Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 26 (2023), No. 3, pp. 519–530
DOI : 10.47974/JIM-1678

A novel approach of unsupervised feature selection using iterative shrinking and expansion algorithm

V. Dankan Gowda *
Department of Electronics and Communication Engineering
BMS Institute of Technology and Management
Bangalore
Karnataka
India

Avinash Sharma †
Department of Computer Science & Engineering
M. M. Engineering College
Maharishi Markandeshwar (Deemed To Be) University
Mullana
Ambala
India

S. Kumaraswamy §
Department of Computer Science and Engineering
Global Academy of Technology
Bengaluru
Karanataka
India

Parismita Sarma ‡
Department of Information Technology
Gauhati University
Guwahati
Assam
India

* *E-mail:* dankan.v@bmsit.in (Corresponding Author)
† *E-mail:* asharma@mmumullana.org
§ *E-mail:* skswamy@gat.ac.in
‡ *E-mail:* parismita.sarma@gmail.com

© Taru Publications

Naziya Hussain [@]
School of Computers
IPS Academy
Indore
Madhya Pradesh
India

Santosh Kumar Dixit [#]
Amity School of Engineering and Technology (ASET)
Amity University
Patna
Bihar
India

Anand Kumar Gupta [$]
Department of Information Technology
BlueCrest University
Monrovia
Liberia

Abstract

An major constraint in the realm of feature selection is that users choose the ideal number of characteristics that must be picked. In this article, an effort is made to automate the process of determining a suitable value for the appropriate the quantity of characteristics that must be chosen for better recognition tasks. To estimate the ideal amount of features that should be maintained for properly describing the data, we use the dense subgraph discovery approach for this goal. Notably, the existing methods uses a similar kind of approach called the dense subgraph finding. But the earlier approach obtains a single dense subgraph, while the task of dense subgraphs finding obtains a number of dense subgraphs that are important for learning the internal structure of any network. Thus, dense subgraphs finding is likely to be adopted as a natural choice for realizing the complex relations among the nodes of massive real-life networks, such as biological networks, social networks.

Subject Classification: *Primary 93A30, Secondary 49K15, 65Y10, 65F08 and 93B47.*

Keywords: *Shrinking algorithm, Expansion algorithm, Subgragh, Pattern recognition, Emotion recognition and machine learning.*

[@] *E-mail:* **naziyahussain@gmail.com**
[#] *E-mail:* **skdixit@ptn.amity.edu**
[$] *E-mail:* **Ganand40@yahoo.co.in**

1. Introduction

Recognizing data patterns is a fundamental human task since ages. People perform several types of tasks related to pattern recognition on numerous occasions to facilitate decision-making in their daily lives [1]. At the basic level, humans accept signals through various sensory organs. The brain then processes and summarizes these data, before storing the data in the neural network in an encoded form. Using this data representation, the brain is subsequently able to perform many tasks, such as extracting meaningful information, making predictions, observing trends, grouping similar objects, and so on. That is, humans are able to grasp some characteristic features of the data patterns in order to perform different types of recognition activities. These pattern recognition tasks can be completed subconsciously or consciously. For example, when a person sits in front of a television, he may choose his favorite channel based upon his domain of interest. Similarly, when a person enters a restaurant, she can recognize foods that she likes on the menu and will order the meal that appeals to her most at that point in time. Such pattern recognition tasks are performed routinely by human beings.

However, making a machine recognize both patterns and a person's mental state is a significant challenge. There are other problems in pattern recognition that have proven difficult to study. For example, disease prediction [2], handwritten character recognition [3], fraud detection [4], emotion recognition, chess playing system formation [5], robot construction, and image processing are challenges for both humans and machines. Scientists have worked on this issue for decades; much progress has been made. Deep Blue, developed by IBM, has been one of the most widely used chess-playing computers, featuring massive computational power [6]. The dimensionality of a datum is generally reduced for three reasons: to reduce the training time needed In an effort to accelerate the procedure of knowledge discovery; to visualize insights into the underlying data for selecting an appropriate learning model; and t in order to enhance the efficiency of the relevant learning model, a better classification model must be constructed [7, 8]. Unsupervised feature selection, on the other hand, adopts certain criteria without the involvement of the class labels while choosing a minimal subset of original features[9]. Many supervised feature selection algorithms have already been devised. Thus, producing a good feature subset in the unsupervised framework continues to attract a great deal of attention.

