ISSN: 2321-7782 (Online)

Volume 3, Issue 3, March 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Blend of SVM, MultiBoost, Decorate and Bagging Classifiers for Improving Accuracy and Security of Big Data

Yugandhara Rajendra Patil[1]
Computer Science and Engineering
G.H. Raisoni Institute of Engineering and Management
Jalgaon, India

Sonal Patil[2]
Computer Science and Engineering
G.H. Raisoni Institute of Engineering and Management
Jalgaon, India

Abstract: Big data achieves more and more attention from researchers in recent years because it has become ubiquitous in numerous application domains. The proposed classifier combines SVM (Support Vector Machine), MultiBoost, Decorate and Bagging classifiers with base classifiers for improving performance of classification significantly. SVM is employed for higher accuracy and it can produce powerful results in range from good to excellent. Decorate can reduce errors of regression methods. MultiBoost is used as extension of AdaBoost and it produces result with lower error rates. Bagging performs better with noise. The planned classifier is very large because it is specially tailored for handling Big data. In this classifier, ensemble classifiers are combined at each tier. Each tier will collect outputs from previous tier, analyse and combine them and send their output to the next tier. Here multitiers are used because of many tiers, work is divided into each of these tiers so that speed and accuracy increases. It is easy to set up and run. It includes different ensemble classifiers on several levels, combining strengths of their methods. This classifier is also concern for security of Big data.

Keywords: SVM,MultiBoost, AdaBoost, Bagging,Decorate.

I. INTRODUCTION

Data Mining is the technology to extract the knowledge from the pre-existing databases. It is used to explore and analyse the same. The data which is to be mined varies from a small data-set to a large data-set i.e. Big Data.Big data is so large that it does not fit in the main memory of a single machine, and the it need to process big data by efficient algorithms. Modern computing has entered the era of Big Data.

The investigation of this new construction is important, because the role of algorithms for analysis of Big Data has been growing. It also helps in improving security of Big data. The main aim of this paper is to develop the classifier as a general technique that may be useful for the analysis of Big Data in various application domains.This construction is illustrated in Figure 1.

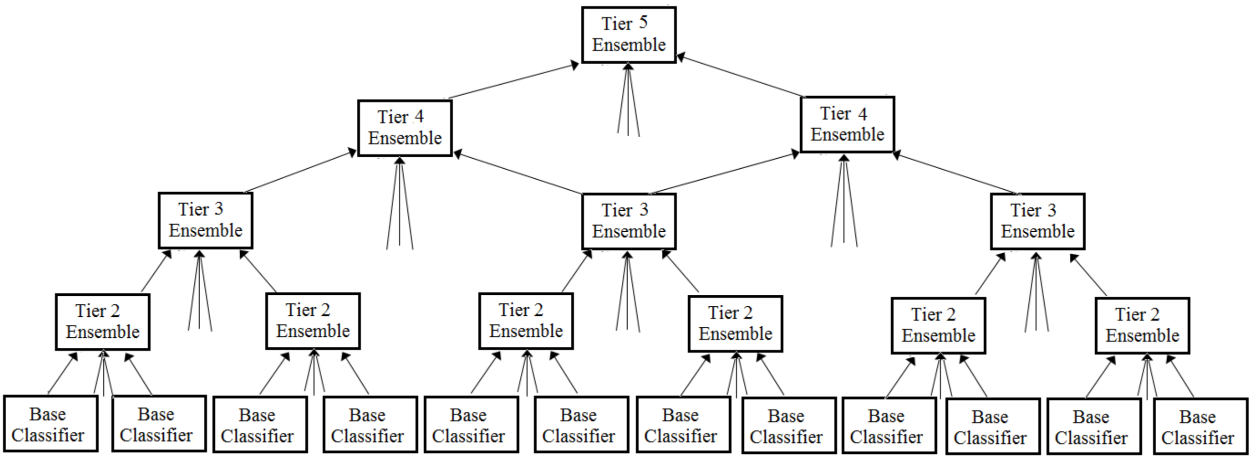

Fig 1.Five- tier classifier processing big data. The direction of arrows shows data flow.

