Deception Detection for MU3D using Random Forest-based Ensemble Learning

Kun Bu kunbu@usf.edu
Department of Mathematics & Statistics
University of South Florida
Tampa, FL 33620-5700, USA

Kandethody Ramachandran ram@usf.edu
Department of Mathematics & Statistics
University of South Florida
Tampa, FL 33620-5700, USA

Abstract

The purpose of this work is to detect people lying using different and newly developed machine learning algorithms to conclude a better prediction model through comparison. Random Forest (RF) did an efficient work while dealing with both classification and regression problems; In this paper, we proposed a newly developed Random Forest-based ensemble learning, which is the combination of RF with SVM, GLM, KNNs, and GLM to improve the model performance. The data set that we used to fit into the machine learning models is Miami University Deception Detection Database (MU3D). MU3D is a free resource containing 320 videos of Black and White targets, female and male, telling truths and lies. We fit the MU3D video level data set into Random Forest-based ensemble learning models, which includes RF+SVM.Linear, RF+SVM.Poly, RF+GLM, RF+KNNs, RF+GBM(Stochastic Gradient Boosting) and RF+WSRF(Weighted Subspace Random Forest). As a comprehensive comparison of the model performance, we conclude our new combination of algorithms performs better than the traditional machine learning models. Our contribution in this work provides a robust prediction method which improves the ~~predicted~~ performance while avoiding model overfitting.

Keywords: Ensemble learning, deception detection, exploratory data analysis (EDA), weighted subspace random forest, generalized linear model, stochastic gradient boosting.

1. Introduction

Traditional lie detection machine is a polygraph, which can provide people with an averaging accuracy between 58% to 90%. With 90% accuracy, it seems to do a very good job on detecting lying, however, with 58% accuracy, we can hardly have much confidence to say a person is lying. In other words, the polygraph test is easy to pass for those well-trained people (ie. company spies or country spies). Even ordinary people who search for the word “polygraph” online, the next searching suggestion would be “How to Pass a Polygraph Test?” Since the polygraph operating principle is to detect lies by looking for signs of an examinee’s physiological changes. Once the examinee lies, it puts a blip on the polygraph machine that serves as a signature of that examinee’s lies. Besides, polygraph test is a time based test that only captures the examinee’s body reaction in each specific question, which means the examinees themselves know that they’re being tested whether they are lying. Therefore, polygraphs are not useful for

those underground and secret cases. Therefore, artificial intelligence (AI) approaches come to scientist's minds. Why don't we just detect lying by applying machine learning algorithms to see if the accuracy of deception detection would be improved.

The Miami University Deception Detection Database (MU3D) is a free resource containing 320 videos of Black and White targets, female and male, telling truths and lies. Eighty (20 Black female, 20 Black male, 20 White female, and 20 White male) targets were recorded speaking honestly and dishonestly about their social relationships. Each target generated four different videos (i.e., positive truth, negative truth, positive lie, negative lie), yielding 320 videos fully crossing target race, target gender, statement valence, and statement veracity. The stimuli and an information codebook can be accessed free of charge for academic research purposes from http://hdl.handle.net/2374.MIA/6067. In the previous studies of MU3D, scholars conducted research using standardized stimuli that can aid in building comprehensive theories of interpersonal sensitivity, enhance replication among labs, facilitate the use of signal detection analyses, and promote consideration of race, gender, and their interactive effects in deception detection research.

Ensemble learning, sometimes referred to as a multi-classifier system, builds and combines multiple classifiers to complete the learning task. Generally speaking, there are two choices for getting multiple classifiers. The first is supposed that all individual classifiers are of the same type, or homogenous. For example, both decision tree individual classifiers, or both neural network individual classifiers (i.e. Bagging and boosting, for example, Random Forest). The second is supposed that all individual classifiers are not homogeneous, or heterogeneous. For example, in this paper, we have a classification problem of deception detection, we use support vector machine (SVM) individual learner, logistic regression (LR) individual learner and k-Nearest Neighbors (KNNs) individual lea[illegible] to learn the training set, and then determine the final strong classifier by some combination strategy. This integration is called Stacking. In the experimental section, we applied both Bagging, boosting and stacking, and selected a better ensemble model to predict people lying.

This paper is organized as follows. In section 2, we will talk about the related work in deception detection; section 3 provides the exploratory data analysis (EDA) of a dataset to solve deception detection. Section 4 explains our selected methods for deception detection based on the statistical analysis according to section 2. Section 5 introduced the experimental setup, and finally in section 6, we discuss our results and drawbacks ~~respectively~~.

