An online-tool for tuning ensemble learning algorithms

Muhammed Maruf ÖZTÜRK[1]

[1] Suleyman Demirel University, Isparta/Turkey, muhammedozturk@sdu.edu.tr

Abstract **– Machine learning algorithms have configurable parameters. Known as hyperparameters, they are generally used with their default settings. However, in order to increase the success of a machine learning algorithm, it is required to develop sophisticated techniques to tune hyperparameters. Tuning a machine learning algorithms need great effort. However, existing methods can only be performed via discrete programming tools. In this paper, a user-friendly hyperparameter tuning tool is proposed for ensemble learning. It encompasses selecting tuning algorithm, data set, and performance visualization. Besides them, developed tool is compatible with executing R codes to conduct big data experiments.**

Keywords **– Hyperparameter tuning, ensemble learning, defect prediction.**

I. INTRODUCTION

Machine learning algorithms are devised with some changeable elements [1]. For instance, in random forest, depth of the trees, number of iterations, and learning rate are some of the changeable elements. They are called hyperparameters [2]. If any classifier is used, it presents various options to its practitioners. Generally, a classifier is used with default settings of hyperparameters. However, it is not sufficient to use a classifier with default settings in case of performance bottlenecks [3].

In such cases, hyperparameters are exposed to a tuning process called hyperparameter optimization (HO) [4]. HO consists of searching a set of values which will be used in the related operation. To search a parameter, random and grid search algorithms are common among researchers [5].

If one classifier is not sufficient to increase the success of a learning algorithm, combining more than one classifier may be a good solution [6]. It is defined as ensemble learning. However, in doing so, practitioners have some alternatives such as stacking, bagging, and boosting. They change the way of labeling instances. However, regardless of used approach, an ensemble learning algorithm requires a tuning process to achieve optimal configuration.

Over the past decade, researchers have strived to find optimal settings of hyperparameters [7], [8], [9], [10]. Further, various HO algorithms have been proposed in this period [11]. However, to the best of our knowledge, HO has only been investigated in terms of individual classifiers [12]. The works related to the HO lack examining adverse of favorable effects of tuning hyperparameters of ensemble learning algorithms.

Moreover, in this domain, there is a need for performing ensemble learning based on a user-friendly tool. It could help practitioners to figure out to what extent HO can improve machine learning performance. Note that researchers who work on ensemble learning can enrich and ease their knowledge by this way.

In this respect, this paper proposes a novel online-tool for tuning ensemble learning process. It is capable of tuning an ensemble algorithm with parameters selected by the user. Proposed tool configures ensemble learning algorithms including AdaBoost, GradientBoostLearner, and Random Forest. It also enables users to select a parameter search method. GridSearch, GlobalizedBoundedNelderMead, ParticleSwarm, and Bayesian are the search methods presented in the tool. It has been coded with .Net and included an R execution panel to harness R package scripts. Developed tool also provides ROC analysis to illustrate performance of an ensemble learning algorithm.

The rest of the paper is organized as follows: Section II presents related works. Proposed tool is elaborated in Section III. Threats to the validity are in Section IV. Last, Section V concludes the results.

II. RELATED WORKS

A. Hyperparameter Optimization

HO is an intriguing topic for machine learning researchers. In particular, various HO algorithms have been developed in the last decade [13], [14], [15].

Initially, some classifiers such as Random Forest and Naïve Bayes were much popular among practitioners. However, in recent years, online and cloud-based algorithms have frequently investigated in terms of HO.

Big data is an interesting topic of machine learning. To cope with big data, traditional methods were advised. Instead, some sophisticated methods, such as deep neural network, have been performed when the scale of the experimental data is large to be examined [16].

A deep neural network has a great number of layers compared with traditional neural network so that valuable information can be extracted via specific machine learning techniques.

Kaneko and Funatsu proposed a grid-search based HO method for support vector regression models [17]. They were able to increase both prediction performance and the speed of the classifier.

Springenberg et al. developed BOHAMIANN which is fast and scalable for Bayesian optimization [18]. It relies on a specific scale adaptation technique to improve the robustness of learning.

