International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Development of a Rule Based Classification System to Identify a Suitable Classifier for a Particular Dataset

Subham Datta*, Gautam, Tapas Saha
Department of Computer Science, Pondicherry University, Puducherry, India

ABSTRACT

Big data in the present world have a tremendous scope of research along with machine learning. Given a large amount of data, performing operations on these data is a very tedious task. One such process involves the classification of these huge amounts of the dataset. For classification of a dataset, we use different set of classifiers. The dataset, when tested on these classifiers, shows some results which are obviously different from each other. For a known dataset and the set of classifiers, we know the classifier from the set of classifiers that gives the best result. However, it might be the case that some unknown, a newly created or a modified dataset, the set of classifiers which gives the best result is a real challenge [14]. In this paper, we have applied 16 classifiers on IRIS dataset and the experimental results show GRID search classifier provides the best accuracy. So, from here on, we can conclude that the dataset which has similarity with IRIS dataset, GRID search classifier can be applied to get high accuracy as compared to other classifiers.

Keywords: Classification, Holdout Method, Rule Based Classification, GRID Search.

I. INTRODUCTION

Machine Learning [3] is the field of study that gives computers the ability to learn without being explicitly programmed. Machine learning programs detect patterns in data and adjust program actions accordingly. Machine Learning [3][28] is a scientific discipline that addresses the following question: How can we program systems to automatically learn and to improve with experience? Learning in this context is not learning by heart but recognizing complex patterns and make intelligent decisions based on data.

Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data and information. In machine learning computers don't have to be explicitly programmed [8] but can change and improve their algorithms by themselves.

Today, machine learning algorithms enable computers to communicate with humans, autonomously drive cars, write and publish sport match reports, and find terrorist suspects. Machine learning will severely impact most industries and the jobs[1] within them, which is why every manager should have at least some grasp of what machine learning is and how it is evolving.

Machine learning makes it relatively easier to develop sophisticated software systems [3][18] without much effort on the human side. Instead of spending years handcrafting features or fine tuning a system with a lot of parameters, machine learning does that quicker. It also only requires training data to learn better features or parameters needed to improve a given system. Machine learning algorithms are applicable to many real-life problems. For example, Facebook's News Feed changes according to the user's personal interactions with other users [9]. If a user frequently tags a friend in photos, writes on his wall or "likes" his links, the News Feed will show more of that friend's activity in the user's News Feed due to presumed closeness.

The organization of this document is as follows. In Section 2 (**TYPE OF LEARNING**), I'll discuss popular learning techniques. In Section 3 (**MAJOR SUPERVISED CLASSIFIERS**), presents existing supervised classifiers and some details about the classifiers. Discussed in Section 4 is all about

Validating techniques in Supervised learning. In Section 5, we have given a list of Classifiers Evaluation Metrics. In Section 6, we have given a precise problem statement. Section 7 discuss about the software, methods, and dataset used for the experiment followed by the experiment result. In Section 8, we have discussed the classifiers result and conclude that GRID search has the better accuracy for "iris" dataset. Also some of future directions of this work is also given in this section. In the Section 8, we listed all the references that are used in this paper.

II. TYPES OF LEARNING

A. Supervised Learning

Supervised learning[30] is also called classification or inductive learning in machine learning. This type of learning is analogous to human learning from past experiences to gain new knowledge in order to improve our ability to perform real-world tasks.

A supervised learning algorithm takes a known set of input data and known responses to the data (output), and *trains* a model to generate reasonable predictions[6] for the response to new data. Supervised learning splits into two broad categories: classification and regression.

1) In classification, the goal is to assign a class (or *label*) from a finite set of classes to an observation. That is, responses are categorical variables. Applications include spam filters, advertisement recommendation systems, and image and speech recognition. Predicting whether a patient will have a heart attack within a year is a classification problem, and the possible classes are true and false. Classification algorithms usually apply to nominal response values. However, some algorithms can accommodate ordinal classes
2) In regression, the goal is to predict a continuous measurement for an observation. That is, the responses variables are real numbers. Applications include forecasting stock prices, energy consumption, or disease incidence.