2. Related Work

As a widely studied phenomenon in many disciplines, deception, in psychology, is defined as an act that is intended to foster in another person a belief or understanding which the deceiver considers false (Krauss et al. (1976)). Previous work on deception detection has focused on a combination of different factors including verbal and non-verbal aspects. Text/audio only approaches alone using RNN or LSTM architecture were able to achieve only a moderate amount of accuracy 76% - 84% (Venkatesh et al. (2019)) . Micro-expression only approaches achieved higher accuracy of 77% - 88% (Venkatesh et al. (2019)). Abouelenien et al., Akoglu et al., and Ott et al. have extensively studied the challenging task for human deception detection. In their previous work, the accuracy of deception detection that is predicted

by machines is approximately as high as 90%, which is a great achievement compared with the accuracy that is predicted by humans (Ott et al. (2011)). The task introduced by Ott et al. provides an ideal sandbox to understand human predictions with assistance from machine learning models (Lai and Tan (2019)). Lai et al. summarize related work in two areas to put the deception detection work in context: interpretable machine learning and deception and misinformation (Lai and Tan (2019)). They claim that Machine learning models remain as black boxes despite wide adoption. Blindly following machine predictions may lead to dire repercussions, especially in scenarios such as medical diagnosis and justice systems (Caruana et al. (2015)). Therefore, improving their accuracy and interpretability has become the new trend in ML studies (Kim et al.(2016)). To improve the ML performance, Chen et al. investigates the performance of combining support vector machines (SVM) and various feature selection strategies, for example, they applied both filter-type approaches and wrapper-type methods for feature selection (Chen and Lin (2006)). Xiong et al. introduce a new large margin classifier, named SVM/LDA, which is an extension of support vector machine by incorporating some global information about the data, and their new formulation is expected to perform better or similar to standard SVM classifiers, and this assertion is empirically verified using several artificial and real-life benchmark data sets (Xiong and Cherkassky (2005)).

3. Exploratory Data Analysis

3.1 Data Description

As mentioned in the introduction part, the MU3D is collected by recording 80 targets speaking honestly and dishonestly about their social relationships. The dataset was divided into two parts: video level and target level. In the video level dataset, information such as the valence, indicating whether the statement in the video is negative or positive; VidLength ms and VidLength sec indicates the length of video in millisecond and seconds, respectively. There are a total number of 12 variables with 1 label variable called Veracity in the video level dataset, a short variable description is shown in Table 1.

Table 1: MU3D Video level database variables description (Hugenberg et al. (2017)).

Video-Level Variables	Description
VideoID	ID associated with the video.
Veracity	Indicates whether the statement in the video is a truth or a lie: value of 0 indicates a lie, value of 1 indicates a truth.
Valence	Indicates whether the statement in the video is negative or positive: value of 0 a indicates negative statement, value of 1 indicates a positive statement.
Sex	Indicates target's sex: value of 0 indicates a female target, value of 1 indicates a male target.
Race	Indicates target's race: value of 0 indicates a Black target, value of 1 indicates a White target.

VidLength_ms	Indicates length of the video in milliseconds.
VidLength_sec	Indicates length of the video in seconds.
WordCount	Indicates the number of words contained in the full transcription of the video.
Accuracy	Indicates average accuracy (i.e., proportion correct) across raters who viewed the video.
TruthProp	Indicates average truth proportion (i.e., proportion of truth responses) across raters who viewed the video.
Attractive	Indicates average attractiveness ratings (measured on a scale ranging from 1 ”Not at all” to 7 ”Extremely”) across raters who viewed the video.
Trustworthy	Indicates average trustworthiness ratings (measured on a scale ranging from 1 ”Not at all” to 7 ”Extremely”) across raters who viewed the video.
Anxious	Indicates average anxiousness ratings (measured on a scale ranging from 1 ”Not at all” to 7 ”Extremely”) across raters who viewed the video.
Transcription	Full transcription of the video.

3.2 Statistical Data Analysis

Due to the data properties, normalizing the data so that they are of the same order of magnitude is much better for machine learning. Except for VideoID and the last variable Transcription, a normalized box-plot was created for all other variables by Veracity. As shown in Figure 1, only the variables Accuracy, TruthProp and Trustworthy have differences in Veracity, which is hard for us to start choosing features to train a classification prediction model.