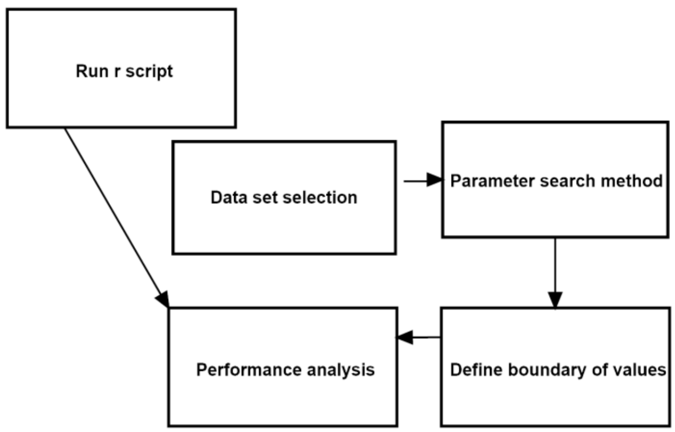

Figure 1: Main steps of the proposed tool.

B. Ensemble Learning

Ensemble learning has been employed in various fields since 1990’s. They include speech recognition [19], sentiment analysis [20], software engineering [21], and information systems [22].

Ensemble learning was also used in the classification of noisy data sets [22]. Thus, a model having high error tolerance and accuracy can only be obtained by that way.

In [23], a fuzz cluster-based ensemble learning approach namely IFCESR was proposed. It employs soft clustering techniques to create ensemble clusters. The effectiveness of the method was then tested on UCI data sets. According to the obtained results, IFCESR surpassed state of the art alternatives in terms of clustering accuracy.

Customer scoring is an interesting field in which ensemble learning was utilized [24]. The method using hybrid methods simultaneously has better performance results with AdaBoost than other methods. Moreover, PCA is much feasible for feature selection rather than information gain and GA. Fuzzy cognitive map was improved with ensemble approach [25]. In doing so, it was observed that the performance of fuzzy cognitive map decreases remarkably when it is employed with Hebbian learning.

Pratama et al. presented a new ensemble learning method namely pEnsemble [25]. It consists of three components: drift detection, ensemble pruning, online feature selection. The main advantage of pEnsemble is that it features less complexity than its alternatives.

III. METHOD

Proposed tool has been developed through SharpLearning (https://github.com/mdabros/SharpLearning). It is an open-source library coded with C#. The main goal of the library is to provide a great number of machine learning algorithms and models to practitioners. Algorithms and HO parameter search methods presented by SharpLearning are given in Table 1.

Proposed tool consists of three parts. First part encompasses the operations related to ensemble learning. A user can select a data set to be exposed to learning process. This part can also provide parameter search methods. Four parameter search methods are presented to user. Thereafter, hyperparameter bounds are defined. The tool gives four options to restrict parameter values. Since, HO includes a great number of parameters to be tuned. In the experiment, the most used ones are involved.

Table 1: Algorithms and parameter search methods of SharpLearning.

Method	Type
DecisionTrees	Learning algorithm
AdaBoost	Learning algorithm
GradientBoost	Learning algorithm
RandomForest	Learning algorithm
ExtraTrees	Learning algorithm
NeuralNets	Learning algorithm
GridSearch	Parameter Search
RandomSearch	Parameter Search
ParticleSwarm	Parameter Search
GlobalizedBoundedNelderMead	Parameter Search
BayesianOptimization	Parameter Search

For instance, iteration number is frequently applied by practitioners to demonstrate learning performance. By this way, critical bottlenecks of an algorithm can be detected. Learning rate is another prominent hyperparameter for learning algorithm. It is generally changes between 0.1-1. If a learning rate is close to 1, it means that the classifier is so sensitive to training data. Therefore, an optimal value should be selected to yield reliable testing results.

Besides them, if a tree-based classifier is used, maximum number of tree can be tuned to find optimal HO settings. Further, maximum tree of depth is important for tuning operation.

Developed tool provides a performance analysis including confusion matrix and ROC analysis. Predicted and actual values of testing instances can be seen from this analysis.