B. Unsupervised Learning

Unsupervised learning[30] is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is Cluster analysis. Clustering is a technique for finding similarity groups in data, called clusters.

Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast with SUPERVISED LEARNING or REINFORCEMENT LEARNING, there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output. The only things that unsupervised learning methods have to work with are the observed input patterns xi, which are often assumed to be independent samples from an underlying unknown probability distribution PI[x] and some explicit or implicit a priori information as to what is important.

The goal is to have the computer learn how to do something that we don't tell it how to do! But this kind of learning can be powerful because it assumes no pre-discovered classification of examples. In some cases, for example, our classifications may not be the best possible. One striking example is that the conventional wisdom about the game of backgammon was turned on its head when a series of computer programs (neuro-gammon and TD-gammon) that learned through unsupervised learning became stronger than the best human chess players merely by playing themselves over and over. These programs discovered some principles that surprised the backgammon experts and performed better than backgammon programs trained on pre-classified examples.

A second type of unsupervised learning is called clustering. In this type of learning, the goal is not to maximize a utility function, but simply to find similarities in the training data. The assumption is often that the clusters discovered will match reasonably well with an intuitive classification. For instance, clustering individuals based on demographics might result in a clustering of the wealthy in one group and the poor in another.

C. Reinforcement Learning

Reinforcement Learning [13] is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal. There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms. In the problem, an agent is supposed decide the best action to select based on his current state.
Reinforcement Learning allows the machine or software agent to learn its behavior based on feedback from the environment. This behavior can be learnt once and for all, or keep on adapting as time goes by. If the problem is modeled with care, some Reinforcement Learning algorithms can converge to the global optimum; this is the ideal behavior that maximizes the reward.

D. Deep Learning

Deep learning[4][28] (also known as deep structured learning, hierarchical learning or deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations.

Deep learning algorithms are based on distributed representations[5]. The underlying assumption behind distributed representations is that observed data are generated by the interactions of factors organized in layers. Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can be used to provide different amounts of abstraction.

III. MAJOR SUPERVISED CLASSIFIERS

A. Decision Tree

Decision Tree Classifier applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

B. Neural Network

A neural network[12] consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand. This is the learning phase.

C. Support Vector Machine

In machine learning, support vector machines[11] (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model[23] is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

IV. VALIDATION TECHNIQUES USED IN SUPERVISED LEARNING

A. The Holdout method

The holdout method[7] is the simplest kind of cross validation[20]. The data set is separated into two sets, called the training set and the testing set. Then predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute
Split dataset into two groups

- ✓ Training set: used to train the classifier.
- ✓ Test set: used to estimate the error rate of the trained classifier.

Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an "unfortunate" split. The limitations of the holdout can be overcome with a family of resampling methods.

B. Random Subsampling

Steps involving Random Subsampling[2] are as follows

- ✓ Random Subsampling performs K data splits of the dataset.
- ✓ Each split randomly selects a (fixed) no. examples without replacement.
- ✓ For each data split we retrain the classifier from scratch with the training.
- ✓ Examples and estimate Ei with the test examples.

The true error estimate is obtained as the average of the separate estimates Ei

$$E=\frac{1}{K}\sum_{i=0}^{k} E_i$$

This estimate is significantly better than the holdout estimate.

V. CLASSIFIER EVALUATION METHODOLOGY

A. Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, **a** confusion matrix[16][19]**,** also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix[17]). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa).

