A Practical Ensemble Modeling Approach to Remediate Low Error Tolerance

Ashlee Edwards, Spencer Smith, Jason Orender
Frontier Technology, Inc.
Chesapeake, VA
aedwards@FTI-net.com,
ssmith@FTI-net.com,
jorender@FTI-net.com

Yanhua Feng, Cristina Rider
Naval Safety Command
Norfolk, VA
yanhua.feng.civ@us.navy.mil,
cristina.a.rider.civ@us.navy.mil

ABSTRACT

Modeling for safety risk often involves infrequent events with sparse data, and there is simultaneously a low tolerance for both type 1 and type 2 errors. Frequent type 1 errors will cause a numbness to develop with respect to any warnings generated by the model, while frequent type 2 errors are a symptom of chronic underestimation of risk. Managing type 1 and type 2 errors in a balanced way is the objective of ensemble modeling. This method can help reduce type 1 errors so that a more sensitive model can be produced, which also in turn remediates the chronic risk underestimation. Ensemble modeling can leverage multiple techniques that tend to agree on the true positive and true negative results, while tending to have a diverse response with respect to false positive and false negative results. When these results are unified under a top layer model, each lower layer model's classification results are interpreted in context with all lower layer model results. The outcome of this process is a more robust model which amplifies the true positive and negative results that are common between ensemble components while deemphasizing the results that the components disagree on. This paper will review some generally accepted ensemble modeling strategies and provide a practical walkthrough of a layered ensemble process using Naval safety data and the R programming language, along with an evaluation of the limits associated with the improvement that can be expected from this technique.

ABOUT THE AUTHORS

Ashlee Edwards is a Data Scientist at Frontier Technology Inc. She has received her Master's and PhD in Applied Computational Mathematics at Old Dominion University.

Jason Orender is a Senior Advisory Data Scientist at Frontier Technology, Inc., and he works on the Naval Safety Command's data science team as a technical lead. He retired in 2014 from the US Navy as a Nuclear Power Program Designated Surface Warfare Officer. He has since acquired his Master's in Computer Science and is a PhD candidate at Old Dominion University

Spencer Smith is a Software Developer with Frontier Technology Inc. Graduated from Old Dominion University in 2016 with a Bachelor's of Science in Modeling, Simulation, and Visualization Engineering.

Yanhua Feng is an Operations Research Analyst/Data Scientist with Naval Safety Command. Graduated from College of William and Mary in 2022 with a Master's degree in Computer Science.

Cristina Rider is an Operations Research Analyst/Data Scientist with the Naval Safety Command. Graduated from University of Michigan in 2011 with a Bachelor's of Science degree in Pure Mathematics. Prior Naval Officer and designated Surface Warfare Officer.

A Practical Ensemble Modeling Approach to Remediate Low Error Tolerance

Ashlee Edwards, Spencer Smith, Jason Orender
Frontier Technology, Inc.
Chesapeake, VA
aedwards@FTI-net.com,
ssmith@FTI-net.com,
jorender@FTI-net.com

Yanhua Feng, Cristina Rider
Naval Safety Command
Norfolk, VA
yanhua.feng.civ@us.navy.mil,
cristina.a.rider.civ@us.navy.mil

INTRODUCTION

Problem Statement

Data is essential for model training in machine learning. However, in real applications the distribution of two classes can be severely imbalanced (Ganganwar, 2012). Machine learning with imbalanced datasets is difficult but can provide insights which cannot be obtained any other way. Examples of an area in which imbalanced datasets are a problem include rockbursts, insurance fraud, disease epidemics, and safety. These datasets are often comprised of an uneven mix of rare events and frequent non-events (King, 2001). In many cases, the rare events are high-severity and can have catastrophic consequences. Using machine learning approaches to classify events would require training algorithms with this type of data, but the scarcity of events in an imbalanced dataset make learning rare events difficult.