R package is a statistical tool which has gained great interests in recent years. It can also be used for big data and machine learning operations. Further, R provides rich options to visualize big data. For this reason, proposed tool includes an R management panel. By using this panel, an R script can be executed from .Net platform. Obtained results can be illustrated via the results returned by R package.

The IDEs providing executing R scripts are flexible and easy to use. However, a web-based online R framework is not available in terms of .Net compliance. The tool presented in this paper could fill this gap and encourage researchers to develop web-based user friendly machine learning tools. Main steps of the tool are given in Figure 1.

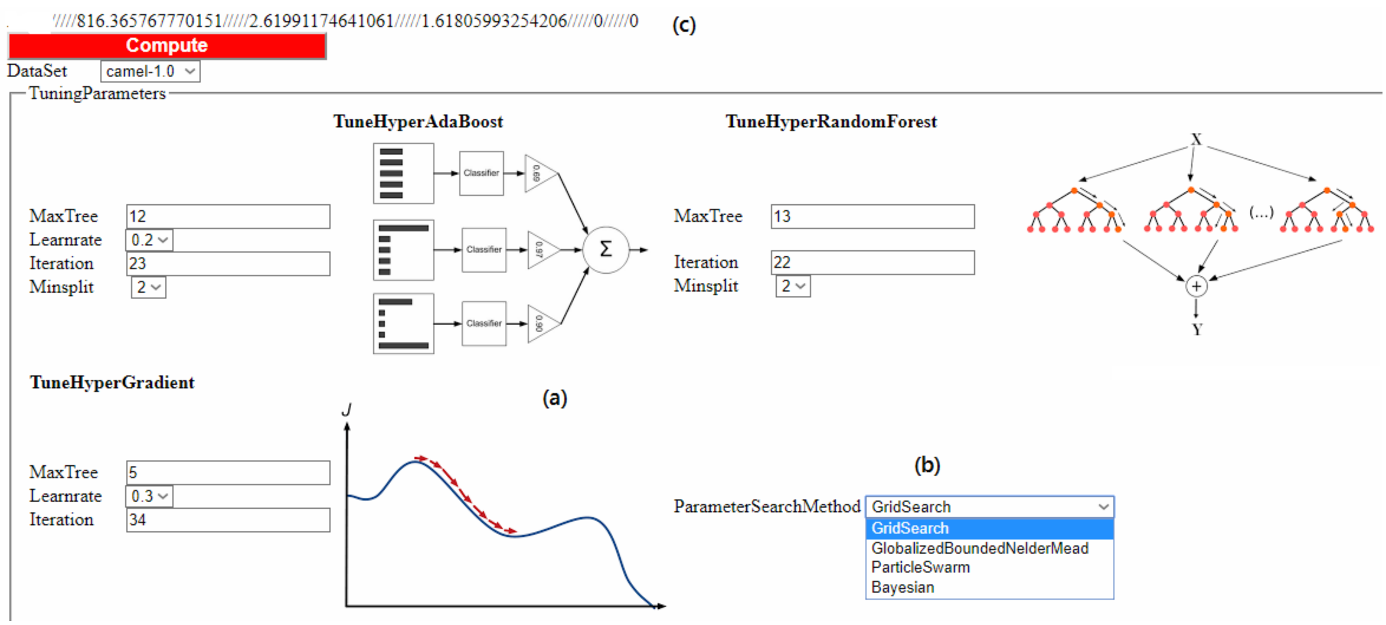

Figure 2: Main screen of the online-tool for ensemble learning.