Five-tier LIME classifiers achieved well higher performance compared with the base classifiers or standard ensemble meta classifier classifiers. This demonstrates that our new technique of combining diverse ensemble meta classifiers into one unified five-tier ensemble incorporating diverse ensemble meta classifiers as elements of different ensemble meta classifiers can be applied to enhance classifications.

This paper is organized as follows. Section II contains brief overview of previous related work. Section III describes five-tier classifier investigated in this paper. Section IV describes the base classifier which is used in this planned classifier and Section V deals with the ensamble meta classifiers used in this classifier.Lastly, conclusion is presented.

II. Related Work

Researchers in [1] proposed the four-tier Large Iterative Multitier Ensamble (LIME) classifier which is used for security of the big data. This classifier puts the idea of combining multiple classifiers at several levels.

The paper [3] investigates an iterative hierarchical key exchange scheme for secure scheduling of big data applications in cloud computing. The privacy preservation over big data on cloud is considered in [4].

The first classification method integrating static and dynamic features into a single test was presented in [5]. The approach proposed there improved on previous results using individual features collected separately. The time required for the test was reduced by half.

III. Proposed Classifier

If a dataset isn't massive enough, then the classifier can revert to using solely a base classifier or simply a small part of the complete system and can not improve the standard of the classification. This classifier will mix numerous ensemble meta classifiers into one reiterative gradable system and may be terribly massive. Each ensemble meta classifier combines a group of base classifiers into a typical arrangement. Economical multi-tier classifiers and additional general multi-classifier systems are explored, for instance, within the previous publications [6],[7].Our construction of this classifier was impressed by previous analysis within the literature, however is completely different.

Traditional ensemble meta classifiers generate their collection of base classifiers given an indication, or an example, or a model of just one base classifier as an input parameter. After the generation stage, they use the complete ensemble of the base classifiers to method process, collect their outputs and mix them to prepare the final decision.

The present article is dedicated to a new theme that makes very massive classifiers tailored for handling big data. These classifiers automate the method of generating a large multitier system. they create it simple to generate very massive classifiers combining diverse ensemble meta classifier methods at many levels. These classifiers are used with five tiers in this paper. They incorporate diverse ensemble meta classifiers into second, third, fourth and fifth tiers at the same time and combine them into one integrated iterative system so that fourth tier ensemble meta classifiers acts as an integral part of the fifth tier ensemble meta classifier, third tier ensemble meta classifiers acts as an integral a part of the fourth tier ensemble meta classifier, and every second tier ensemble meta classifier is an integral a part of its third tier ensemble meta classifier parent as shown in figure 1. The fifth tier ensemble meta classifier of this construction invokes fourth tier ensemble meta classifiers, and successively they invoke their third and second tier ensemble meta classifiers in an reiterative fashion.

It is simple to set up and generate this classifier. All fourth tier ensemble meta classifiers are generated by the fifth tier ensemble meta classifier given just one instance of a third tier ensemble as an input parameter for the generation stage. The forth tier ensemble meta classifier generates all third tier ensemble meta classifiers and executes them in exactly identical manner because it usually handles base classifiers. Similarly, every third tier ensemble meta classifier applies its method to generate and mix its second tier ensemble meta classifiers. Finally, the second tier ensemble meta classifiers generate, execute and combine their base classifiers in their standard fashion.

To start the method a designer should initialize a five-tier classifier by specifying which ensemble meta classifier can operate at the fifth tier. Then the designer provides a parameter to the fifth tier ensemble meta classifier indicating, which fourth tier ensemble meta classifier is to be used as a part of the quality generation method of the fifth tier ensemble meta classifier. After that, the designer specifies the second tier ensemble meta classifier method to be used by the third tier ensemble meta classifier, and the base classifier handled by the second tier ensemble meta classifier. The initialization step is shown in Figure 2.