1. True Positive Rate (TPR) (hit rate)

Recall/sensitivity $=\frac{TP}{(TP+FN)}$

$=\frac{A}{A+B}$

2. True Negative Rate (TNR)

Specificity $=\frac{TN}{(TN+FP)}$

$=\frac{D}{C+D}$

3. False positive rate (FPR) (also called false alarm rate)

FalseAlarm $=\frac{FP}{(TN+FP)}$

$=\frac{C}{C+D}$ = 1-TNR

4. Precision (also positive predictive value)

Precision $=\frac{TP}{(TP+FP)}$

$=\frac{A}{A+C}$

5**. Kappa**$=\frac{accuracy-expectedAccuracy}{1-expectedAccuracy}$

VI. PROBLEM STATEMENT

Different Datasets shows different properties. Different Classifiers work on different Dataset, classify it and provides different accuracy level. There are large numbers of classifiers widely used in different fields. If we have an available dataset then it's a real challenge to know which classifier is best for classification of the given dataset instantly. So our aim is to develop a rule based classification system to identify a suitable classifier for a particular dataset. We are using two concepts i.e. different properties of dataset and different performance of the classifiers to develop this rule based classification system. A dataset can be Boolean, nominal, ordinal, continuous etc. and we also observe the data distribution properties. Based on these properties our model should be smart enough to predict a superior classifier which will outperform on the given dataset.

VII. EXPERIMENT AND RESULT

- ➢ **Classifiers used in the experiment**

LDA, NAIVEBAYES, DECISION TREE, MDA, QDA, RDA, NEURAL NETWORK, FDA, KNN, SVM, LOGISTIC REGRESSION, BAGGING CART, RANDOM FOREST, BOOSTED C5.0, ADABOOST, GRID SEARCH.

- **Validation technique used in the experiment** Holdout Method for splitting the dataset randomly into Training(train) set and Testing(test) set for k-th(k=3) iteration. 70% of whole dataset is split for Training and 30% is for Testing.
- **Experimental Programming Language:** *R*
- **Experimental Dataset:** "iris"

VIII. RESULTS AND DISCUSSION

Classifier	Iteration	Dataset: IRIS			
		class name->	setosa	versicolor	virginica
LDA	1	precision	1	0.9473684	0.9090909
		recall	1	0.9473684	0.9090909
		f1	1	0.9473684	0.9090909
		accuracy	0.9555556		
		kappa	0.9532164		
	2	precision	1	1	1
		recall	1	1	1
		f1	1	1	1
		accuracy	1		
		kappa	1		
	3	precision	1	1	0.9230769
		recall	1	0.9444444	1
		f1	1	0.9444444	0.96
		accuracy	0.9777778		
		kappa	0.9766082		
		Avg. precision	1	0.9824561	0.94405593
		Avg. recall	1	0.9639376	0.96969697
		Avg. f1	1	0.9639376	0.95636363
		Avg. accuracy	0.9777778		
		Avg. kappa	0.9766082		

Classifier	Iteration	Dataset: IRIS			
		class name->	setosa	versicolor	virginica
NaiveBayes	1	precision	1	1	0.8181818
		recall	1	0.9047619	1
		f1	1	0.95	0.9
		accuracy	0.9555556		
		kappa	0.9532164		
	2	precision	1	0.9285714	1
		recall	1	1	0.9473684
		f1	1	0.962963	0.972973
		accuracy	0.9777778		
		kappa	0.9766082		
	3	precision	1	1	0.8461538
		recall	1	0.8947368	1
		f1	1	0.9444444	0.9166667

		accuracy	0.9555556		
		kappa	0.9532164		
		Avg. precision	1	0.9761905	0.88811187
		Avg. recall	1	0.9331662	0.98245613
		Avg. f1	1	0.9524691	0.9298799
		Avg. accuracy	0.962963		
		Avg. kappa	0.961013667		

Classifier	Iteration	Dataset: IRIS class name->	setosa	versicolor	virginica
QDA	1	precision	1	0.9473684	0.9090909
		recall	1	0.9473684	0.9090909
		f1	1	0.9473684	0.9090909
		accuracy	0.9555556		
		kappa	0.9532164		
	2	precision	1	1	1
		recall	1	1	1
		f1	1	1	1
		accuracy	1		
		kappa	1		
	3	precision	1	1	0.9230769
		recall	1	0.9444444	1
		f1	1	0.9714286	0.96
		accuracy	0.9777778		
		kappa	0.9766082		
		Avg. precision	1	0.9824561	0.94405593
		Avg. recall	1	0.9639376	0.96969697
		Avg. f1	1	0.9729323	0.95636363
		Avg. accuracy	0.9777778		
		Avg. kappa	0.9766082		

