Ilkogretim Online - Elementary Education Online, 2020; 19 (4): pp. 366-372
http://ilkogretim-online.org
doi:10.17051/ilkonline.2020.04.137

New Learning to Rank Method Based on Extreme Learning Machines in Information Retrieval Recommender System

Mohammed Hayder Kadhim, *Razi University, Kermanshah, Department of Computer Engineering and Information Technology, Iran, 1990mohammed.sh@gmail.com*
Farhad Mardukhi, *Razi University, Kermanshah, Department of Computer Engineering and Information Technology, Iran, mardukhi@razi.ac.ir*

Abstract. Nowadays, a large number of third-party services suppliers provide service-based applications or provide innovation value services to assist customers. The Recommender systems in this area play an active role to offer these applications to users. At the same time, a new algorithm for selecting and ranking such diverse applications is an important issue that attracted the attention of researchers. Ranking learning had been widely studied and globally used in data recovery. Global learning to rank methods is useful in this field. In this method, a general ranking was made between the queries extracted by a general ranking function. Then, with the help of the ELM The elaboration likelihood model, the features of these queries are learned. Based on the output of the learning model, a re-ranking is considered for items again. tentative outputs demonstrate that the suggest a way that can significantly develop the state of the art learn to rank way on score recuperation dataset.
Keywords: Extreme learn machines, Recommendation system, item ranking, learn to rank; local rank context.

Received: 03.10.2020	Accepted: 17.11.2020	Published: 23.12.2020

INTRODUCTION

These days, a huge number of recommendation systems suppliers provide service-oriented base applications and giving innovation valuable services to help customers. With time, a lot of intelligent objects will be connected and services will be introduced. Many customers will show value-added service in the coming time.

A content-based filter provides a recommendation with finding regularities in the items, like user profile and product description, where customers and items are represented with a set of explicit specifications. Nowadays efforts on rank-based recommender systems with [4] that particularly made use of learning to rank techniques. Learn to rank [5] use overhang and semi overhang machine learn methods to automagical construct a rank model depends on a given trained data. It has been extensively applying to the rank item in the recommender system [2].

Ideally, the collation functions will be generated for each query [6, 7], Also this may drive to un-reasonable cost with low generalizability as the amount of potential query is nearly unlimited. As an adjustment, and more practical way is to learn a local form for each quick-to-use query and use it to refer the rank result. For example, in [6] well thought-out frameworks are to represent each query with the most important feedback is the local arrangement context. Previous studies [7, 8, 9] have shown that pseudo linkway learned from the same local taxonomic context could greatly enhance the performance of many text-based retrievals model.

Common ranking solutions focus on methods of problem transformation [10]. The most popular category is voted on, and the final score is based on the majority of the tree [17]. Instead of that, DL deep learning can be constructed with deep architecture to adaptively capturing agent information with the raw vibration dataset across many nonlinear feature's mappings with approximate complexity nonlinear functions with the smallest error [16]. However, the computationally of complexity for DL deep learning in the trained phases are controversial to the arrangement of the elements. Besides, the system suffered from the drawbacks of increasing the phase with the computational load, and its architecture is much complex for fine-tuning as well. Most importantly, the system could ignore the inherent correlations ways among sub-label sets, and each binary classifier for each label may be will face the issue of class misbalancing when the exact volume for the labels is too large, and the other labels density are low.

Meanwhile, a Double probability multi-label classifier (PPMLC) is also an available method [17], Which are normally trained by Bayes' rule and perform well in indicating the probability distributions for the candidate's nomenclature for the same spotted query. However, the potential of index-based ways require agent trained samples, the decision makes the threshold fully dependent on unique validations dataset.

for solving the deficiencies for the aforementioned multi-element classifier in term for trained efficiency, Accuracy to the computational complexity, Huang Groups suggested ELM networks, which offers efficient with the standardized solution for generalizing front feeder network. Huang Fei worked. Demonstrated that the ELM with its variants tend to be having the best scalability (for regression with binary classes cases) and better generalization of the performance (to multi-layered cases) with a faster learn speeds (reach to thousand times) compare with the SVM with less vector toolbox support (LS-SVM). Considering that compound-query was a typical mode of MIMO issue, the original ELM could not sort multiple output labels $t_2^{opt\rightarrow m}$ Thus it cannot be directly applied to solving the compound query issue. Alternatively, an instance of a compound query can be treated as a stand-alone filter in a set of all business conditions rather than a set of labels for a single query. Because the specified styles perform roles that cannot be distinguished as a single query, all instances of the compound query must be registered [17].
In this paper, a new way to the ranking learning issue is introduced using an Extreme learning machine (ELM). Experimental outputs showing that the proposed method can significantly enhance the state-of-the-art learn to rank method on benchmarks retrievals dataset.