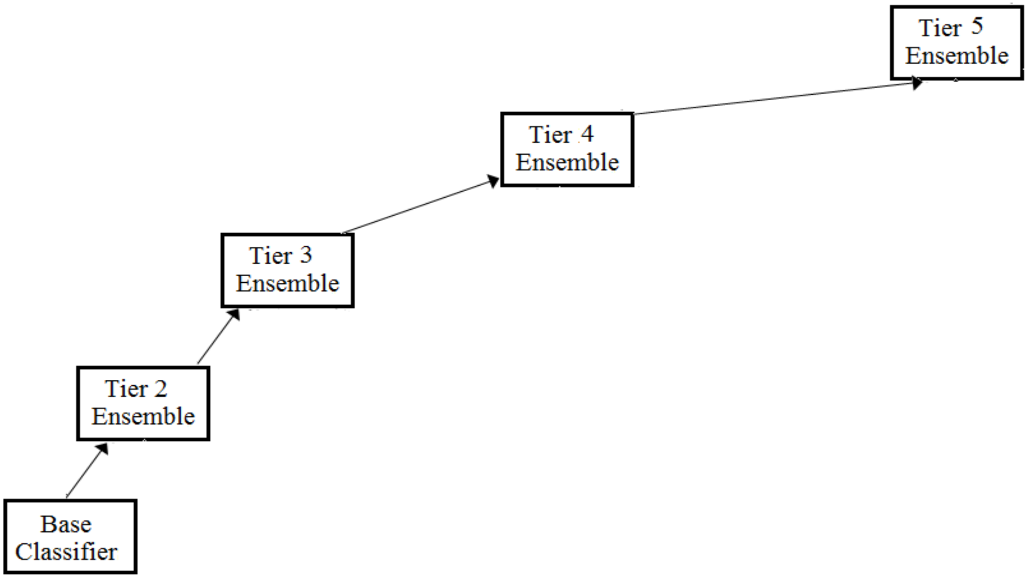

Fig. 2 Initialization of five-tier classifier.

In this paper we used various ensemble meta classifiers and base classifiers implemented within the Waikato environment for knowledge Analysis (WEKA). After choosing proper choices, the whole system is generated automatically by the SimpleCLI, using the embedded iterative and recursive capability of Java programming.

After initialization every of the ensemble meta classifiers chosen by the designer uses its own technique of generating the classifiers at the lower tier. First, the fifth level ensemble meta classifier generates a group of the classifiers at the fourth tier as shown in Figure 3. Second, every of the fourth tier ensemble meta classifiers created in Figure 3 applies its own scheme of generating third tier ensemble meta classifiers as shown in Figure 4. Third, every of the third tier ensemble meta classifiers created in Figure 4 applies its own theme of generating second tier ensemble meta classifiers as shown in Figure 5. Lastly the complete generated classifier is shown in figure 6.