Training using multiple machine algorithms and then utilizing some rules to combine them is one way to combat the issue of high bias and variance. Combining the algorithms could also combine the strengths of several weaker models and learn the structure of the data more robustly. Ensemble techniques are designed to help reduce errors, and thus enhance the overall accuracy of the final model (Diettrich, 1997). However, there are some necessary conditions to achieve better performance accuracy: 1) The machine learning results which will be combined must differ in some way, whether that be different algorithms or different subsamples, and greater differentiation should correlate with a greater chance of model improvement, and 2) The machine learning algorithms that will be combined should perform better than random guessing.

These conditions will ensure that generalization of error of the prediction is reduced. In order to build a more stable, reliable, and accurate model, we will train different machine leaning algorithms using different splits of training and test data from engineered (synthetic) data. These models will then be ensembled together and the effectiveness of the ensemble will be analyzed in comparison to the individual model results. The conditions bounding the usefulness of this method will also be explored.

Related Works

Because of the various techniques and avenues available to acquire data, there has been an explosion of different types and quantities of data. However, in the real-world, data is often imbalanced. The problem of imbalanced data is certainly among the many challenges in machine learning and data mining (Yang, 2006). When this type of data is used to train individual machine learning algorithms, the accuracy and recognition of the minority class is negatively affected (He, 2013). Popular methods for addressing the data imbalance include modifying classification algorithms to be sensitive to imbalanced data (Branco, 2017), subsampling the data (Kotu and Deshpande, 2019), weighting the classes of imbalanced data to balance the classes (He and Cheng, 2021), generating synthetic data to increase the majority class (Sun 2015), and using ensemble methods (Sun 2015). An ensemble approach can be used in practice by combining one or more of these methods together to improve the accuracy of both classes (Lee, 2010; Salunkhe, 2016).

Two common resampling techniques include over- and under- sampling. Over-sampling involves duplicating the rare event class samples while under-sampling discards the majority class samples in order to modify the class distribution. Neither method results in significant performance changes as over-sampling leads to overfitting and under-sampling leads to training with less data (Kotu and Deshpande, 2019).

Machine learning algorithms used to predict rare events often underestimate the probability of the rare event (He and Cheng, 2021). Such algorithms will always have overall high prediction accuracy because of the correct prediction on the large number of nonevents, but the true positive rate is extremely low (Branco, 2006), even with respect to the low numbers of the minority class overall. Handling an imbalanced set using just machine learning algorithms which preferentially choose the dominant class is not useful as the main interest is the prediction or identification of the rare events.

Motivation

Ensemble models are attractive because they can boost the individual algorithms from weaker models (slightly better than random guesses) to stronger models with better predictive ability (Kotu and Deshpande, 2019). In this paper, we use a stacking ensemble method to not only combine the strengths of the individual algorithms (improved model accuracy) using generated data and anonymized real data, but also compensate for the weaknesses of the individual algorithms (and successfully minimize type 1 and type 2errors).

Contribution

The main contributions to this paper are:

- An introduction of ensemble learning techniques
- To propose an ensemble model that takes advantage of key components of ensemble learning.
- To develop and train an ensemble model through controlled experiments using engineered data.
- To evaluate the effectiveness of the ensemble model and demonstrate its application to real data.

Paper Organization

The paper is organized as follows: After the Introduction section, the Background section will give a review of the concepts and techniques needed to understand the experiment. A detailed explanation of the procedures used to implement the experiment will follow. The results section will discuss the outcomes of the experiment as well as core findings. Finally, the conclusion section will summarize the paper and discuss the potential next steps in research.

BACKGROUND

Ensemble modeling is a process where multiple diverse component models, built with different algorithms or using different training data sets, are unified to create a single model to predict an outcome. The ensemble model aggregates the predictions of each component model in making a final prediction for the unseen data. The motivation for using ensemble models is to reduce the generalization error of the prediction. As long as the base models are diverse in some way, the prediction error decreases when the ensemble approach is used. Even though the ensemble model has multiple base models within the model, it acts as a single model. Ensembles appear complex yet tend to strongly outperform their component models on new data. Many practical data science applications utilize ensemble modeling techniques (Kotu and Deshpande, 2019).