2. Related work

This section discusses several fundamental concepts about the machine recognition of patterns. In today's era of large datasets, both the pattern recognition and machine learning [10] algorithms are complementary to each other for discovering valid, novel and potentially useful knowledge from the databases. In general, these massive sized databases are composed of several pictorial and other non-numeric information acquired through the interaction among science, engineering, technology and society [11]. There exist several

valuable motivations behind the progress of pattern recognition research. One of the main reasons may be seen as the need for humans to communicate normally with the computer technology. Importantly, the fact that scientists are worried about inventing and constructing smart robots to do specific recognizing tasks that we sometimes accomplish as humans is also a source of inspiration. Moreover, the most interesting phenomenon observed so far is the idea of discovering the next-generation computing systems [12]. In general, a pattern recognition system is a three-tier process that encompasses (i) data acquisition, (ii) dimensionality reduction, and (iii) pattern recognition via classification or clustering [13]. Pattern recognition techniques can be extensively applied in numerous fields of research, where the data are assumed to be qualitative, quantitative or both. The data perhaps even be linguistic, numerical, pictographic, or any combination of them [14]. In order to build a robust recognition system corresponding to a particular problem, the related data are first accumulated through a collection of intellectual sensors across the environment. The collection of all these data gathered in so style forms the data acquisition phase. Let an object data consists of a number of feature vectors. Then the object data can be represented as $D^{obj} = \{{D_1}^{obj}, {D_2}^{obj}, {D_3}^{obj} \dots\dots\dots {D_4}^{obj}\}$, where ${D_r}^{obj}$ is the numerical representation of the r^{th} $(r = 1,2,3, \dots\dots, n)$object. Each object ${D_r}^{obj}$, in turn, can also be considered as a d-dimensional feature vector, often represented as ${D_r}^{obj} = {D_{r,1}}^{obj}, {D_{r,2}}^{obj}, {D_{r,3}}^{obj}, \dots\dots {D_{r,d}}^{obj})$, where ${D_{r,c}}^{obj}$ denotes the c^{th} $(c = 1,2,3, \dots\dots, d)$ feature value corresponding to the r^{th} object. In contrast to an object data, a relational data generally consists of a set of n^2 numerical relationships, say $\{R_{r,c}: r = 1,2,3, \dots\dots, n = 1,2, \dots\dots, n\}$, for every possible pair of objects. In other words, $R_{r,c}$ denotes the degree to which r-th and c-th objects are associated with each other via some binary relationship. Noise cancellation, filtering, encoding, and augmentation are all examples of pre-processing operations that may be carried out during the data gathering phase in preparation for mining the pattern vectors [15]. For example, if the input data are an image, this phase extracts the related features in order to build a good recognition system.

3. Proposed Feature Selection Technique

Finding the right amount of features to pick in order to effectively represent the feature space is one of the main issues with the majority of the current unsupervised feature selection algorithms, including the previously suggested IMoDEUFS and DSFFC methods. The size of the condensed feature set is typically set at 50% of the total number of original features[16]. The growth phase of these two stages continues until no more vertices can be added. Algorithms 1 and 2 give the fundamental structures for the expansion and shrinking stages, respectively. Algorithm 1 takes as input: (a) a current support CurSupport, (b) a current point CurPoint, and (c) an affinity matrix A. Step 1 first builds a subgraph $SG_D = (V_D, E_D)$, where V_D is the set of all vertices belonging to the current support CurSupport and $E_D = \{(vi, vj)|(vi, vj) \in E, vi \in$

V_D, vj ∈ V_D} is set of all edges induced on V_D. Steps 2-9 then obtain all the neighbors of SG_D excluding the vertices belonging to the current vertex set V_D and stores them into a new set V_C. Steps 10-12 then compute the reward $Reward_{vi}$ for every vertex $V_i \in V_C$. Steps 13-18 first updates the reward $Reward_{vi}$ for every vertex $v_i \in V_C$ by subtracting curValue from it and then updates the vertex set V_C in such a way that a vertex having a reward equal to or less than a pre-defined threshold value (minValue) is discarded from it. Steps 19-23 then compute the ω. Step 24-29 then the calculate expansion vector b to enhance the probability of vertices having large rewards and to reduce the probability of vertices belonging to the current subgraph. Additionally, a new support vector New Support is also created along this new direction. In this regard, step 27 also updates curPoint by adding a zero vector of size $|V_C|$ with it. Then step 31 computes coeff in an effort to increase the gap between the previously discovered affinity value and the newly discovered affinity value. Steps 32-35 calculate the value of bestt depending upon the value of coeff thus obtained and subsequently steps 36-38 update the current point. Step 39 then computes the affinity value of the updated current subgraph induced on New Support. Finally, step 40 stores CurrentPoint, NewSupport and curValue into three other variables, namely, Expanded Point, Expanded Support and expanded Value.