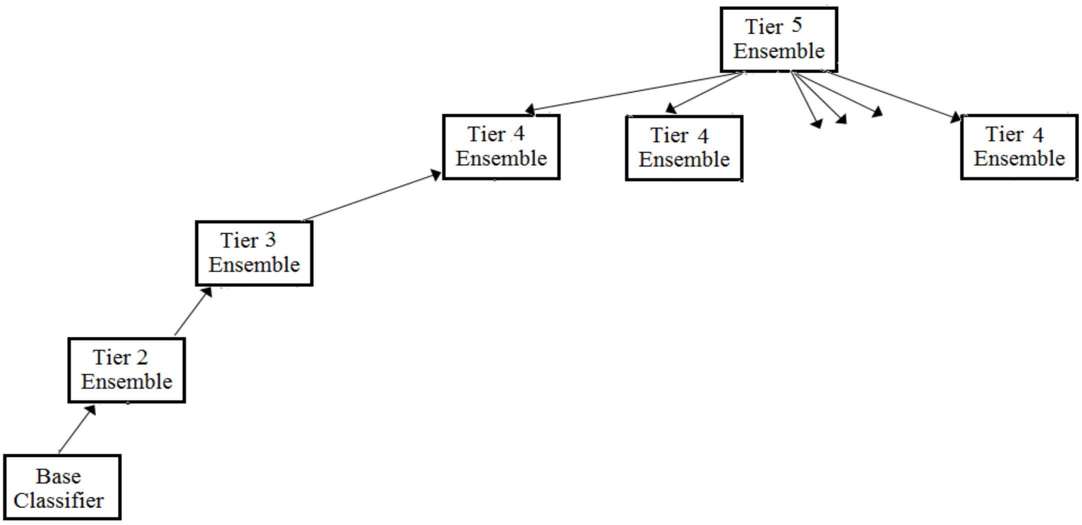

Fig. 3 Stage 1 of generating Five-tier classifier.

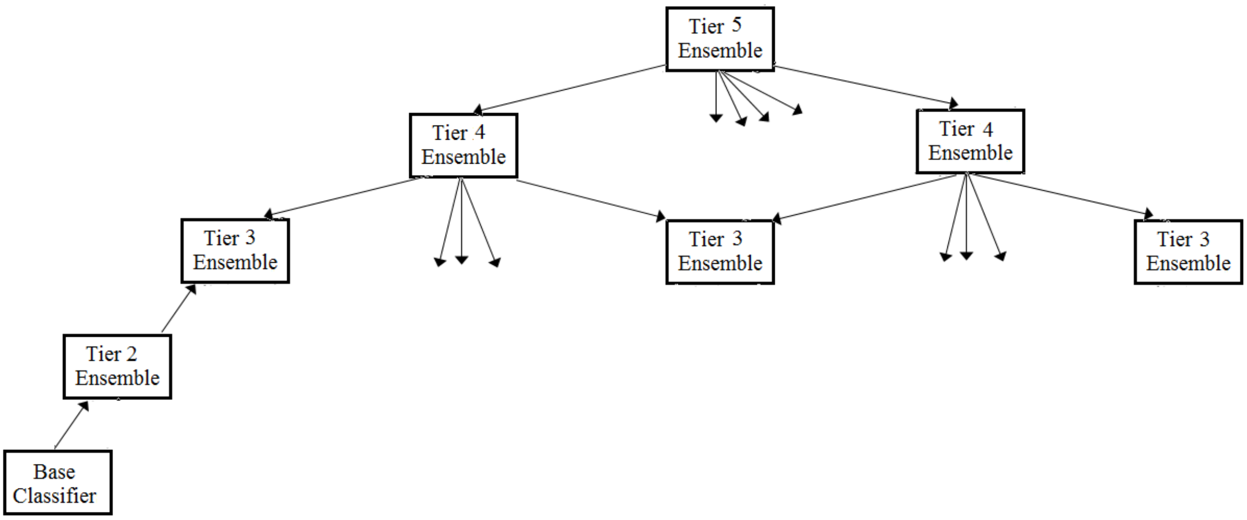

Fig. 4 Stage 2 of generating Five-tier classifier.