Ensemble Methods Techniques

An ensemble method is a machine learning technique that combines several component models in order to produce a superior predictive model. Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models. There are two broad categories sequential ensemble techniques and parallel ensemble techniques; and various ensemble methods such as stacking, bagging, and boosting.

The majority of ensemble techniques apply a single algorithm in base learning which results in homogeneity in all component models (CFI Team, 2022). Homogenous component models refer to component models of the same type,

with similar qualities. Other methods apply heterogeneous component models, giving rise to heterogeneous ensembles. Heterogeneous component models are models of differing types.

Sequential ensemble techniques
The sequential ensemble techniques generate component models in a sequence, e.g., extreme gradient boosting and adaptive boosting. It promotes the dependence between the component models. The performance of the model is then improved by assigning higher weights to previously misrepresented examples (CFI Team, 2022).

Parallel ensemble techniques
The parallel ensemble techniques generate component models in a parallel format, e.g., random forest. They utilize parallel generation of component models to encourage independence between the component models. The independence of component models significantly reduces the error due to the application of averages (CFI Team, 2022).

Ensemble Algorithms

Boosting
Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future. It combines several weak component models to form one strong learner which improves the predictability of models (CFI Team, 2022). Boosting works by arranging weak models in a sequence and learn from the next learner in the sequence to create better predictive models. Boosting takes many forms including adaptive boosting and gradient boosting.

The gradient boost tree model objective is to minimize the loss function of the model by adding weak models using gradient descent (Chen, T. and Guestrin, C., 2016). Gradient descent is a first order iterative optimization algorithm for finding a local minimum of a differentiable function (Kwiatkowski, 2021). As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc.

A gradient boost tree model is a gradual, additive and sequential model that generates models during the learning process. The contribution of the weak learner to the ensemble is based on the gradient descent optimization process. The calculated contribution of each tree is based on minimizing the overall error of the strong learner.

Gradient boost tree models do not modify the sample distribution as weak models train on the remaining residual errors of a strong learner. By training on the residuals of the model, this is an alternative means to give more importance to misclassified observations. New weak models are being added to concentrate on the areas where the existing models are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner (Chen, T. and Guestrin, C., 2016).

Extreme Gradient Boosting (XGBoost) is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. It is designed to enhance the performance and speed of a machine learning model (CFI Team, 2022). XGBoost uses pre-sorted algorithm and histogram-based algorithm for computing the best split. The histogram-based algorithm splits all the data points for a feature into discrete bins and uses these bins to find the split value of the histogram. Also, the trees can have a varying number of terminal nodes and left weights of the trees that are calculated with less evidence if shrunk more heavily (Chen, T., and He, T., 2021).

Bagging
Bagging is the shortened term for bootstrap aggregation. “Bootstrapping” is random sampling with replacement, and “aggregation” is how the trees in the forest are combined to make a final prediction. For each tree in the forest, bootstrapping is applied to the training set. Each tree is trained separately prior to aggregating the results to make a final vote. Thus, bootstrapping allows for slightly different training sets for each tree in the forest effectively creating distinct trees, which is important to reducing variance. Additionally, at each node split, a random sample of features are selected to create branches of the trees. This further ensures that each tree in the forest will produce sufficiently distinct trees to reduce variance and produce more accurate results (Nisbet, R., Miner, G., & Yale, K., 2018).

Random forest models are ensemble models that use bagging, multiple decision trees, and majority voting to build a low bias (requires few assumptions on the target variable), high variance (sensitive to changes in training data) classifier. Advantages of random forest models include that they are non-parametric, they are known for being accurate with only a few tuning parameters, and they are able to capture non-linear trends in data. Several parameters can be tuned in a random forest model to improve model accuracy and reduce overfitting (Nisbet, R., Miner, G., & Yale, K., 2018). These are a favorite for use as a top-level model type to combine the component models. (Branco, Torgo, and Ribeiro, 2017).