Classifier	Iteration	Dataset: IRIS class name->	setosa	versicolor	virginica
Neural Network	1	precision	1	0.9473684	1
		recall	1	1	0.9166667
		f1	1	0.972973	0.9565217
		accuracy	0.9777778		
		kappa	0.9766082		
	2	precision	1	1	1
		recall	1	1	1
		f1	1	1	1
		accuracy	1		
		kappa	1		
	3	precision	1	1	0.9230769
		recall	1	0.9444444	1

		f1	1	0.9714286	0.96
		accuracy	0.9777778		
		kappa	0.9766082		
		Avg. precision	1	0.9824561	0.97435897
		Avg. recall	1	0.9814815	0.97222223
		Avg. f1	1	0.9814672	0.9721739
		Avg. accuracy	0.9851852		
		Avg. kappa	0.984405467		

Classifier	Iteration	Dataset: IRIS class name->	setosa	versicolor	virginica
KNN	1	precision	1	0.9473684	1
		recall	1	1	0.9166667
		f1	1	0.972973	0.9565217
		accuracy	0.9777778		
		kappa	0.9766082		
	2	precision	1	1	1
		recall	1	1	1
		f1	1	1	1
		accuracy	1		
		kappa	1		
	3	precision	1	1	0.9230769
		recall	1	0.9444444	1
		f1	1	0.9714286	0.96
		accuracy	0.9777778		
		kappa	0.9766082		
		Avg. precision	1	0.9824561	0.97435897
		Avg. recall	1	0.9814815	0.97222223
		Avg. f1	1	0.9814672	0.9721739
		Avg. accuracy	0.9851852		
		Avg. kappa	0.984405467		

Classifier	Iteration	Dataset: IRIS class name->	setosa	versicolor	virginica
Logistic Regression	1	Precision	0.9333333	0.9473684	0.9090909
		Recall	1	0.9473684	0.8333333
		f1	0.9655172	0.9473684	0.8695652
		Accuracy	0.9333333		
		Kappa	0.9298246		
	2	Precision	1	1	1
		Recall	1	1	1
		f1	1	1	
		Accuracy	1		
		Kappa	1		
	3	Precision	1	1	0.8461538
		Recall	1	0.8947368	1

		f1	1	0.9444444	0.9166667
		Accuracy	0.9555556		
		Kappa	0.9532164		
		Avg. precision	1	0.9824561	0.9184149
		Avg. recall	1	0.9473684	0.94444443
		Avg. f1	1	0.9639376	0.89311595
		Avg. accuracy	0.962962967		
		Avg. kappa	0.961013667		

Classifier	Iteration	Dataset: IRIS class name->	setosa	versicolor	virginica
Random Forest	1	Precision	1	0.9473684	0.8181818
		Recall	1	0.9	0.9
		f1	1	0.9230769	0.8571429
		Accuracy	0.9333333		
		Kappa	0.9298246		
	2	Precision	1	1	1
		Recall	1	1	1
		f1	1	1	1
		Accuracy	1		
		Kappa	1		
	3	Precision	1	1	0.8461538
		Recall	1	0.8947368	1
		f1	1	0.9444444	0.9166667
		Accuracy	0.9555556		
		Kappa	0.9532164		
		Avg. precision	1	0.9824561	0.88811187
		Avg. recall	1	0.9315789	0.96666667
		Avg. f1	1	0.9558404	0.9246032
		Avg. accuracy	0.962962967		
		Avg. kappa	0.961013667		