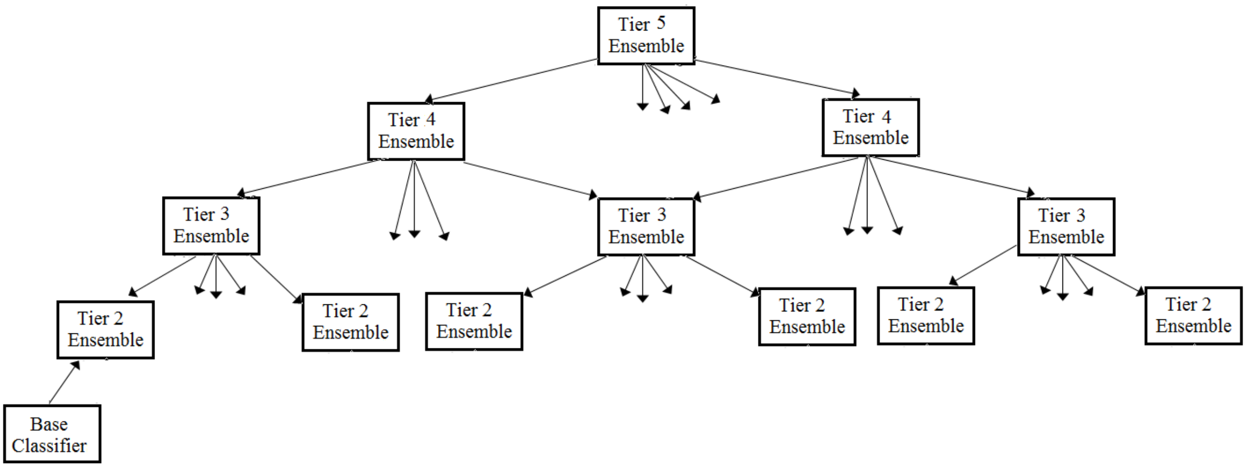

Fig 5. Stage 3 of generating Five-tier classifier.

Finally,every of the third tier ensemble meta classifiers, made at the preceding stage uses its technique of generating a set of base classifiers as per the type of the base classifier. This concludes the generation stage of work of the planned classifier.

This classifier processes knowledge as shown in Figure 1, The direction of arrows indicates knowledge flow. Edges not connected to classifiers indicate the direction of possible knowledge flow from various classifiers that aren't depicted within the diagram.

The base classifiers analyze the features of the main instances and pass on their output to the second tier ensemble meta classifiers. The second tier ensemble meta classifiers collect all outputs of the base classifiers, combine them, and send their own output to their parent third tier ensemble meta classifiers. as same the third tier ensemble meta classifiers gathered the outputs of the second tier ensemble meta classifiers analyze and combine them, and send their own output to the fourth tier ensemble meta classifier. The Fourth tier ensemble classifier collects the output from third tier meta classifier and send their own output to the fifth tier ensemble meta classifier. The fifth tier ensemble meta classifier analyses the results of the third tier ensemble meta classifiers and produces the final decision of the classifier.