Stacking

Stacking is often referred to as stacked generalization. This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions. The practice of stacking multiple models has shown the ability to provide improved performance as compared with a single model (Kotu and Deshpande, 2019; Nisbet, Miner, and Yale, 2018).

Longitudinal Data and Imbalance

Longitudinal data, also known as panel data, tracks the same sample at different points in time. It allows for the measurement of within-sample change over time, enables the measurement of the duration of events, and records the timing of various events.

The number of instances in which a mishap did not occur vastly outnumbers the instances in which one did, creating an imbalance. Collecting as many definitively negative examples as possible will still enhance the resulting model, even if the number of those negative examples greatly exceeds the number of positive examples. If a model reliably predicts all negative examples, it will have a high accuracy percentage and bias due to the imbalanced nature of the dataset, but it will not provide useful insights for preventing future mishaps (Ganganwar, 2012).

There is no particular technique that will always work for imbalanced datasets, but a combination of various methods can be used as a starting point for enhancing the utility of the models. Several approaches to deal with imbalanced dataset problems (Pandey, 2022):

1. Evaluation Metrics –
    a. Recall/Sensitivity – for one class, how many samples are predicted correctly
    b. Precision/Specificity – allows calculation of a class specific problem
    c. F1 Score – Harmonic mean of recall and precision
    d. Matthews correlation coefficient (MCC) – calculation of the correlation coefficient between predicted and observed binary classifications
    e. Area under the curve (AUC)-Receiver operating characteristic curve (ROC) – being independent of changes in proportion of responders, it infers the relation between false positive rate and true positive rate
2. Resampling –
    a. Oversampling – implemented when the quantity of data is insufficient. The minority group is increased to balance the dataset
    b. Undersampling – reduces the size of the majority group; thus the size is balanced by selection of an equal number of samples from the majority group to create a new dataset for further modeling.
3. Synthetic Minority Oversampling Technique (SMOTE) – random point is picked from the minority group and the K-nearest neighbor is calculated, followed by the addition of random points around the chosen point. The relevant points are added without altering the accuracy of the model. This method therefore provides better results when compared to simple undersampling and oversampling.
4. K-fold cross validation – cross validating the dataset after it is generated by the process of oversampling since it makes predicting the minority group easier. It prevents data leakage from the validation set.
5. Ensembling resampled datasets – by using multiple learning algorithms and models to obtain better performance on the same dataset after it is resampled using oversampling or undersampling.
6. Choosing the right model – there are models that are suited to work with imbalanced datasets and do not require you to make changes to the data, like XGBoost.

IMPLEMENTATION

Data in the study was generated using build_engineered_data_2.R. 4911 examples were generated. 491 examples are in the positive class and 4420 examples are in the negative class. The ratio of the majority class (negative class) to the minority class (positive class) is 9:1.

Engineered features were designed to mimic longitudinal data. There are 20 variables (Vm, m = 1...20). V5, V8, V9, V10 and V13 are relevant variables, and the remaining are non-relevant variables. Each variable has 11 time-steps (for example, for V1, there are V1TMn, n = 0...11), each a separate feature for a total of 11 features. Therefore, there are 220 features in total.

Training set includes 10% of the positive examples and 10% of the negative examples, randomly sampled from each class. Test set includes the remaining examples. Features in the training set and test set include the time steps of all the relevant variables and randomly selected 13 non-relevant variables.

Performance Comparison between Stacking Ensemble and Its Component Models

Stacking ensemble learning includes two levels of learning. In the first level, three individual models were created on the same training data with each model using a different classifier. Three classifiers used are Least Absolute Shrinkage and Selection Operator (LASSO), Ridge Regression and Extreme Gradient Boosting (XGboost). The LASSO model and Ridge Regression model were carried out using the R "glmnet" library and the function of the same name. XGboost model was carried out using the R "xgboost" library and the function of the same name. The first-layer models are referred to as the "component models". In the second layer, a random forest tree model was created using Algorithm 1, where each of the component models was used to predict the responses for the examples in the training set, and then a random forest tree model was fit on the predictions of three component models combined. The block diagram of the stacking ensemble is showed in Figure 1.