AdaBoost	1	Precision	1	0.9473684	0.9090909
		Recall	1	0.9473684	0.9090909
		f1	1	0.9473684	0.9090909
		Accuracy	0.9555556 0.9532164		
		Kappa	0.9532164		
	2	Precision	1	1	1
		Recall	1	1	1
		f1	1	1	1
		Accuracy	1		
		Kappa	1		
	3	Precision	1	1	0.8461538
		Recall	1	0.8947368	1

		f1	1	0.9444444	0.9166667
		Accuracy	0.9555556		
		Kappa	0.9532164		
		Avg. precision	1	0.9824561	0.9184149
		Avg. recall	1	0.9473684	0.96969697
		Avg. f1	1	0.9639376	0.9419192
		Avg. accuracy	0.651851867		
		Avg. kappa	0.968810933		

IX. CONCLUSION

We have run 16 classifiers on a dataset "iris" 3 times randomly choosing training set and test set. We've measured precision, recall, f-measure, accuracy, and kappa. We observe the result and conclude that GRID SEARCH classifier outperforms among all on the "iris" dataset. So based on the properties of "iris" dataset we can say that GRID SEARCH classifier is the superior classifier on those datasets whose properties are similar with "iris". So the more we observe the performance of different classifiers on different dataset, more we can accurately predict the superior classifier. Extension part of our project work can be exampled as MOVIE BUSINESS PREDICTION where based on the attributes of a Movie we can choose a superior classifier which will predict the outcome of the movie in box-office. There are more widely scopes for future work based on our rule based classification system.

X. REFERENCES

[1] A. M. Law and W. D. Kelton. 2000. Simulation Modeling and Analysis. Boston, MA, USA: McGraw Hill.

[2] A. M. Molinaro, R. Simon, and R. M. Pfeiffer. 2005. "Prediction error estimation: a comparison of resampling methods," Bioinformatics, vol. 21, no. 15, pp. 3301–3307.

[3] Briand, L. C. 2008. Novel Applications of Machine Learning in Software Testing. 2008 The Eighth International Conference on Quality Software, 3–10. https://doi.org/10.1109/QSIC.2008.29

[4] Chen, X.W. and Lin, X. 2014. Big data deep learning: challenges and perspectives. IEEE Access. 2, 514–525

[5] Chung, C.-T., Tsai, C.-Y., Liu, C.-H., & Lee, L.-S. 2017. Unsupervised Iterative Deep Learning of Speech Features and Acoustic Tokens with Applications to Spoken Term Detection, 25(10), 1914–1928. Retrieved from http://arxiv.org/abs/1707.05315

[6] Dong, W., & Zhou, M. 2017. A Supervised Learning and Control Method to Improve Particle Swarm Optimization Algorithms. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7), 1135–1148. https://doi.org/10.1109/TSMC.2016.2560128

[7] Dougherty, E. R., Brun, M., & Xu, Q. 2008. Which is better: Holdout or full-sample classifier design? Eurasip Journal on Bioinformatics and Systems Biology. https://doi.org/10.1155/2008/297945

[8] H. He and E. A. Garcia. 2009. "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284,(Sep. 2009).

[9] Hea, Tate. 2007. COGNITIVE WIRELESS NETWORKS APPLICATIONS OF MACHINE LEARNING TO COGNITIVE RADIO NETWORKS., (2007,August), 47–52.

[10] J. Karimi, H. Nobahari, and S. H. Pourtakdoust, . 2012 "A new hybrid approach for dynamic continuous optimization problems," Appl. Soft Comput., vol. 12, no. 3, pp. 1158–1167,(Mar. 2012)

[11] J.Zhu and T.Hastie. 2005 "Kernel logistic regression and the import vector machine," J. Comput. Graph. Statist., vol. 14, no. 1, pp. 185–205.