A correlation scatter plot was shown in Figure 2, it indicates VidLength ms and VidLength sec, variable TruthProp and Trustworthy, variable Accuracy and TruthProp are highly correlated, however, the effect here is similar to that of multicollinearity in linear regression. Our learned model may not be particularly stable against small variations in the training set, because different weight vectors will have similar outputs. The training set predictions, though, will be fairly stable, and so will test predictions if they come from the same distribution. Based on the variable's linear correlation relationship in Figure 2, we can reduce the features dimension from 11 to 9, where the VidLength_ms and TruthProp were removed because they are linearly correlated with VidLength_sec and Accuracy, respectively.

According to the difficulties that arise from statistical data analysis, we applied both normalization and Principal Component Analysis (PCA) in the beginning of section 5 as an experiment setup. See details in section 5.

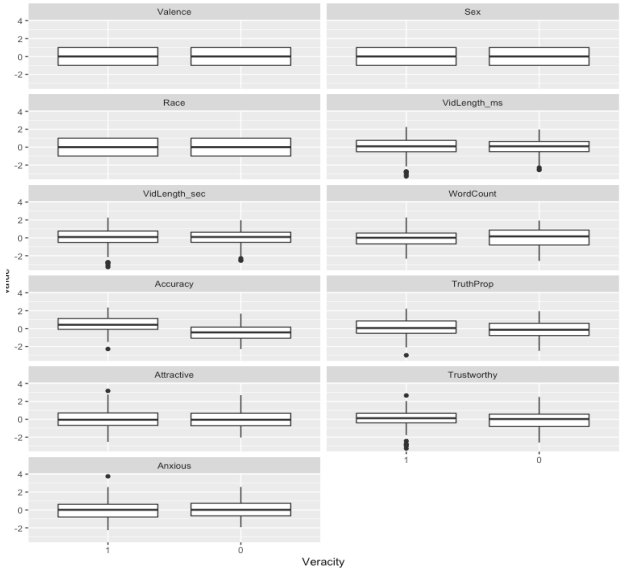

Fig. 1: Boxplot for video level dataset variables.

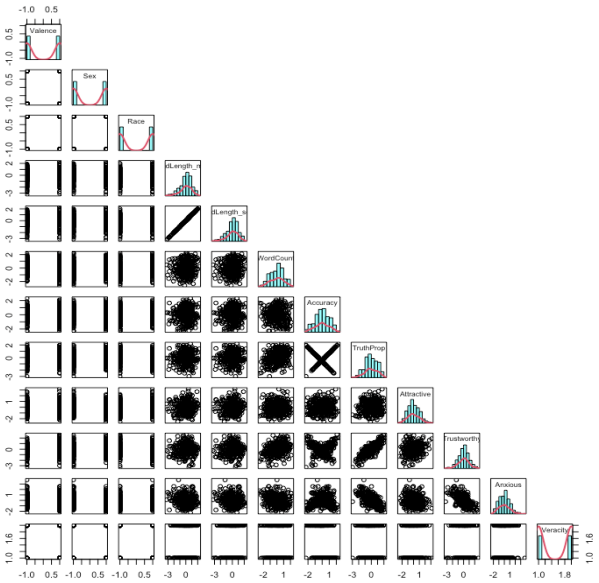

Fig. 2: Correlation scatter plot for video level dataset.

4. Ensemble Learning for Deception Detection

4.1 Algorithms Selecting Procedures

The next step is to fit MU3D into a machine learning model and see how the computer performs on detecting lies. We first trained three different models based on the data properties, including Support vector machine (SVM), Binary Logistic Regression (BLR) and Random forest (RF) to predict the deception. The basic idea of these three selected algorithms are based on the ~~below~~ algorithm flowchart. The purpose of this algorithm flowchart is to create a tool that helps not only select the possible modeling techniques but also understand deeper about the problem itself. As shown in Figure 3, by answering the following questions from the flowchart, we initially decided the main three modeling techniques to be applied in this prediction. In the next subsection 4.2, we explained the reason why these three machine learning were chosen based on the data properties and the model assumptions.

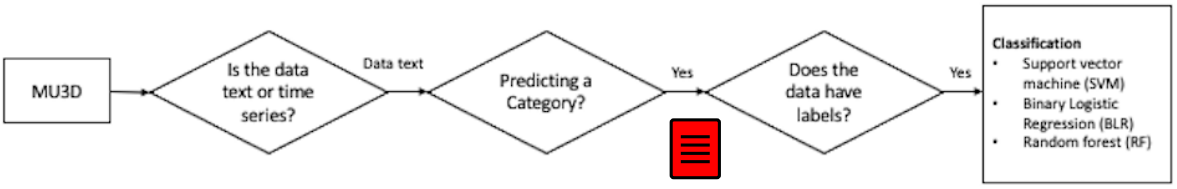

Fig. 3: A Flowchart of the Basic Ideas in Deception Detection.