Algorithm 1 The Ensemble Fit

Input: Training data $(\boldsymbol{X}, \boldsymbol{y})$, $\boldsymbol{l}$ **Output: Z,** $\boldsymbol{m}$

1. Use each of the models in the list $\boldsymbol{l}$, $\boldsymbol{l_j}$ (where $\boldsymbol{j}$ is the index of the model) to predict responses for the examples in the matrix $\boldsymbol{X}$ and put the predicted responses in a vector $\boldsymbol{p_j}$.
2. Stack all the predicted response vectors into a matrix $\boldsymbol{P}$.
3. Combine P and y to a data frame $\boldsymbol{Z}$.
4. Fit a random forest model on $\boldsymbol{Z}$ and return a model $\boldsymbol{m}$.

Both the stacking ensemble model and its component models were used to predict the responses for the examples in the test set. The resulting test set true positive rate (TPR), false positive rate (FPR) and F1 score were used to evaluate the model performance.

Performance Comparison between Sampled Ensemble and Its Component Models

Sampled ensemble learning is a variation of bagging ensemble. It also includes two levels of learning. Different from the stacking ensemble learning, in the first level, three individual models were created using a single classifier (LASSO), but each model was fit on a different subset of the same training data. Each subset is randomly sampled 90% of the training data without replacement. In the second level, a random forest decision tree model was fit on the predictions of the component models, which is an implementation of stacked ensemble learning. The block of the sampled ensemble is showed in Figure 1.

Both the sampled ensemble model and its component models were used to predict the responses for the examples in the test set. The resulting test set true positive rate (TPR), false positive rate (FPR) and F1 score were used to evaluate the model performance.

For both the stacking ensemble experiment and the sampled ensemble experiment, five trials were carried out. Training data and test data were different in each trial because of the random split and non-relevant variables being randomly selected. The random forest fit was also different in each trial.

