generic features rather than a specific task, alternatively, it can be seen as an *Implicit regularization* [3], to avoid the learned model overfit on some small portion the tasks just in order to decrease the one loss for all tasks by finding the steepest direction, and this is the case for the Fully-connected model.

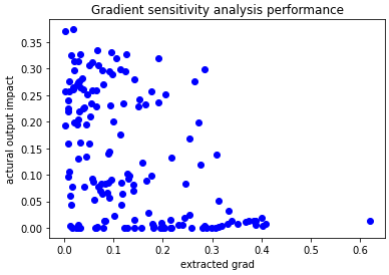
From Fig.4 and Fig.5, because the shared layer inclines to learn general feature representation from all tasks, therefore fitting specific tasks becomes slower. As we can see in epoch step 0-10, there is a much slighter oscillation for training loss curve on Multi-task learning model than Fully-connected model, indicating none of these 23 tasks are heavily 'penalized' because of learning other tasks. In later steps, the training loss variance for Multi-task Learning model between most tasks shrinks steadily, while Fully-connected model 'spikes' twice after 10 epochs, this indicates again the shared layer parameters are only fine-tuned be-

cause in the first 10 epochs, generic feature representation has been learned. However, Fully-connected model's parameters vibrate heavily for a relatively much longer time, because although reducing the one for all loss follows the Adam defined steepest descent direction, but results in some tasks are learned quickly while some are penalised heavily at the same time, so generic feature representation is not learned as effectively as Multi-task Learning model.

3.3 Rule Extraction results and discussion

For Multi-task Learning model the explanation accuracy is 54%, for Fully-connected model the explanation accuracy is 58%, both explanation accuracy are almost at chance level. To evaluate the validity of gradient based method, I randomly pick one sample from dataset, then test the actual impact for each attribute to compare with its maximum gradients, which convey attributes significance in Gradient sensitivity analysis. That is, if to test the impact for attribute i, then fix other attributes of this sample and increment only attribute i with step length 0.05, then feed all incremented data into trained model, a series of model output from these recreated data can be obtained. I did for all the attributes. Then calculate the average of first order difference for the output series, and compare with extracted gradient. If a nice positive correlation can be found between the extracted gradient for attribute i and series' first order difference average, then this approach should be sensible. However, after testing, as shown in Fig.5,

Fig.5 extracted attribute grads vs attribute actual influence



there is poorly any positive correlation between extracted gradients and the actual impact in Fully-connected model. For Multi-task Learning model, it has 23 outputs and it can be seen as a combination of 23 Fully-connected model. For a single Fully-connected model we can not obtain a decent positive correlation, it is expected that Multi-task Learning model as a mixture of Fully-connected models will obtain an even worse result.

One reason that can possibly cause bad result is model convergence(most extracted grads are less than 0.1 in Fig.5), because our models are obtained

after all training loss decrease almost to 0 (together with validation loss decrease to 0), gradients for any attributes becomes very small, thus changing a single attribute maybe too weak to cause prediction changes, then vote from a single rule becomes indecisive, then maybe only when a collection of attributes have accumulated a large enough change, the corresponding rule can be more decisive.

4 Conclusion and Future Work

Overall, this paper use bio-signal data and build a Fully-connected model and a Multi-task Learning model, both achieving perfect result, revealing bio-signal data can be used to detect liars effectively, undoubtedly superior than human recognizability. However, gradient analysis based rule extraction is unsuccessful, the reasons both come from the convergence issue, and the nature of multiple outputs in Multi-task learning model, so gradient based method for rule extraction is not quite suitable.

In future, Multitask-learning + LSTM model will be attempted, the input side is still statistical quantities, but after shared layer, 4*23 LSTM streams will be appended, so each tower contains 4 streams of LSTM, to process BVP, GSR, ST, PD time series data respectively. For me this is a novel idea, because it combines the power of Multi-task learning model in generic feature learning and the power of LSTM for learning time series data. For rule extraction non-gradient based method will be attempted, like using decision tree and feed model outputs as decision tree input.

References

1. Gedeon, T. D., Turner, S. (1993, October). Explaining student grades predicted by a neural network. In Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on (Vol. 1, pp. 609-612). IEEE.
2. Zhu X., Qin Z., Gedeon T., Jones R., Hossain M.Z., Caldwell S. (2018) Detecting the Doubt Effect and Subjective Beliefs Using Neural Networks and Observers' Pupillary Responses. In: Cheng L., Leung A., Ozawa S. (eds) Neural Information Processing. ICONIP 2018. Lecture Notes in Computer Science, vol 11304. Springer, Cham. https://doi.org/10.1007/978-3-030- 04212-7_54.
3. 3.Ruder, Sebastian. "An Overview of Multi-Task Learning in Deep Neural Networks." ArXiv:1706.05098 [Cs, Stat], June 2017. arXiv.org, http://arxiv.org/abs/1706.05098.
4. Engelbrecht A., Viktor H. (1999) Rule improvement through decision boundary detection using sensitivity analysis. In: Mira J., Sánchez-Andrés J.V. (eds) Engineering Applications of Bio-Inspired Artificial Neural Networks. IWANN 1999. Lecture Notes in Computer Science, vol 1607. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0100474.
5. Kingma, Diederik P. and Jimmy Ba. "Adam: A Method for Stochastic Optimization." CoRR abs/1412.6980 (2015).
6. Hochreiter, S. and Schmidhuber, J. "Long short-term memory". Neural computation, 9(8), 1735–1780 (1997).