The remaining section is organized as follows: first, the proposed model is exposed, after that, the experiment is detailed. Finally, the conclusion and the future work looking for implementing.

The overall framework of the proposed ranking method

A summary of the proposed method is presented theoretically. In a general fashion, the service-oriented system is the relation between users, objects, and services that can be modeled as a tripartite graph with hyper-edges between them. The ternary graph is a graph whose vertices are divided into three separate groups: U = {U1, U2,..,Um} denotes a group of users m, O = {O1, O2,.., On} denotes a group of The N objects, and S = { S1, S2,., Sk } denote a group of K services [15]. We define Y as a triple relationship among these three components that represent users ' subscriptions to services depend on the things they own. Y is defined in Eq.1.

$$Y \subseteq \{u, o, s : u \in U, o \in O, s \in S\} \quad (1)$$

Thus, a redundant system can be defined as a group that describes the users U, services S, and O objects and the triangular relationship between them. The group is given in the equation. 2.

$$I = (U;\ S;\ O;\ Y) \quad (2)$$

In what follows, the issue of how to adapt learn frameworks to the rank is formalized with a context of local rank from the most recalled elements of the recommendation system. gives an exact queries q, the vector x (Q, D) must be extracted and use like a features representation of the d element. Traditional classification learning algorithms assume an ideal global classification function that takes x (Q, D) like the inputs and produces a rate scored for the element. The same way to finding this optimum is to reduce the loss function of L that is known as

$$\mathcal{L} = \sum_{q\in Q} \ell\left(\{y(q,d), f\ (x(q,d)) | d \in D\}\right) \quad (3)$$

Where is the Q is the group for all possible queries, D is the groups of candidate's items, ℓ it is the local loss compute by the items scores $f\ (x(q,d))$ and all of the corresponding's relevance judgments $y(q,d)$. suppose that we could be capturing the locally rank context of q with locally context models $I\ (R_q, X_q)$ where $R_q = \{d\ sorted\ by\ f\ (x(q,d))\}$ and $X_q = \{x(q,d)\ | d \in R_q\}$, that means the loss for learning to rank with local context could also be formulated like [22]:

$$\mathcal{L} = \sum_{q\in} \ell\left(\{y(q,d), \phi\ (x(q,d), I\ (R_q, X_q)) | d \in D\}\right) \quad (4)$$

Where the ϕ is a score function that ranks the items depend on both these features with the local context models $I\ (R_q, X_q)$. The target is to find the optimal value for I and ϕ could minimize these losing functions

$\mathcal{L}$. That will make effective use of the local contextual ranking, design a menu context template I You must fulfill all requirements. First, it must be able to directly manipulate numerical features. the classification learns systems convert rank signals, whether discrete or continuously, into a vector of records. If the context menu forms I These indices cannot be dealt with directly, we must extract the raw dataset from the elements and using manual enhance experimental models to modeling the local classification context, which is much difficult and fully ineffective. 2nd, that should take into account the positioning effecting of the most important recovered item. The value of the item in the highest score is not selfsame and the positions ranked with the rating function are the strongest factor of their importance. outwardly explicitly modeling the location effect, we will be losing the classification data and damage the generalize abilities of the entire systems. The general idea of our template is to encode the most important feedback for each query in a tutorial. Then cultivate the ordered list depend on the encrypted local context model. We aimed to our template like a deep menu context templates (DLCM). DLCM the document arrangement pipelines including three steps: 1- The 1st step is an initial retrieval using a standardized collation learn algorithm, 2- at this level, every query (q, d) it becomes to a features vector $x_{(q,d)}$ and a rank list R_q^N with size, N is created for query Q depend on a globally rank functions f. At the 2nd level is an encryption process that using a model to encoding the features of vectors X_q^N of top of the retrieved item. It takes all elements one after one from the lowest location to be the highest one, and it produces a latent element vector s_n to show the encoded local context method $I\ (R_q^N, X_q^N)$. The third step is the rearrangement process wherein the higher-order items are rearranged using the locally rank functions ϕ depend on both s_n and the invisible results o of this Model.