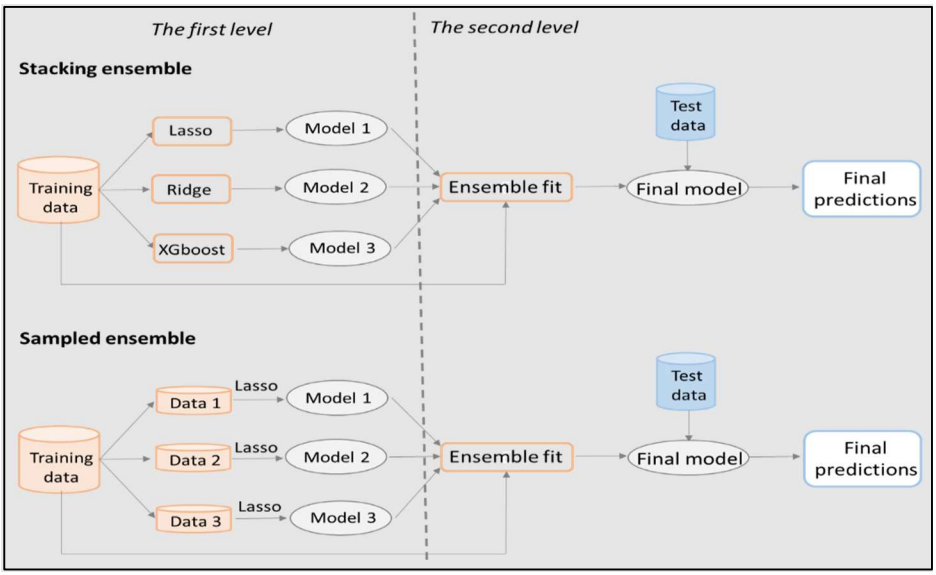

Figure 1. Block Diagram of Ensemble Learning

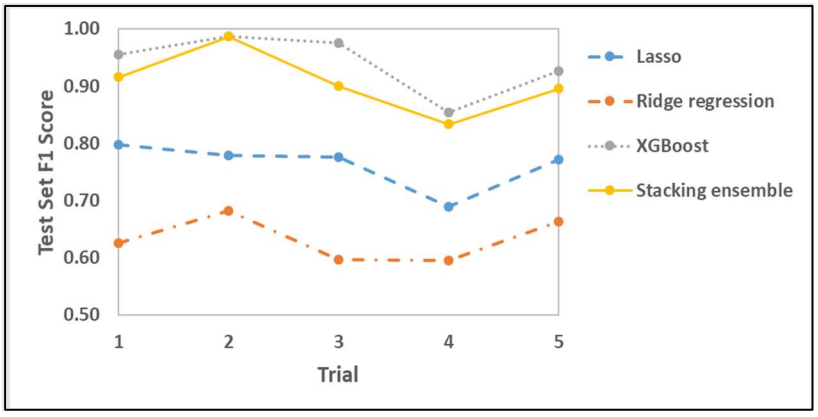

Figure 2. Test Set F1 Score of Stacking Ensemble and Its Component Models

RESULTS

The core takeaways of the two ensemble modeling algorithms proposed above are illustrated in Figures 2, 3 and 4. Figure 2 illustrates the results of the stacked ensemble experiment, while Figures 3 and 4 pertain to the sampled ensemble experiment results. The efficacy of each model is judged by its F1 score, a balanced evaluation metric utilized to measure a model's efficacy. Each trial produces an F1 score for each model, so we can discern the overall performance of each model by observing the F1 score across every trial.

Performance Comparison between Stacking Ensemble and Its Component Models

As illustrated with Figure 1 the stacking ensemble model is made up of three component models: LASSO, Ridge Regression, and Extreme Gradient Boost. In Figure 2 the test set F1 score of the stacked ensemble model, and its individual component models, is measured across each trial. The stacking ensemble out-performs the LASSO and Ridge Regression component models but is consistently lesser when compared to the Extreme Gradient Boost component model using the F1 score. This suggests that the independent superior performance of the Extreme Gradient Boost model will more often than not be preferential over a stacking ensemble composed of lesser component models.

Sampled Ensemble Out-performs Its Component Models

As illustrated in Figure 1, the sampled ensemble model is composed of three LASSO models, fit on three different subsets of the same dataset. In Figure 3 the F1 score of the sampled ensembles, and its individual component models, is measured across each trial. The sampled ensemble out-performs each of the component models, suggesting that an ensemble of sampled training data is consistently superior to the individual models fit to sampled training data.

Table 1. True Positive Rate and False Positive Rate of Stacking Ensemble and Its Component Models

	LASSO		Ridge regression		XGBoost		Stacking ensemble	
Trial #	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**
1	82.6	3.3	88.5	12.7	95.2	0.6	96.7	2.0
2	87.2	4.9	92.0	10.4	97.3	0.0	97.3	0.0
3	82.0	3.9	87.0	14.0	95.2	0.0	96.2	2.4
4	91.2	9.9	87.6	14.3	95.0	3.7	94.3	4.3
5	86.2	5.0	89.1	10.7	89.5	0.5	93.1	2.0
Mean	85.9	5.4	88.8	12.4	94.5	0.9	95.5	2.1

Table 2. True Positive Rate and False Positive Rate of Sampled ensemble and Its Component Models

	Model fit on subset 1		Model fit on subset 2		Model fit on subset 3		Sampled ensemble	
Trial #	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**	**TPR (%)**	**FPR (%)**
1	83.7	4.0	80.1	3.1	82.8	3.5	80.7	2.5
2	87.4	4.6	86.8	5.5	88.9	5.1	84.3	2.9
3	83.7	4.3	81.6	3.8	77.4	4.4	79.2	3.0
4	90.6	8.1	89.7	11.1	88.3	10.0	84.9	5.1
5	91.6	4.6	86.4	4.9	88.7	5.6	89.1	2.7
Mean	87.4	5.1	84.9	5.6	85.2	5.7	83.6	3.2