4.2 Preliminaries Machine Learning Algorithms

4.2.1 Support Vector Machines (SVM)

Support Vector Machines are based on a decision plane concept that defines decision boundaries, and SVMs have been shown to perform well in a variety of settings, and are often considered one of the best ”out of the box” classifiers according to James, G. et al.(2013) James et al. (2013). A decision plane is a separation plane between a set of objects with different types of membership.

SVM is a supervised learning method used to perform binary classification on data. According to the statistical data analysis section, the data properties show that we have exactly two classes : Lies or Truth. Besides, SVM can deal with real valued features, which means there are no categorical variables in the data, such as our dataset above, all of the features excepted the Transcription are numerical numbers, which are much fittable by using SVM. What's more, the SVM can perform well on a large number of features, for example, it works with ten, hundreds and thousands of features. In our dataset, we have more than 10 features which motivates us to choose SVM. Another reason I would like to mention here is that SVM has simple decision boundaries, indicating that there are no issues with over fitting. The SVM can be defined as linear classifiers under the following two assumptions: 1) The distance from the SVM's classification boundary to the nearest data point should be as large as possible; the distance formulas include ~~such as~~ Euclidean distance, Manhattan distance, Chebyshev distance and Minkowski distance. Where the Euclidean distance and Manhattan distance are special forms of the Minkowski distance. The formula are:

$$\text{Euclidean distance: } d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$\text{Manhattan distance: } d(x,y) = \sum_{i=1}^{n}(|x_i - y_i|)$$

$$\text{Chebyshev distance: } d(x,y) = max_i(|x_i - y_i|)$$

$$\text{Minkowski distance: } d(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$$

Where x and y are n-dimensional arrays, i.e.

$$x = (x_1, x_2, \ldots, x_n)$$

$$y = (y_1, y_2, \ldots, y_n)$$

2) The support vectors are the most useful data points because they are the ones most likely to be incorrectly classified. This means that the primary goal of training SVMs is to find support vectors in the dataset that both separate the data and find the maximum margin between classes.

4.2.2 Binary Logistic Regression (BLR)

Binary logistic regression (LR) is a regression model where the target variable is binary, that is, it can take only two values, 0 or 1. It is the most utilized regression model in readmission prediction, given that the output is modeled as readmitted (1) or not readmitted (0). BLR is a statistical tool that classifies local node behavior to either malicious or benign. BLR has two stages: training and evaluation. At the training stage it uses node behavior from both benign and malicious node activity and derives a detection module. At the evaluation phase, data that was not used in the training stage, is used to evaluate the detection model.

Logistic regression is also called logit regression, and therefore we usually build a Logistic regression model by starting to build the model by combining the generalized linear model with logit function, or from the point of a random variable following the logistic distribution. Binary Logistic Regression h the following assumptions: 1) adequate sample size, 2) absence of multicollinearity and 3) no outliers.

The mathematics expression of the logistic regression is

$$g(y) = ln(\frac{y}{1-y}) = \hat{w}^T \cdot \hat{x}$$

Where $g(y) = ln(\frac{y}{1-y})$ is called logit function and it is the link function for the generalized linear model. This logistic regression model can be expressed as

$$y = \frac{1}{1+e^{-x}}$$

Where e is the natural logarithm, and the above function is called sigmoid function.

4.2.3 Random forest (RF)

According to ~~Fernandez Delgado et al. (2014)~~Fernández-Delgado et al. (2014) "The classifier most likely to be the best are the random forest versions, the best of which (implemented in R and accessed caret), achieves 94.1% of the maximum accuracy overcoming 90% in the 84.3% of the data sets." This quote clearly pointed out the power of RF in the classification field. The basic idea of RF is to produce numerous trees and combine the results. The random forest technique does this by app g two different tricks in model developmen he first is the use of bootstrap aggregation or short agging. In the bagging process, a single tree is built on a random sample of the dataset, which accounts for about two-thirds of the total observations (note that the remaining third is called out-of-bag (oob)) . Repeat this dozens or hundreds of times, and then calculate the average of the results. The growth and pruning of each tree is not based on any error measure, which means that the variance of ea ee is high. However, by averaging the results, you can reduce the variance without increasing the bias.

BAGGING

Training phase

1. Initialize the parameters
   - $\mathcal{D} = \emptyset$, the ensemble.
   - L, the number of classifiers to train.
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from $\mathbf{Z}$.
   - Build a classifier D_k using S_k as the training set.
   - Add the classifier to the current ensemble, $\mathcal{D} = \mathcal{D} \cup D_k$.
3. Return $\mathcal{D}$.