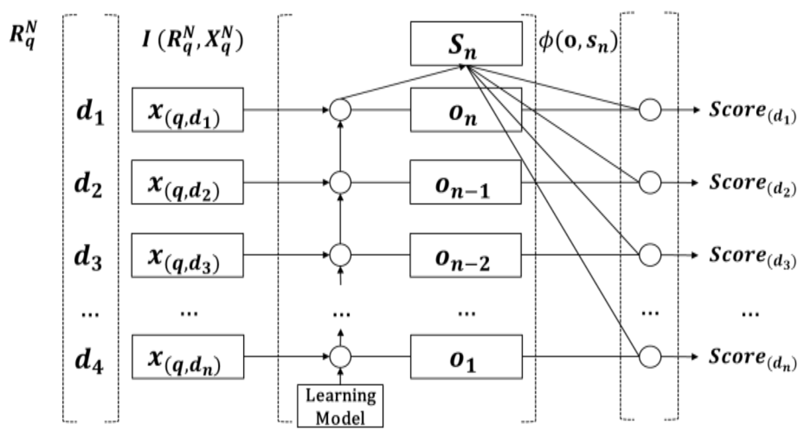

Figure 1. *structures of the learning to rank algorithm.*

All efforts to improve the ranking in the proposed system are intended to increase the predictive power of the system in providing service to the user. The prediction is to extract the correct items from the set of items in the response to the query sent to the Recommender Systems.

Multi-output-node ELM based Local Context Model

In this paper, an ELM model is using to encode basic features $X_q^N = x_{(q,d_i)} | d_i \in R_q^N$ extracted from the unknown label query. Given an unlabeled data $X_q^N = \{x_i\}_{i=1}^N$, where N defined as the number of trained patterns. The goal of the main ELM is to find the original data infrastructure. In this paper, the supervising learns of ELM meet the next assumption:

1. The unclassified data X It is derived from the exact peripheral distribution PX;
2. Probabilities of the condition $P(y|x_1)$ and $P(y|x_2)$ It's would be applied to assess the relationship between the 1st and 2nd names x_1 and x_2. If x_1 is much near to x_2, Then these two possibilities should also be similar.

To impose the above assumption from the same data point of the view, the many regulatory frameworks are limited by reducing the following loss functions:

$$L_m = \frac{1}{2}\sum_{i,j} \omega_{i,j} \, ||P(y|x_i) - P(y|x_j)||^2 \tag{5}$$

where $\omega_{i,j}$ It refers to the marital similarity among two designations x_i and x_j. It would be renowned that is the likeness vector $W = \{\omega_{i,j}\}_{i=1}^{N}$ It is often sparse, which is means we can assign a non-zero weight between x_i and x_j. And the Gaussian function $exp(-\left|\left|x_i - x_j\right|\right|^2 /2\sigma^2)$ It is applied to calculate the non-zero weight. Just to calculate the odds $P(y\,|x)$ in Eq. (5), the costing function L_m shows as,

$$\hat{L}_m = \frac{1}{2}\sum_{i,j} \omega_{i,j} \, ||\hat{y}_i - \hat{y}_j||^2 \tag{6}$$

where $\hat{y}_i$ and $\hat{y}_j$ denote the predictions of x_1 and x_2, Respectively. EQ. (6) It is using the simplified in above the expressions in the following form.

$$\hat{L}_m = T_r(\hat{Y}^T L \hat{Y}) \tag{7}$$

where $T_r(*)$ it is the tracing for the matrix, and $L = D - W$ Shows the Laplacian diagram. D is exact diagonal matrixes with the same diagonal vector $D_{ii} = \sum_{j=1}^{l+u} \omega_{i,j}$.

According to previous knowledge, we could dispose of L using $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ immediately. For unclassified train data, then the unsupervised ELM It aims to solve the following of optimization issue:

$$\min_{\boldsymbol{\beta} \in \boldsymbol{R}^{\boldsymbol{n_h} \times \boldsymbol{n_0}}} ||\boldsymbol{\beta}||^2 + \boldsymbol{\lambda} T_r(\beta^T H^T \mathrm{L} H \beta) \tag{8}$$

where $(H\beta)^T H\beta = I_{n0}$.