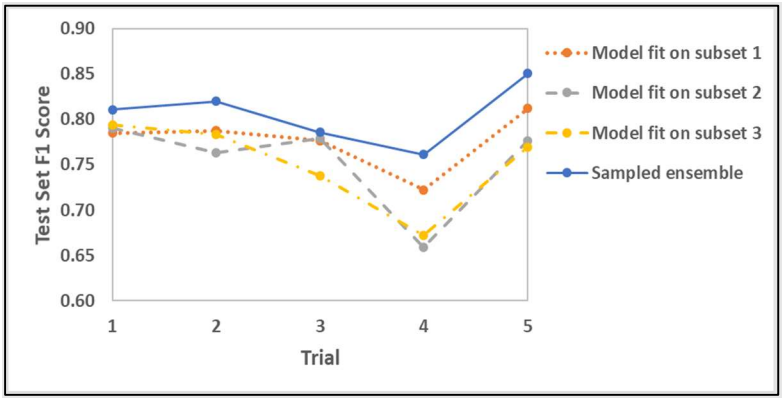

Figure 3. Test Set F1 Score of Sampled Ensemble and Its Component Models

Sampled Ensemble Out-performs Model Fit on the Entire Training Set

One more experiment was performed with the sampled ensemble, where the sampled ensemble was compared to a model fit on the entire data set, rather than different subsets. In Figure 4 the F1 score of the sampled ensemble, and a model fit on the entire training set, is measured across each trial. The sampled ensemble out-performs the model that was created on the entire training set. So, by sampling the training set in different ways, and making an ensemble of the resulting models, it can out-perform a model that was made using 100% of the training set.

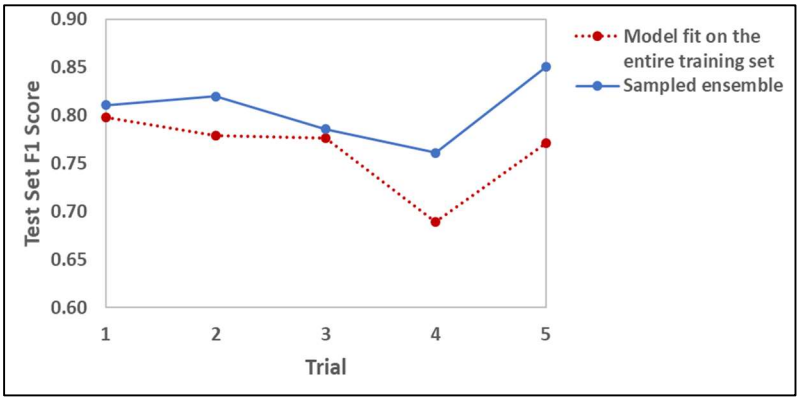

Figure 4. Test Set F1 Score of Sampled Ensemble and Model Fit on the Entire Training Set