Classification phase

4. Run $D_1, \ldots, D_L$ on the input $\mathbf{x}$.
5. The class with the maximum number of votes is chosen as the label for $\mathbf{x}$.

Fig. 4: Bagging Algorithm.

The next thing that random forest brings to the table is that concurrently with the random sample of the data, that is, bagging, it also takes a random sample of the input features at each split. We will use the default random number of the predictors that are sampled, which, for classification problems, is square root of the total predictors. The advantage of RF by doing this random sample of the features at each split and incorporating it into the methodology, you can mitigate the effect of a highly correlated predictor becoming the main driver in all of your bootstrapped trees, preventing you from reducing the variance that you hoped to achieve with bagging. The subsequent averaging of the trees that are less correlated to each other is more generalizable and robust to outliers than if you only performed bagging.

4.2.4 Random Forest-ed Ensemble Learning

Thomas G. Dietterich pointed out that the effectiveness of ensemble learning can be attributed to the reasons from both statistical and computational. For statistical reasons: For normal study tasks, often want to search the assumption of space is very large, but can be used for training classifier training focused instance number is not enough to accurately learn goal hypothesis, this time the results of the study may be a series of meet the training set of assumptions, and the learning algorithm can choose one of these assumptions as a classifier to learning to the output. However, through the overfitti roblem of machine learning, we can see that the hypothesis that can meet the training set may not have the same good performance in practical application, so the learning algorithm faces certain risks when choosing which hypothesis to output. This risk can be mitigated by the integration of multiple hypotheses (which can be understood as the integration of the individual hypotheses and the tar ypothesis to some extent offset the error). For computational reasons: It has been proved that in Ann learning and decision tree learning, learning the best Ann or decision tree is an NP-hard problem, and other classifier models also face similar computational complexity problems. This leaves us with some heuristic methods to reduce the comple of finding the target hypothesis, but the result is that the hypothesis found is not necessarily optimal. By integrating multiple assumptions, the final result is closer to the actual objective function value. Therefore, in this paper e applied the Random Forest-based Ensemble Learning to improve the prediction performance.

We fit the MU3D video level data set into Random rest-based ensemble learning models, which includes RF+SVM.Linear(SVM with linear kernel), RF+SVM.Poly(SVM with polynomial kernel), RF+GLM(Generalized Linear Model), RF+KNNs(k-Nearest Neighbors), RF+GBM(Stochastic Gradient Boosting) and RF+WSRF(Weighted Subspace Random Forest). Section 5 explains a comprehensive comparison of the model performance based on the experimental, and then we co ude our new combination of algorithms performs better than the traditional machine learning models.

5. Experimental

5.1 Experiments Setup

As mentioned in the Data description section, the MU3D contains 320 data cells that tell truths and lies. 80 targets were recorded speaking honestly and dishonestly about their social relationships. Each target generated 4 different levels (i.e., positive truth, negative truth, positive lie, negative lie), yielding 320 videos data cells. Before building our models with SVM and BLR, we choose a common split ratio as 80:20 for training/validation and testing. As for the RF, we do not split our dataset because Random

Forest does not require a split sampling method to assess accuracy of the model. It performs internal validation as 2-3rd of available training data is used to grow each tree and the remaining one-third portion of training data is always used to calculate out-of-bag error to assess model performance.

5.2 Normalization

Normalization is a data preparation technique that is frequently used in machine learning. The process of transforming the columns in a dataset to the same scale is referred to as normalization. Every dataset does not need to be normalized for machine learning. It is only required when the ranges of characteristics are different. Standardization Scaling, that is, centering variables at zero and standardizing the variance at one. Subtracting the mean of each observation and then dividing by the standard deviation is the procedure as

$$X^{'} = \frac{X-\mu}{\sigma}$$

For example, the ranges of variables VidLength ms, VidLength sec and Attractive, Trust-worthy, Anxious are varied. Therefore, normalization was applied for the MU3D. Note that Normalization was done after splitting the data between training and test set, using only the data from the training set. This is because when normalizing the test set, one should apply the normalization parameters previously obtained from the training set as is. Recalculate them on the test set would be inconsistent with the model and this would produce wrong predictions, and the test set plays the role of fresh unseen data, so it's not supposed to be accessible at the training stage. Using any information coming from the test set before or during training is a potential bias in the evaluation of the performance.