Algorithm 1 Expansion Algorithm

Input: CurSupport, CurPoint and A.

Output: ExpandedSupport, ExpandedPoint and expandedV alue.

Algorithm:

Step 1: Construct $SG_D = (V_D, E_D) using$ CurSupport.
Step 2: $\boldsymbol{V_c} = \emptyset$.
Step 3: for each vertex $v_i \in V_D$
Step 4: for each vertex $v_j \in N$ Neighbors (v_i)
Step 5: if $v_j \notin V_D$
Step 6: $V_c = V_c \cup v_j$
Step 7: end if
Step 8: end for
Step 9: end for
Step 10: for each vertex $v_i \in V_c$
Step 11: $\text{Reward}_{vi} = \sum_{vj \in VD \wedge vi \in Neighbors\, Vj} CurPoint_{Vj} * A_{v_{j,Vi}}$
Step 12: end for
Step 13: for each vertex $v_i \in V_c$
Step 14: $\text{Reward}_{vi} = \text{Reward}_{vi} - CurValue.$
Step 15: if $\text{Reward}_{vi} \leq \min Value$
Step 16: $V_c = V_c | v_i$
Step 17: end if
Step 18: end for
Step 19: $s = 0;\ s_2 = 0$
Step 20: for each vertex $v_i \in V_c$

Step 21: $s = s +$ Reward$_{vi}$; $s_2 = s_2 +$ Reward$_{vi} *$ Reward$_{vi}$
Step 22: end for
Step 23: $\omega = \sum_{v_i, v_{j \in V_c}}$ Reward$_{vi} * A_{v_i, v_j} *$ Reward$_{vj}$.
Step 24: for each vertex $v_i \in V_D$
Step 25: NewSupport$_{v_i}$ = CurSupport$_{v_i}$; $b_{v_i} = -$CurSupport$_{v_i} * s$.
Step 26: end for
Step 27: CurPoint = {CurPoint zeros $(1, |V_c|)$}.
Step 28: for each vertex $v_i \in V_c$
Step 29: NewSupport$_{v_i} = v_i$; b_{v_i} = Reward$_{v_i}$.
Step 30: end for
Step 31: $coef\ f =$ curV alue $*$ s $*$ s + 2 $*$ s $*$ s2 – ω;
Step 32: $best\ T = \frac{1.0}{s}$;
Step 34: $best\ T = \min(\frac{1.0}{s}, \frac{s2}{coeff})$;
Step 35: end if
Step 36: for each vertex $v_i \in$ NewSupport
Step 37: $CurPoint_{v_i} = CurPoint_{v_i} + bestT * b_{v_i}$
Step 38: end for
Step 39: $CurValue = CurValue - cief\ f * bestT * best\ T + 2$ * s2 * bestT;
Step 40: ExpandedPoint = CurPoint, ExpandedSupport = NewSupport and expandedV alue = curV alue.

On the other hand, Algorithm 2 takes as input: (a) a current support CurSupport, (b) a current point CurPoint, and (c) an affinity matrix A. Step 1 first builds the current subgraph S_{GC} = (V_C,E_C) corresponding to the current support CurSupport of a given graph G = (V,E). Step 2 then initializes a temporary variable preValue used to store the affinity value of the subgraph obtained in the previous iteration with a value of 0. For each vertex $v_i \in V_C$, steps 3-6 then compute the reward Reward$_{vi}$ using the current point CurPoint and the affinity matrix A as follows.

$$\text{Reward}_{vi} = \sum_{vj \in Neighbors\ Vi} CurPoint_{Vj} * A_{v_{j,Vi}} \quad (1)$$

where Neighbors$_{vi}$ is the set of all neighbors of the vertex v_i.

For each vertex vi ∈ VC, steps 8-10 then update the current point CurPointv_i by multiplying it by the corresponding reward Reward$_{vi}$. In this regard, steps 8-10 also computes the average affinity value (as defined in Equation (1)) of the subgraph SGC and assigns the resultant value into another variable curV alue. Subsequently, steps 11-13 normalize the updated current point CurPointvi by dividing it by the sum of all individual elements of CurPoint. Steps 14 then calculates the ratio between the initial affinity value (preV alue) and the more updated affinity value (curV alue). Step 15 then sets preV alue with curV alue [17]. Step 16 then verifies whether the value of the ratio is less than some predefined value of a threshold parameter, say minChange. If the answer is “no”, the shrinking phase again follows the same

procedure (starting from step 3 up to step 17) of computing the reward vector, updating the current point vector, normalizing the updated current point, and computing the average affinity value of the updated current subgraph. Otherwise, this phase first updates the subgraph SGC by removing those vertices whose value of the current point is determined to be below or equal to a certain value (alpha) and subsequently obtains a new support NewSupport and the corresponding new point NewPoint (steps 19-29). Finally, step 30 stores them into two other variables, namely, ShrinkedPoint and Shrinked Support.