The perfect solution for Eq. (8) takes β As a matrix, with columns preserved β Do the eigenvectors (normal to meet the constraints) correspond to the first n0 smallest eigenvalues of these commons eigenvalue:

$$\left(\boldsymbol{I_{n_h}} + \boldsymbol{\lambda} H^T L H\right) v = \gamma H^T H v \tag{9}$$

The Laplacian Eigen mapping algorithm needs that the 1st eigenvector be eliminated because it is ever a fixed vector proportional to it. Thus, in the proposed agent ELM train model, the 1st eigenvector is outcast. Let $\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, , \boldsymbol{\gamma_{n0+1}}$ be the $(n0 + 1)$ small eigenvalues of Eq (9), and keeping $\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, \ldots \leq \boldsymbol{\gamma_{n0+1}}$. Let the vectors $\boldsymbol{v_1}, \boldsymbol{v_2}, , \boldsymbol{v_{n0+1}}$ to the correspondent eigenvectors. The out matrix β It could be represented by the series of normal eigenvectors.

$$\boldsymbol{\beta}^* = \{\boldsymbol{\hat{v}_2}, \boldsymbol{\hat{v}_3}, \ldots, \boldsymbol{\hat{v}_{n0+1}}\} \tag{10}$$

where $\boldsymbol{\hat{v}_1} = \frac{\boldsymbol{v_i}}{\left|\left|\boldsymbol{H_{v_i}}\right|\right|}, (\boldsymbol{i} = \boldsymbol{2}, \boldsymbol{3}, \ldots, \boldsymbol{n0} + \boldsymbol{1})$ Are these natural eigenvectors? When the input dimensions are smaller than the invisible nodes, Eq. (9) could be expressed with the following.

$$(\boldsymbol{I_u} + \boldsymbol{\lambda} L H H^T) u = \gamma H^T H u \tag{11}$$

where $\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_{n0+1}}$ They are the eigenvectors generalized to (n0 + 1) the smallest eigenvalues in the equation. (11). Then the resulting weight β It can be given by

$$\boldsymbol{\beta}^* = \boldsymbol{H^T}\{\boldsymbol{\hat{u}_1}, \boldsymbol{\hat{u}_2}, \ldots, \boldsymbol{\hat{u}_{n0+1}}\} \tag{12}$$

Where a renewed eigenvector normalizes $\boldsymbol{u_i}$ equals to $\frac{\boldsymbol{u_i}}{\left|\left|\boldsymbol{H}\boldsymbol{H^T}\boldsymbol{u_i}\right|\right|}, (\boldsymbol{i} = \boldsymbol{2}, \boldsymbol{3}, \ldots, \boldsymbol{n0} + \boldsymbol{1})$. Then The resulting expression is the same original ELM in [18].

$$\boldsymbol{O} = \boldsymbol{H}\boldsymbol{\beta}^* \tag{13}$$

where the matrix $\boldsymbol{O} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, \ldots, \boldsymbol{o_n}\}$ denotes the input of the Re-ranking function ϕ in Figure 1.

Re-rank by Local Context Model

The last level of the proposed arrangement to create a fresh order list by sorting the documents by the locale φ rank function. When predicting the outcome of the arrangement, the ϕ function considers both hidden ELM outputs and hidden views encoded from the local ranking field. Let's o_{n+1-i} show the output of the item $d_i \in R_q^N$, we locate a locally rank function ϕ as:

$$\phi(o_{n+1-i}\,, s_n) = \alpha 1.\, \phi(o_{n+1-i}\,.\, tanh(\alpha 2.\, s_n + b_\emptyset)) \tag{14}$$

where $1 \in R^{a \times k \times a}$,$b_\emptyset \in R^{k \times a}$, $\alpha 2 \in R^k$ also, k is a hyperparameter that controls the number of invisible units.

RESULTS

In this section, the proposed method is evaluated using four large datasets called Movie Lens UCI machine learning dataset. This database (ml - 20m) describes a 5-star rate and the free text tag activity formatting Movie Lens, a movie recommender service. Having 20,000,263 rates and 465564 tagging apps more than 27,278 films. This data was generated by 138,493 customers between January 09, 1995, and March 31, 2015.

This database was done on October 17, 2016. customers have been chosen at random for inclusion. All selected customers rated at least 20 movies. Demographic information is not included. Every user is represented by an identifier, and no other information is provided. Data is included in six files, genome-scores.csv, genome-tags.csv, links.csv, movies.csv, rating.csv, and tags.csv.

A recommender system has been implemented to introduce the necessary data in the Movie Lens dataset using the proposed algorithm. All implementations and evaluations were performed in MATLAB R2018b run on a personal computer with Microsoft Windows 10 OS, Intel-i7 processor, and 16GB memory.