5.3 Dimensionality Reduction

Since there are more than 10 dimensions in this MU3D, as shown in Table 1 and Figure 2, multicollinearity needs to be addressed and removed. We applied the Principal Component Analysis (PCA), which takes advantage of multicollinearity and combines the highly correlated variables into a set of uncorrelated variables. Therefore, PCA can effectively eliminate multicollinearity between features. Here in Figure 5 shows the scree-plot and the cumulative variance plot from PCA, which indicated the cutoff number of PCs is 5.

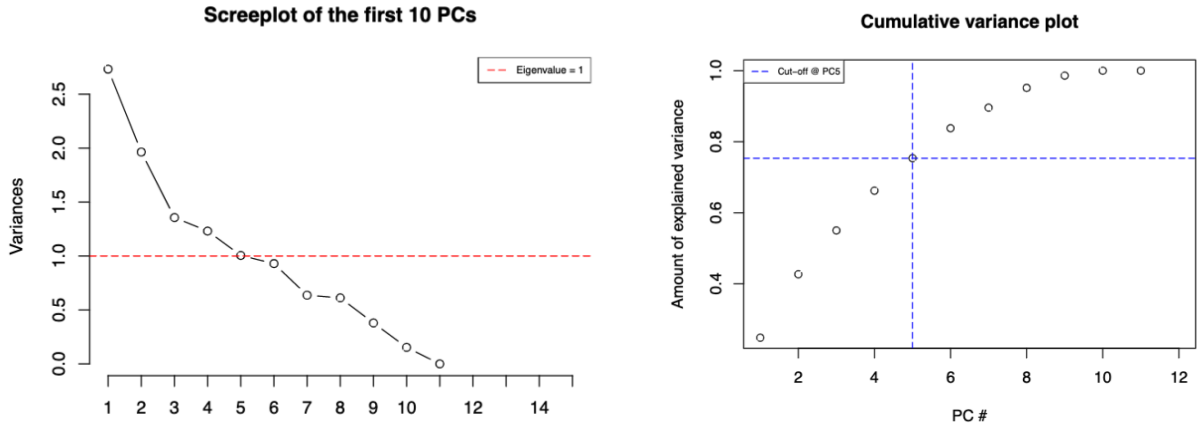

Fig. 5: Scree plot and the cumulative variance plot with cutoff = 5 of the first 10 PCs based on the PCA.

5.4 Parameter Tuning for SVM

According to Martins et. al. Martins et al. (2016), the referred SVM function has the following parameters to be defined: the error tolerance (ε or C), the pyramid depths (P), the radial basis function parameter (γ), and the threshold. The most effective combination of input parameters to RBF was C = 100; P = 2, γ = 0.1, threshold = 0.05. Therefore, our parameters tuning process was based on these references. The kernel functions we have applied include linear, radial, polynomial and sigmoid. The best performance kernel was polynomial with degree is 4, coef 0 = 4, misclassification rate no larger than 19%. Thus, in the next ensemble learning part, we added both the polynomial kernel(best tuning parameters) and the linear kernel.

5.5 Random Forest MSE

Figure 6 shows the MSE by the number of trees in the model. You can see that as the trees are added, significant improvement in MSE occurs early on and then flatlines just before 400 trees are built in the forest, and the optimal tree can be specified in the model. Here our model's optimal number of trees is ntree = 334. Based on this optimal ntree, the error rate in the training set is just 0.78%. Figure 7 and is the Random forest variance importance plot and Gini plot of random forest, which indicated the 5 selected features are Accuracy, Trustworthy, Anxious, WordCount and Attractive.

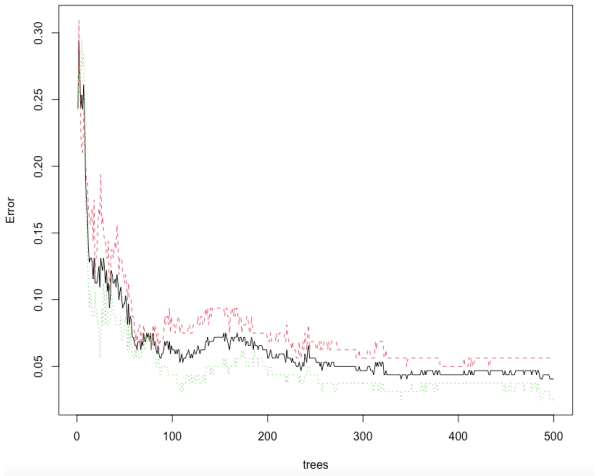

Fig. 6: MSE by the number of trees in the random forest model.