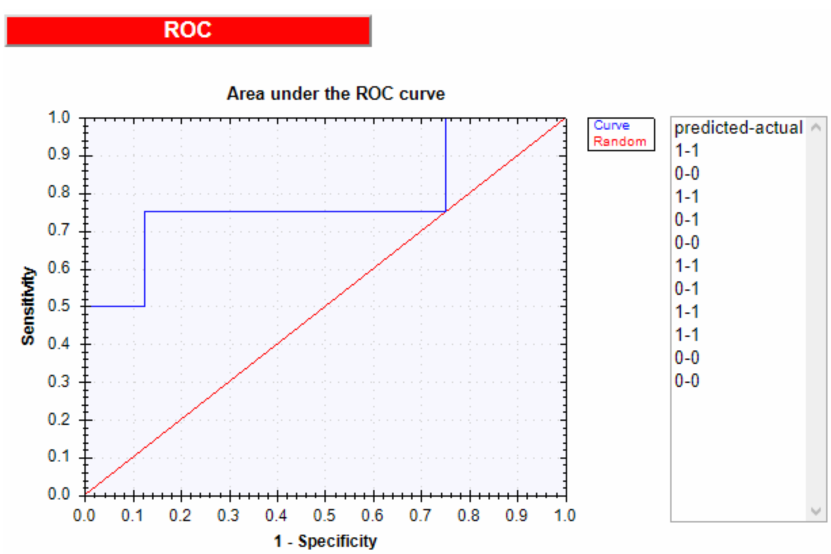

Figure 3: ROC analysis panel of the proposed tool. In this analysis, predicted and actual values of testing results can be examined.

Main screen of the proposed tool is seen in Figure 2. Figure 2 (a) includes three setting panels. In this panel, working mechanism of the algorithms is given nearby the parameter selection area. Figure 2 (b) provides four parameter search methods. Optimal values of HO are found by this section. Data sets and computation button are in Figure 2 (c). Tuning parameters proposed by the method is given and performance results are recorded in a .csv file. Testing results of error rates are given in Table 2. These results were yielded with camel-1.0 data set which is used for defect prediction experiments.

Proposed tool can perform and illustrate a ROC analysis after the tuning and learning operations are completed. An example of ROC demonstrated with the proposed tool is seen in Figure 3.

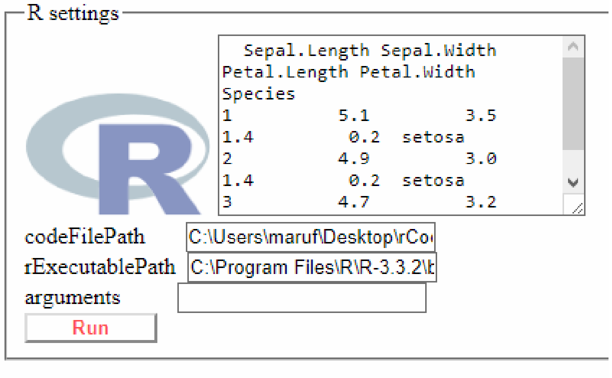

Figure 4: R code execution panel.

Figure 4 shows R execution panel of the proposed tool. In this panel, a user can determine the path of code file, R executable, and additional arguments.

Table 2: Error rates of iterations.

Tree of depth	Iterations	Learning rate	Error rate
12	20	0.028	0.142857
12	40	0.028	0.16071428
12	60	0.028	0.13616071
12	80	0.028	0.13392857
12	100	0.028	0.14955357
12	120	0.028	0.0915178
12	140	0.028	0.1450892
12	160	0.028	0.125
12	180	0.028	0.1361607

VI. THREATS TO VALIDITY

Internal Validity: The tool elaborated across the paper has some functions associated with ensemble learning and its tuning operations. Further, it provides a performance analysis panel to record learning rate. However, for its current form, proposed tool is not capable of performing other machine learning operations such as clustering, normalization, and graph modeling. It is planned to improve the tool by considering internal validity issues.

External Validitiy: To date, ensemble learning studies are focused on online-learning. It is quietly different from the design presented in this paper. Online-learning aims to update an ensemble learning method by considering new instances taken during the training process. Thus, the number of the instances is not stable in online-learning. On the other hand, the main objective of this paper is to develop a user friendly web-based tool for ensemble learning with respect to HO. Therefore, a comparison was not made due to the limited and dissimilar literature.

V. CONCLUSION

This paper proposes a novel online ensemble learning tool to tune the parameters of ensemble learning algorithms. It has been devised by considering three classifiers which construct ensemble learners. The main advantage of the tool is that it is easy to use comparing with the traditional methods based on naïve programming codes. Moreover, the tool could help researchers to understand the underlying mechanism of ensemble learners. Practitioners generally avoid conducting effort-intensive operations on software systems. User-friendly designs may alleviate this burden and encourage practitioners to use machine learning facilities. Such a design has been presented in this paper. In future works, big data focused web-based tool will be developed.