Performance evaluation of ELM prediction and comparison with other work

Since the data obtained from the dataset data has a higher dimension (for example, 270 causes more invisible layers nodes in ELMs, and lead to a high value of computational load in the learning phases), this paper presents a method to extract the existing features that have been developed At [23]: PSO-VMD to reduce the dimensions of the new extract feature.

In the multi-tag context model stage, a second ELM network with multiple outputs for target prediction is applied t_i directly. It is worth noting with the applications of the kernel $K(u; v)$ In the output computation stage, there is not a big need to set L dimensions for the feature area (number of invisible nodes). The critical parameter is the number of invisible N neurons in the ELM that needs to be fully optimized beforehand. This paper first describes the experimental outs for the four scales in the limits of invisible nodes and explores the relationship between invisible layer nodes and measure groups.

Figure 2 shows these curve lines of four scales in the cases of another invisible layer node. The amount number of invisible layers node increases from 20 to 400 at interval 20. As presented in Fig 2. (a), the maximum single error is 0.576 and the minimum is 0.02 when the numbers of invisible layers node are 100. (b) indicates coverage changes Which decrease with increasing nodes with the hidden layer, and 305 becoming stable at point N = 100.

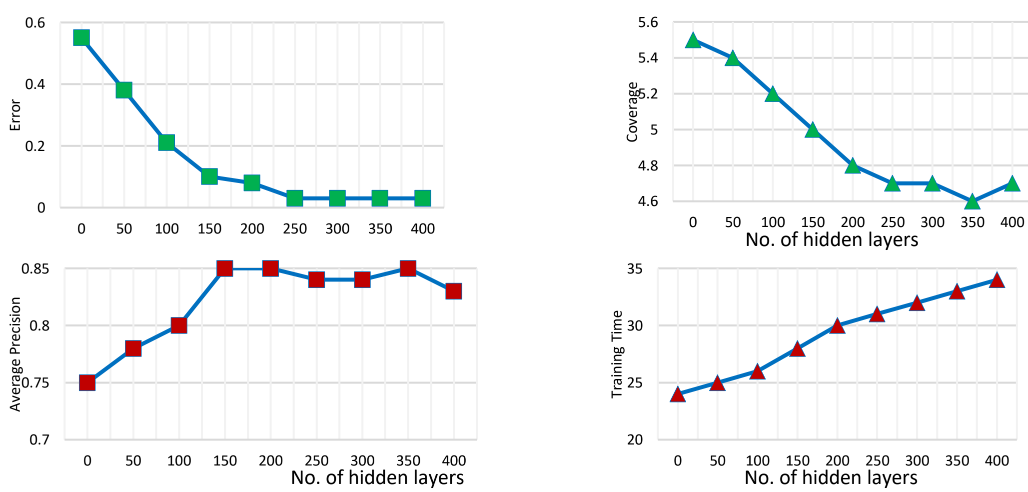

Figure 2: *Four-scale curves with a different number of hidden layer nodes.*

As shown in Figure 2. (c), The average accuracy increases as the invisible node increase, but the values are nearly constant when the number of invisible layers node is 100. due to the experimental outputs of the train time as showing in Figure 2. (d), Training time continues to increase. This paper specifies the numbers of nodes of the invisible layer n = 100 as the optimizer parameter.

Table 1 showing the evolution of total RMSE as a function of the number of queries in averaging and ranking methods. In this table, a complete simulation has been performed for each category of videos in the MovieLens dataset. The accuracy of the item items is matched by the ranking algorithm with the corresponding class.

As shown in this Table There is a reduction in the total RMSE for each method in the transition period, due to an increase in the number of Category 3, who manufacture low-quality benevolence. However, the rank model is more robust to reduce the effects of these nasty contributions and still achieves a good level of RMSE. This is because of the correct modification of features to learn of queries.