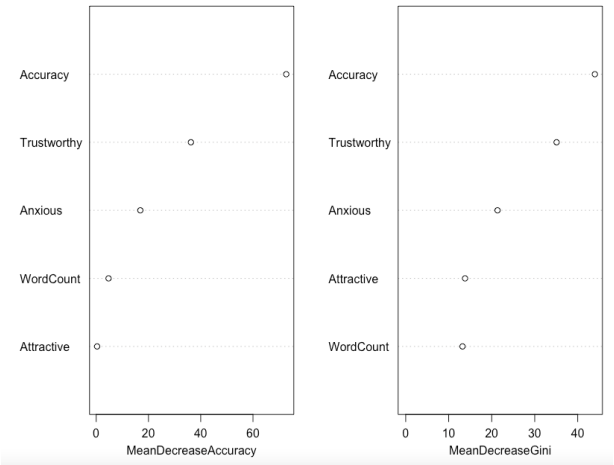

Fig. 7: RF variance importance plot and Gini plot.

5.6 Ensemble Learning with Selected Algorithms

Since RF itself is an ensemble learning method, the widely discussed problem for RF is its overfitting problem. Some researchers believe that RF does not overfit because of the two random processes in random sampling for each tree node and random selecting features for each splitting. However, since the random forest is based on the decision tree, the decision tree has been proved to have an overfit problem. In addition, most machine learning scientists believe that those two random processes in RF can only help to reduce the chance of overfitting but can not avoid it. Based on this opinion, this paper implements the

Random forest-based ensemble learning method, which helps improve the model prediction performance while reducing the chance of model overfitting. The core idea is by the different types of ensemble, such as Averaging, Majority vote, and Weighted average.

Ensemble learning is the key ingredient for winning almost all of the machine learning hackathons, and makes the model more robust and stable thus ensuring decent performance on the test cases in most scenarios. One can use ensembling to capture linear and simple as well nonlinear complex relationships in the data. This can be done by using two different models and forming an ensemble of two. In this paper, we include 6 different random-forest based ensemble learning models : RF+SVM.Linear(SVM with linear kernel), RF+SVM.Poly(SVM with polynomial kernel), RF+GLM(Generalized Linear Model), RF+KNNs(k-Nearest Neighbors), RF+GBM(Stochastic Gradient Boosting) and RF+WSRF(Weighted Subspace Random Forest).

Overall, Ensemble method is a meta-algorithm that combines several machine learning techniques into a prediction model to achieve the effect of reducing variance (bagging), boosting (boosting) or improving prediction (stacking).

6. Results and Conclusions

Our results show that RF+GBM(Stochastic Gradient Boosting) provides the highest prediction among the other ensemble learning methods. Table 2 shows the experimental results with accuracy, sensitivity, specificity and kappa value for each ensemble learning.

Table 2: Evaluation of Random-forest based Ensemble Learning Methods on MU3D Video Level Database.

Method	Accuracy	Sensitivity	Specificity	Kappa
RF+GLM	0.7109	0.7344	0.6875	0.4219
RF+KNNs	0.7422	0.7562	0.7281	0.4844
RF+SVM.Poly	0.7969	0.9938	0.6000	0.5938
RF+SVM.Linear	0.7953	1.0000	0.5906	0.5906
RF+WSRF	0.9578	0.9625	0.9531	0.9156
RF+GBM	0.9734	0.9812	0.9656	0.9469

According to Table 2 and Figure 8, our result shows that RF+GBM and RF+WSRF perform better than the other ensemble learning models, with overall accuracy 0.9734 and 0.9534, sensitivity 0.9812 and 0.9625, specificity 0.9656 and 0.9531, Kappa 0.9469 and 0.9156, respectively. Although RF+SVM.Linear and RF+SVM.Poly both have the highest sensitivity, the specificities are the lowest among all of the six methods, which means there are few false negative results and more false positive results, that is, the model correctly predict the lie cases but misclassified the truth.