REFERENCES

[1] N. M. Nasrabadi, “Pattern recognition and machine learning”, *Journal of electronic imaging*, 16(4), 2007.

[2] C. E. Rasmussen, “Gaussian processes in machine learning”, In Advanced lectures on machine learning, pp. 63-71, Springer, Berlin, Heidelberg, 2004.

[3] Y. Lian, et al., "Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic", *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[4] Y. Bengio, “Gradient-based optimization of hyperparameters”, *Neural computation*, 12(8), pp. 1889-1900, 2000.

[5] Y. Bao, Z. Liu, “A fast grid search method in support vector regression forecasting time series”, *In International Conference on Intelligent Data Engineering and Automated Learning*, pp. 504-511, 2006.

[6] T. G. Dietterich, “Ensemble learning. The handbook of brain theory and neural networks”, vol. 2, pp. 110-125, 2002.

[7] V. Strijov, and G. W. Weber, “Nonlinear regression model generation using hyperparameter optimization”, *Computers & Mathematics with Applications*, 60(4), pp. 981-988, 2010.

[8] T. Masada, et al., "Dynamic hyperparameter optimization for bayesian topical trend analysis", Proceedings of the 18th ACM Conference on Information and knowledge management. ACM, pp. 1831-1834, 2009.

[9] X. C. Guo, et al., "A novel LS-SVMs hyper-parameter selection based on particle swarm optimization", *Neurocomputing*, vol. 71, pp. 3211-3215, 2008.

[10] C. S., Foo, C. B., Do, and A. Y. Ng, “Efficient multiple hyperparameter learning for log-linear models”, In Advances in neural information processing systems, pp. 377-384, 2008.

[11] J. S. Bergstra, R. Bardenet, R., Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization”, In Advances in neural information processing systems, pp. 2546-2554, 2011.

[12] C. Thornton, et al., "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms", Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 847-855, 2013.

[13] D. Maclaurin, D. Duvenaud, and R. Adams, “Gradient-based hyperparameter optimization through reversible learning”, In International Conference on Machine Learning, pp. 2113-2122, 2015.

[14] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, and R. Adams, R., “Scalable bayesian optimization using deep neural networks”, In International conference on machine learning, pp. 2171-2180, 2015.

[15] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A novel bandit-based approach to hyperparameter optimization”, arXiv preprint arXiv:1603.06560, 2016.

[16] T. Domhan, J. T. Springenberg, and F. Hutter, “Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves”, *IJCAI*, Vol. 15, pp. 3460-8, 2015.

[17] H. Kaneko, and K. Funatsu, K., “Fast optimization of hyperparameters for support vector regression models with highly predictive ability”, *Chemometrics and Intelligent Laboratory Systems*, vol. 142, pp. 64-69, 2015.

[18] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks", In Advances in Neural Information Processing Systems, pp. 4134-4142, 2016.

[19] L. Deng, and J. C. Platt, "Ensemble deep learning for speech recognition", In Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[20] E. Fersini, E. Messina, F. A. Pozzi, "Sentiment analysis: Bayesian ensemble learning", Decision support systems, vol. 68, pp. 26-38, 2014.

[21] I. H. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features", *Information and Software Technology*, vol. 58, pp. 388-402, 2015.

[22] J. Hu, T. Li, C. Luo, H. Fujita, and S. Li, "Incremental fuzzy probabilistic rough sets over two universes", International Journal of Approximate Reasoning, vol. 81, pp. 28-48, 2017.

[23] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring", *Journal of Retailing and Consumer Services*, vol. 27, pp. 11-23, 2015.

[24] E. I. Papageorgiou, K. D. Aggelopoulou, T. A. Gemtos, and G. D. Nanos, "Yield prediction in apples using Fuzzy Cognitive Map learning approach", *Computers and electronics in agriculture*, vol. 91, pp. 19-29, 2013.

[25] M. Pratama, E. Dimla, E. Lughofer, W. Pedrycz, and T. Tjahjowidowo, "Online Tool Condition Monitoring Based on Parsimonious Ensemble+", arXiv preprint arXiv:1711.01843, 2017.