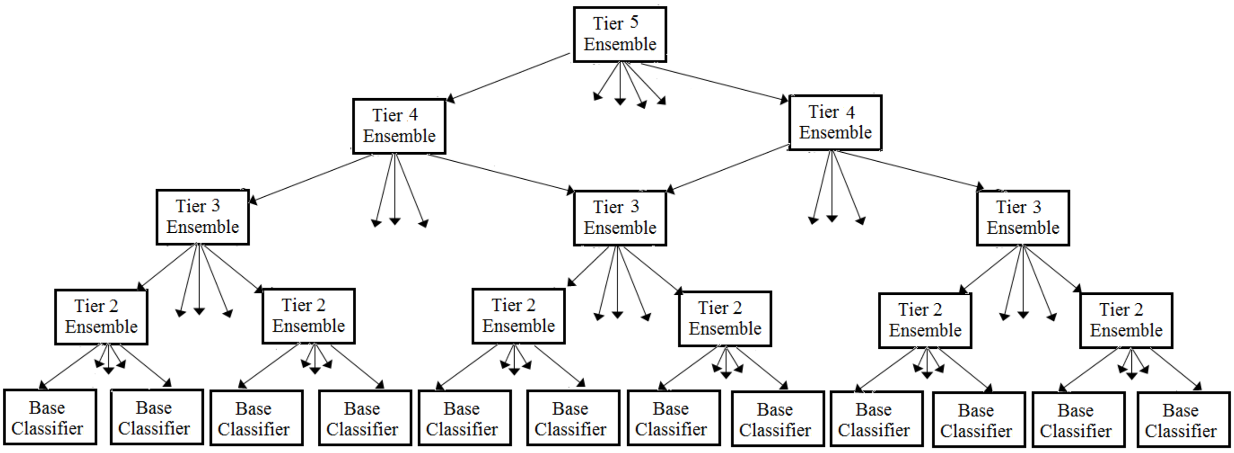

Fig 6. Stage 4 of generating Five-tier classifier.

The new addition of this article is in generating new large systems as repetitive ensembles of ensembles by linking a fifth tier ensemble meta classifier to a different fourth tier ensemble meta classifier rather than a base classifier and linking the fourth tier ensemble meta classifier to a third tier ensemble meta classifier, second tier ensemble meta classifier that successively are connected to their base classifiers. during this manner the fifth tier ensemble meta classifier will generate the full system. These classifiers are a new construction within the framework of this approach for the following two reasons.

First, classifiers embody totally different ensemble meta classifiers on many tiers. Second, they use these ensemble meta classifiers iteratively to get the whole classification system automatically.

In this automatic generation capability includes many large ensemble meta classifiers in several tiers simultaneously and auto combines them into one hierarchical unified system so that one ensemble meta classifier is an integral part of another one.

IV. Base Classifier

Following are the different base classifiers, available in WEKA. BayesNet, DTNB, FURIA, J48, Random Forest, MultilayerPerceptron.

Random Forest:

The Random Forest builds a forest of random trees by generating many decision tree predictors with randomly selected variable subsets and utilizing a different subset of training and validation data for each of these trees, as practitioning in [8].

To control the variation in creating the set of random trees, Random Forest uses the process of random selection of features proposed in [9][10]. After creating many trees, the resulting class prediction is based on votes from the trees. The variables are ranked and variables with lower rank are eliminated based on the basis of empirical performance heuristics.

V. Conclusion Ensemble Meta Classifier

We used Simple CLI command line in WEKA [11] to investigate the performance of the following ensemble meta classifier: SVM, AdaBoost, Bagging, Decorate, MultiBoost.

SVM: Support Vector Machines (SVM) recently became one of the most popular classification ways. they have been employed in a good variety of applications. Support Vector Machines may be thought of as a technique for constructing a special kind of rule, known as a linear classifier, in a way that produces classifiers with theoretical guarantees of excellent predictive performance (the quality of classification on unseen data). The theoretical foundation of this technique is given by statistical learning theory[2].

SVMs can be a great tool for insolvency analysis, within the case of non-regularity within the data, for example when the data are not frequently distributed or have an unknown distribution. SVMs give a good out-of-sample generalization. this suggests that, by selecting an appropriate generalization grade, SVMs may be robust, even when the training sample has some bias[2].

MultiBoost: MultiBoost extends the approach of the AdaBoost with the wagging technique, [12]. Wagging is a variant of bagging where the weights of training instances generated during boosting are utilized in selection of the bootstrap samples. It is explained in [12] that experiments on a huge and diverse collection of UCI data sets have demonstrated that MultiBoost achieves high accuracy significantly more often than wagging or AdaBoost.

AdaBoost: It uses several classifiers in succession. Each classifier is trained on the instances that have turned out more difficult for the preceding classifier.   this end all instances are dedicated weights, and if an instance turns out difficult to classify, then its weight increases. We used the highly successful AdaBoost classier described in [13].

Decorate: It involves constructing special artificial coaching examples to build various ensembles of classifiers. A comprehensive collection of tests have established that design systematically creates ensembles additional accurate than the base classifier, Bagging, Random Forests, that are training correct than Boosting on small training sets, and are comparable to Boosting on large training data sets, [14].