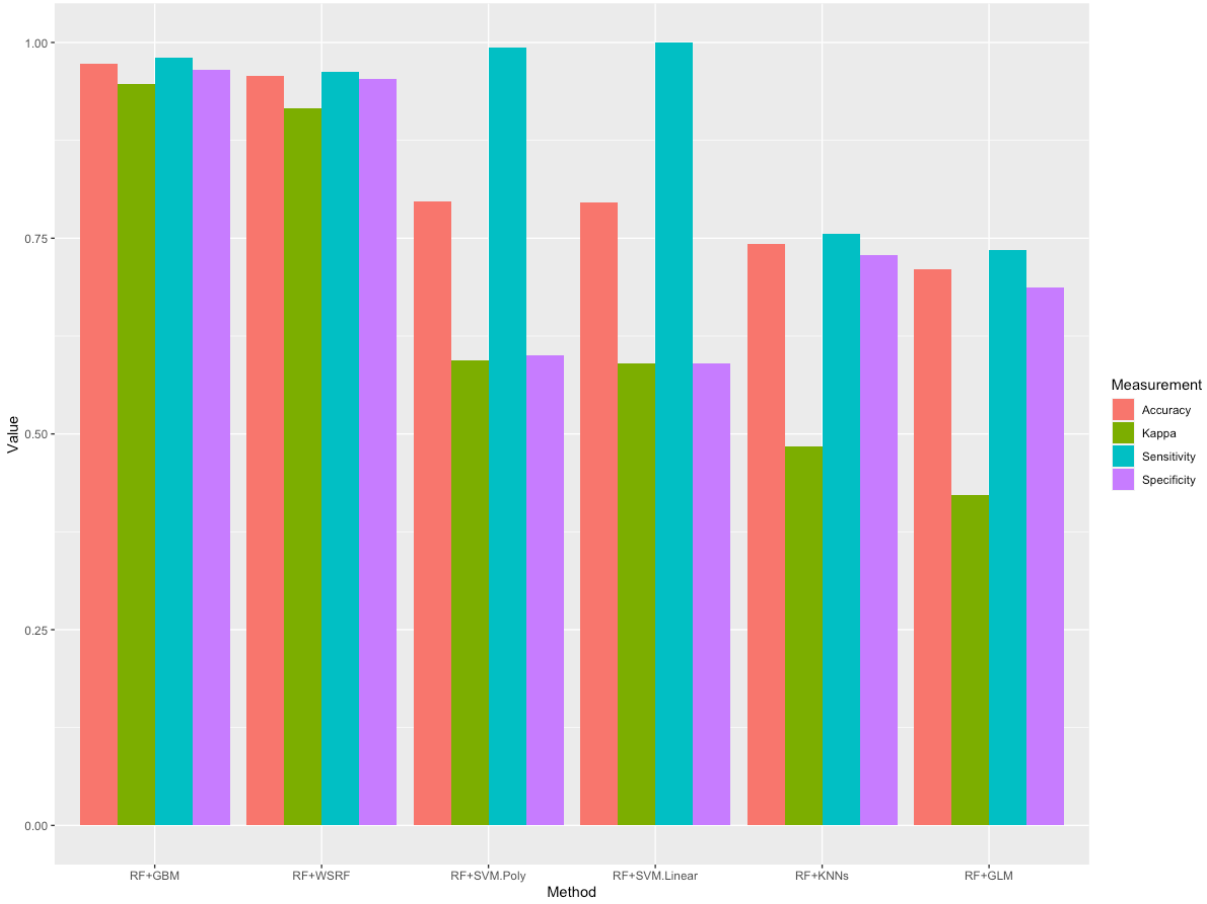

Fig. 8: RF-based ensemble learning performance comparison

Reference

[1] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day read-mission. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pages 1721–1730, 2015.

[2] Yi-Wei Chen and Chih-Jen Lin. Combining svms with various feature selection strategies. In Feature extraction, pages 315–324. Springer, 2006.

[3] Manuel Fern´andez-Delgado, Eva Cernadas, Sen´en Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? The journal of machine learning research, 15(1):3133–3181, 2014.

[4] Kurt Hugenberg, Allen R McConnell, Jonathan W Kunstman, E Paige Lloyd, Jason C Deska, and Brandon Humphrey. Miami university deception detection database. 2017.

[5] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning, volume 112. Springer, 2013.

[6] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. Advances in neural information processing systems, 29, 2016.

[7] Robert M Krauss, Valerie Geller, and Christopher Olson. Modalities and cues in the detection of deception. In 84th Annual Convention of the American Psychological Association, Washington, DC, 1976.

[8] Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In Proceedings of the conference on fairness, accountability, and transparency, pages 29–38, 2019.

[10] Sarah Martins, Nariane Bernardo, Igor Ogashawara, and Enner Alcantara. Support vector machine algorithm optimal parameterization for change detection mapping in funil hydroelectric reservoir (rio de janeiro state, brazil). Modeling Earth Systems and Environment, 2(3):1–10, 2016.

[11] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. arXiv preprint arXiv:1107.4557, 2011.

[12] Sushma Venkatesh, Raghavendra Ramachandra, and Patrick Bours. Robust algorithm for multimodal deception detection. In 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pages 534–537. IEEE, 2019.

[13] Tao Xiong and Vladimir Cherkassky. A combined svm and lda approach for classification. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 3, pages 1455–1459. IEEE, 2005.