Algorithm 2: Shrinking Algorithm
Input: CurSupport, CurPoint and A.
Output: ShrinkingSupport, Shrinking Point.
Algorithm:
Step 1: Construct $SG_c = (V_c, E_D) using$ CurSupport.
Step 2: pre Value=0
Step 3: while(TRUE) do
Step 4: for each vertex $v_j \in V_c$
Step 5: compute

$$\text{Reward}_{vi} = \sum_{vj \in Neighbors\ Vi} CurPoint_{Vj} * A_{v_{j,Vi}}$$

Step 6: end for
Step 7: $CurV\ alue = 0;$
Step 8: for each vertex $v_i \in V_c$
Step 9: compute

$$CurPoint_{Vi} = CurPoint_{Vi} * \text{Reward}_{vi},$$
$$CurV\ alue = CurV\ alue + CurPoint_{Vi}.$$

Step 10: end for
Step 11: for each vertex $v_i \in V_c$
Step 12: compute

$$CurPoint_{Vi} = CurPoint_{Vi}/CurV\ alue.$$

Step 13: end for
Step 14: ratio= $(CurValue - preV\ alue)/preV\ alue.$
Step 15: $preV\ alue = CurValue.$
Step 16: if $(ratio < minChange) break.$
Step 17: end while
Step 18: $sum = 0, isModify = 0.$
Step 19: for each vertex $v_i \in V_c$
Step 20: $if\ (CurPointvi > alpha)$
Step 21: $\text{NewPoint}_{v_i} = \text{CurPoint}_{v_{i,}}$

$$\text{NewSupport}_{v_i} = \text{CurSupport}_{v_i},$$
$$sum = sum + \text{CurSupport}_{v_i}$$

Step 21: else
Step 22: $isModify = 1; Vc = Vc|vi.$
Step 23: end if

Step 24: end for
Step 25: if $isModify == 1$
Step 26: for each vertex $v_i \in V_c$
Step 27: $\text{NewPoint}_{v_i} = \text{NewPoint}_{v_i}/sum$
Step 28: end for
Step 29: end if
Step 30:
$\text{ShrinkedPoint} = \text{NewPoint}$ and $\text{ShrinkedSupport} = \text{NewSupport}$.

4. Results and Discussion

Here, a large number of experiments are run to evaluate the proposed method's efficacy in relation to other existing unsupervised feature selection(UFS) techniques [18]. The performance of the suggested technique is tested against several other unsupervised feature selection.

Table 1
The frequency with which each of six unsupervised feature selection techniques yields the best results is summarized.

Datasets	UFSFS	LSFS	MCFS	UDFS	DSFFC	SEAFC
Colon	0	0	2	0	4	2
Multiple Features	0	0	2	0	1	1
Isolet	0	0	2	0	1	1
Spambase	0	0	0	0	6	2
Ionosphere	0	0	1	0	1	4
WDBC	0	2	1	0	0	4
Sonar	0	2	0	0	0	6
SPECTF	0	0	0	0	3	3
Overall	0	4	8	0	16	23

The suggested technique outperforms all existing classifiers for the seventh dataset Sonar, with the exception of the Naive Bayes classifier. For two classifiers, SVM and Naive Bayes, DSFFC is shown to deliver superior performance than the others for the eight dataset SPECTF, whereas SEAFC does so for the other two classifiers. Table.1 illustrates the number of occurrences where each of the aforementioned six algorithms attains the best results for each dataset [19,20]. According to Table.1, the suggested algorithm ranks first since it achieves the greatest outcomes in the most iterations possible (= 23). In this case, the DSFFC algorithm ranks second since it only has to get the best outcomes eighteen times to achieve this ranking. Alternatively, the MCFS algorithm places the third position; the LSFS algorithm ranks the fourth position. It is also to be noted that both the UFSFS and the UDFS algorithms cannot attain the best result in any case out of the total fifty-six cases.