Table 1: *Evolution of RMSE with Proposed Ranking Model.*

#Queries						
Recommended Category	50	100	150	200	250	300
Action	0.29951	0.29991	0.31991	0.32191	0.33191	0.32191
Adventure	0.22833	0.22843	0.24843	0.24943	0.25043	0.25243
Animation	0.19603	0.19803	0.19903	0.21003	0.21103	0.32103
Children's	0.18252	0.18272	0.20272	0.22272	0.22372	0.23372
Comedy	0.17367	0.19053	0.209053	0.219053	0.219153	0.220153
Crime	0.15789	0.16216	0.18216	0.19316	0.19216	0.12016
Documentary	0.29421	0.29991	0.31991	0.32191	0.33191	0.32591
Drama	0.22346	0.24343	0.22843	0.24943	0.22043	0.26243
Fantasy	0.19764	0.16703	0.21903	0.22003	0.20003	0.39103
Film-Noir	0.18456	0.18972	0.20252	0.24372	0.22172	0.23472
Horror	0.17321	0.19763	0.20763	0.21873	0.203153	0.25053
Musical	0.29789	0.30991	0.32591	0.32191	0.32191	0.22191
Mystery	0.22456	0.21843	0.22543	0.26743	0.24043	0.25653

As could be seen from the same table, there is a slight reduction in the total RMSE after all the transition period. The reasons are that when the element moves to Tier 3, they start to make low-quality contributions. By returning to Tier 1, they recap making high-quality contributions. But since the RMSE is still low, the value obtained is lower than before, but is greater than the nullification threshold. The proposed method is then compared with other work in the field of learning to rank detection. This section compares the evaluation of the proposed method for all data classes.

CONCLUSIONS

Extreme learn machine (ELM) is utilized to produce a new approach to the ranking learning. In this method, a general ranking was made between the queries extracted by a general ranking function. Then, with the help of the ELM model, the features of these queries are learned. Based on the output of the learning model, a re-ranking was considered for items again. Empirical results showing that the proposed method can significantly enhance the latest learning techniques for classification methods on retrieval of criteria dataset.

REFERENCES

L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, 54: pp. 2787-2805(2010).

Telus. (SEP 2015). IoT Marketplace. Available: https://iot.telus.com/

bluerover. (2015, OCT 2015). bluerover. Available:

J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender Systems.

Felfernig, A., Polat-Erdeniz, S., Azzoni, P., Jeran, M., Akcay, A., and Doukas, C. Towards configuration technologies for IoT gateways. In: International Workshop on Configuration 2016 (ConfWS'16), pp. 73-76. Toulouse, France (2016).

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. Grouplens: applying Collaborative Filtering to Usenet news full text. Comm. of the ACM 40(3): 77-87 (1997).

Pazzani, M. and Billsus, D. Learning and revising user profiles: The identification of interesting web sites. Machine Learning 27: 313-331 (1997).

Felfernig, A. and Burke, R. Constraint-based recommender systems: Technologies and research issues. In: ACM International Conference on Electronic Commerce (ICEC08), pp. 17-26. Innsbruck, Austria (2008).

J. Wu, Y. Su, Y. Cheng, X. Shao, C. Deng, C. Liu, Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network-based fuzzy inference system, Applied Soft Computing 68 (3) 13 - 23(2018).

Z. Yang, X. Wang, P. K. Wong, Single, and simultaneous fault diagnosis with application to a multistage gearbox: A versatile dual-elm network approach, IEEE Transactions on Industrial Informatics 14 (12): pp.5245–5255(2018).

A. Elisseeff, J. Weston, A kernel method for multi-labeled classification, in Advances in Neural Information Processing Systems, pp. 681–687(2002).

R. Ziani, A. Felkaoui, R. Zegadi, Bearing fault diagnosis using multiclass support vector machines with binary particle swarm optimization and regularized fishers 405 criteria, Journal of Intelligent Manufacturing 28 (2): pp. 405–417(2017).

C. Vens, J. Struyf, L. Schietgat, S. Dˇzeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, Machine Learning 73 (2): pp.185(2008).

T. Han, D. Jiang, Rolling bearing fault diagnosis method based on VMD-ar model and random forest classifier, Shock and Vibration 2016.

M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering 26 (8): pp.1819–1837(2014).

P. K. Wong, J. Zhong, Z. Yang, C. M. Vong, Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis, Neurocomputing 174: pp.331–343 (2016).

Shubham Sharma & Ahmed J. Obaid (2020) Mathematical modelling, analysis and design of fuzzy logic controller for the control of ventilation systems using MATLAB fuzzy logic toolbox, Journal of Interdisciplinary Mathematics, 23:4, 843-849, DOI: 10.1080/09720502.2020.1727611.

Ahmed J. Obaid, Shubham Sharma, Recent Trends and Development of Heuristic Artificial Intelligence Approach in Mechanical System and Engineering Product Design, Saudi J. Eng. Technol. (SJEAT), Vo. 5, No.2, pp. 86-93, 2020.