***Bagging* (bootstrap aggregating):** It generates a collection of new data sets by resampling the given training set at random and with replacement. These data sets are called bootstrap samples. Novel classifiers are then trained, one for each of these new training data sets. They are amalgamated via a majority vote, [15].

VI. Conclusion

We get new results evaluating performance of such large five-tier classifiers. especially, Random Forest performed best in this setting, which novel five-tier LIME classifiers will be used to accomplish more improvement of the classification outcomes. The five-tier classifiers supported Random Forest achieved higher performance compared with the bottom classifiers or easier ensemble meta classifiers. Five-tier classifier with SVM at fifth tier, Multiport at the fourth tier, the Decorate at the third tier and bagging at the second tier for the best result.

The experimental results show that, these five-tier classifiers will be used to improve classifications. They're effective if diverse ensemble meta classifiers are combined at totally different tiers of the classifier. They need created significant enhancements to the performance of base classifiers and standard ensemble meta classifiers.

Acknowledgment

The authors are grateful to three reviewers for thorough reports with comments and corrections that have helped to improve this article.

References

1. Jemal H. Abawajy, Andrei Kelarev, Morshed Chowdhury," Large Iterative Multitier Ensemble Classifiers for Security of Big Data",in IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, 30 October 2014.
2. Laura Auria, Rouslan A. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis", in Berlin, August 2008.
3. C. Liu et al.,"An iterative hierarchical key exchange scheme for secure scheduling of big data applications in cloud computing," in Proc. 12th IEEE Int. Conf. Trust Security Privacy Comput. Commun., Melbourne, Australia, Jul. 2013, pp. 9-16.
4. X. Zhang, C. Liu, S. Nepal, C. Yang, and J. Chen, "Privacy preservation over big data in cloud systems," in Security, Privacy and Trust in Cloud Systems. Berlin, Germany: Springer-Verlag, 2013, pp. 239-0257.
5. R. Islam, R. Tian, L. M. Batten, and S.Versteeg, "Classification of malware based on integrated static and dynamic features," J. Netw. Comput. Appl., vol. 36, no. 2, pp. 646-656, 2013.
6. R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 324-335, 2013.
7. R. Islam, J. Abawajy, and M. Warren, "Multi-tier phishing email classification with an impact of classifier rescheduling," in Proc. 10th ISPAN, 2009, pp. 789-793.

8. L. Breiman, “Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32,2001.

9. T. K. Ho, :Random decision forest," in Proc. 3rd Int. Conf. Document Anal. Recognit., 1995, pp. 278-282.

10. Y. Amit and D. Geman, “Shape quantization and recognition with randomized trees," Neural Comput., vol. 9, no. 7, pp. 1545-1588, 1997.

11. S. Rajan et al. (2013, Oct. 12). Expanded Top Ten Big Data Security and Privacy Challenges. Cloud Security Alliance, Los Angeles, CA, USA [Online]. Available: http://cloudsecurityalliance.org/research/big-data/

12. G. I. Webb, ``MultiBoosting: A technique for combining boosting and wagging," Mach. Learn., vol. 40, no. 2, pp. 159-196, 2000.

13. Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm," in Proc. 13th Int. Conf. Mach. Learn., 1996, pp. 148-156.

14. P. Melville and R. J. Mooney, `”Creating diversity in ensembles using artificial data," Inf. Fusion, vol. 6, no. 1, pp. 99-111, 2005.

15. L. Breiman, “Bagging predictors," Mach. Learn., vol. 24, no. 2, pp. 123-140, 1996.

AUTHOR(S) PROFILE

Yugandhara Patil, received degree BE in computer engineering in 2014 and now pursuing ME in computer science and Engineering from GHRIEM, Jalgaon.

Sonal Patil, received degree BE,Mtech in computer science and Engineering. She has total 68 publications, out of which 56 are international and remaining are national publications. Since one year she is working as HOD of IT department, GHRIEM, Jalgaon