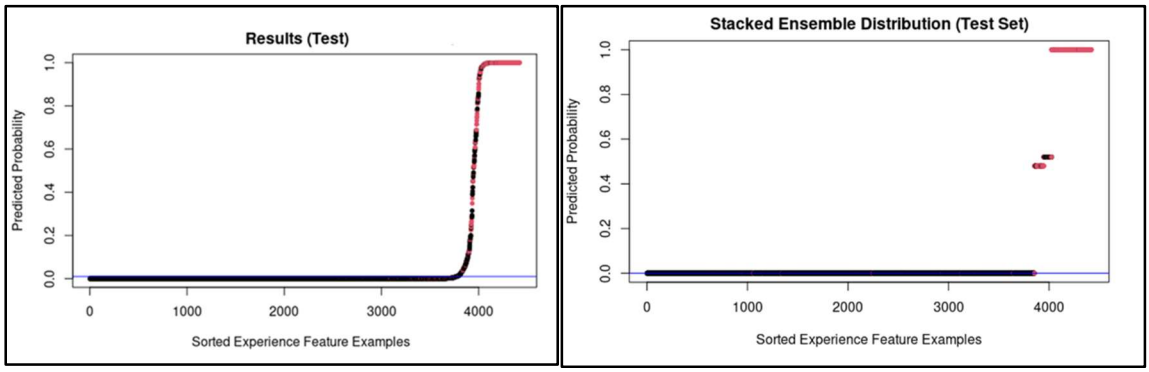

Figure 5. Sorted model responses for a test set using a single learner vs using the stacked ensemble model

CONCLUSION

In this paper, we have presented an ensemble method for dealing with imbalanced longitudinal data applications. The proposed method focuses on drawing together the predictions of multiple weaker component models to improve overall model accuracy.

Given that the mishaps studied are rare events and the choice to use longitudinal data, the strategy to create model ensembles to assess risk has resulted in higher performing models with more separation between the example classes (behaves like a mishap example or does not) as shown in the figure above.
From the experimental results, the proposed ensemble methods generally do achieve better model accuracy than the component models. We have shown that the chosen stacked ensemble model made up of component models with either (a) diverse learning methods or (b) different training sets indeed reduce the generalization error that arises due to overfitting the training data.

ACKNOWLEDGEMENTS

We would like to thank the Naval Safety Command for their support and collaboration with this work.

REFERENCES

Branco, P., Torgo, L., & Ribeiro, R. P. (2017). Relevance-based evaluation metrics for multi-class imbalanced domains. *In Advances in knowledge discovery and data mining*, 698–710.

Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 785-794.

Chen, T., and He, T., "xgboost: eXtreme Gradient Boosting," CRAN Package Version 1.4.1.1, 2021.

CFI Team. (2022). Ensemble Methods: Combining multiple models to improve the desired results.

Dietterich, T. (1997). Machine-Learning Research. *AI Magazine, 18*. 97-136.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering, 2,* 42-47.

He, H. & Ma, Y. (2013). *Imbalanced Learning: Foundations. Algorithms, and Applications,* Hoboken, New Jersey: John Wiley & Sons, Inc.

He, J., & Cheng, M. (2021). Weighting Methods for Rare Event Identification From Imbalanced Datasets. *Frontiers in Big Data, 4*.

King, G. & Zeng, L. (2001) Logistic Regression in Rare Events Data. *Political Analysis, 9,* 137-163.

Kotu, V., Deshpande, B. (2019). *Data Science: Concepts and Practice*, Cambridge, MA: Elsevier, Inc.

Kwiatkowski, R. (2021). Gradient Descent Algorithm – a deep dive, *Towards Data Science.*

Lee, C.Y., Yang, M.R. (2010). A Hybrid Algorithm Applied to Classify Unbalanced Data. *Networked Computing and Advanced Information Management,* 618-621.

Nisbet, R., Miner, G., & Yale, K. (2018). Handbook of Statistical Analysis and Data Mining Applications, 2nd ed., Science Direct.

Pandey, M. (2022). 6 Techniques to Handle Imbalanced Data. Developers Corner.

Salunkhe, U. R., & Mali, S. N. (2016). Classifier ensemble design for imbalanced data classification: A hybrid approach. *Procedia Computer Science*, *85*, 725–732.

Sun, Z., & Song, Q. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition, 48,* 1623-1637.

Yang, Q., Wu, X. (2006). 10 Challenging problems in data mining research. *International Journal of Information Technology & Decision Making, 5,* 597-604.

Yin, X., & Liu, Q. (2021). Strength of Stacking Technique of Ensemble Learning in Rockburst Prediction with Imbalanced Data: Comparison of Eight Single and Ensemble Models. *Natural Resources Research*, *30*, 1795-1814.