[12] Kumar, S. 2016. Kinematic Control of Redundant Manipulators using Neural Networks. Ieee Transactions on Neural Networks and Learning Systems, 28(10), 1–12. https://doi.org/10.1109/TNNLS.2016.2574363

[13] Li, J., Modares, H., Chai, T., Lewis, F. L., & Xie, L. 2017. Off-Policy Reinforcement Learning for

Synchronization in Multiagent Graphical Games. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2434–2445. https://doi.org/10.1109/TNNLS.2016.2609500

[14] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. 2012. "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 42, no. 4, pp. 463–484,(Jul. 2012.)

[15] Machine, R., Algorithms, L., Data, B., & Systems, D. V. 2017. Efficient and Rapid Machine Learning Algorithms for Big Data and Dynamic Varying Systems, 47(10), 2625–2626.

[16] Ohsaki, M., Wang, P., Matsuda, K., Katagiri, S., Watanabe, H., & Ralescu, A. 2017. Confusion-matrix-based Kernel Logistic Regression for Imbalanced Data Classification. IEEE Transactions on Knowledge and Data Engineering, 29(9), 1–1. https://doi.org/10.1109/TKDE.2017.2682249

[17] R. Salakhutdinov and A. Mnih. 2011. "Probabilistic matrix factorization," in Proc. NIPS, vol. 20, pp. 1–8.

[18] Rodrigues, C. N. M., Goncąlves, A. B., Silva, G. G., & Pistori, H. 2015. Evaluation of Machine Learning and Bag of Visual Words Techniques for Pollen Grains Classification. IEEE Latin America Transactions, 13(10), 3498–3504. https://doi.org/10.1109/TLA.9907

[19] S. Wang and X. Yao. 2013. "Relationships between diversity of classification ensembles and single-class performance measures," IEEE Trans. Knowl. Data Eng., vol. 25, no. 1, pp. 206–2019.(Jan. 2013).

[20] U. M. Braga-Neto and E. R. Dougherty. 2004. "Is cross-validation valid for small-sample microarray classification?" Bioinformatics, vol. 20, no. 3, pp. 374–380.

[21] V. Roth. 2001. "Probabilistic discriminative kernel classifiers formulti-class problems," LectureNotes Comput. Sci., vol. 2191, pp. 246–253.

[22] W.Wang, J. Xi and H. Chen. 2014."Modeling and recognizing driver behavior based on driving data: A survey," Math. Probl. Eng., vol.(Feb. 2014), Art. no. 245641.

[23] Wang, W., Xi, J., Chong, A., & Li, L. 2017. Driving Style Classification Using a Semisupervised Support Vector Machine. IEEE Transactions on Human-Machine Systems, 47(5), 650–660. https://doi.org/10.1109/THMS.2017.2736948

[24] Xu, J., Moon, K. H., & Van Der Schaar, M. 2017. A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs. IEEE Journal of Selected Topics in Signal Processing, 11(5), 1–13. https://doi.org/10.1109/JSTSP.2017.2692560

[25] Xu, X., & Hua, Q. 2017. Industrial Big Data Analysis in Smart Factory: Current Status and Research Strategies. IEEE Access, 5, 1–1 https://doi.org/10.1109/ACCESS.2017.2741105

[26] Y. Koren et al. 2009. "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30–37.

[27] Yang, Z., Zhang, T., Lu, J., Zhang, D. & Kalui, D. 2017. Optimizing area under the ROC curve via extreme learning machines. Knowledge-Based Systems, 130, 74–89. https://doi.org/10.1016/j.knosys.2017.05.013

[28] Zhang, L., Tan, J., Han, D., & Zhu, H. 2017. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. Drug Discovery Today, 0(0), 1–6. https://doi.org/10.1016/j.drudis.2017.08.010

[29] Zhen, X., Yu, M., Islam, A., Bhaduri, M., Chan, I., & Li, S. 2016. Descriptor Learning via Supervised Manifold Regularization for Multioutput Regression. IEEE Transactions on Neural Networks and Learning Systems, 28(9), 2035–2047. https://doi.org/10.1109/TNNLS.2016.2573260