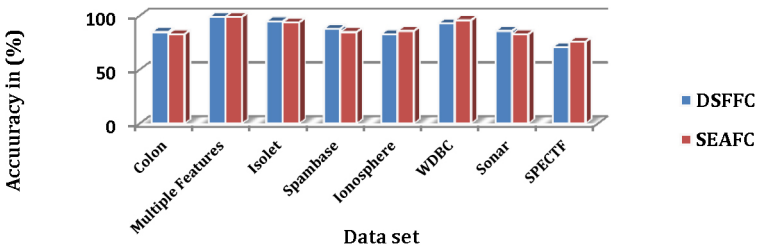


Figure 1

Dataset wise SVM Accuracy Comparison for the two methods

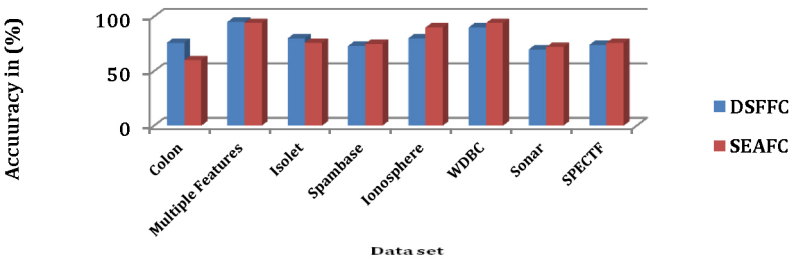


Figure 2

Dataset wise NaiveBayes Accuracy Comparison for the two methods

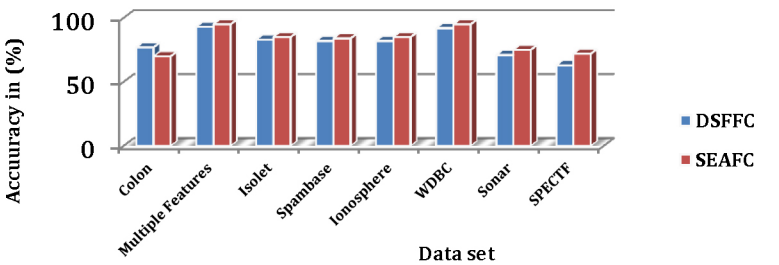


Figure 3

Datasetwise KNN Accuracy Comparison for the two methods

The comparison of dataset-wise classification accuracy of these two methods is shown in Figure 1 to Figure 4. associated with four different classifiers, viz., SVM, Naive Bayes, k-NN, and AdaBoost. For the SVM classifier, the proposed algorithm is found to provide almost the similar performance like the DSFFC algorithm on the first four datasets, however,

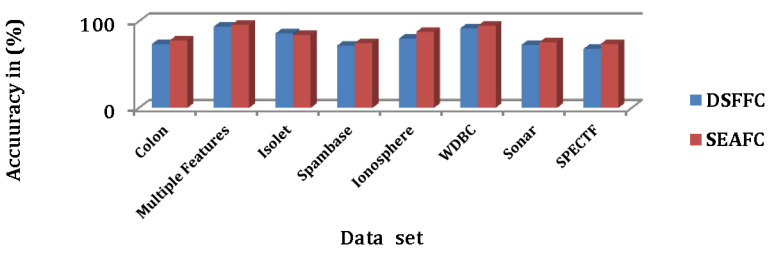


Figure 4

Dataset wise AdaBoost Accuracy Comparison for the two methods

for the final four datasets, the proposed approach outperforms the DSFFC technique in all save the second last. Corresponding to these two classifiers, there exist two datasets, namely, Colon-solet, for which the DSFFC algorithm is found to perform better than the proposed approach.

5. Conclusion

This study presents a unique method for selecting unsupervised features. An innovative shrinking-and-expanding technique has been used in feature selection to identify dense feature subgraphs in the dataset. The suggested feature selection technique comes with two key benefits. One benefit is that it can choose an optimal set of characteristics without any human input, which is a known problem related to many existing algorithms of UFS. The second benefit is that the suggested approach obtains the reduced feature subset with good accuracy and MCC scores while using just around the original characteristics were reduced to between one-fifth and one-third. We have compared the suggested algorithm's performance to that of five existing approaches, viz., UFSFS, LSFS, MCFS, UDFS and DSFFC. According to the outcomes of the experiments, the suggested method is effective and picks less relevant features than previous state of USF selection techniques, which improves classifier performance.

References

[1] C. Lee and G. G. Lee., "Information Gain and Divergence Based Feature Selection for Machine Learning Based Text Categorization," Information Processing & Management, vol. 42, no. 1, pp. 155–165, (2016).

[2] S. Bandyopadhyay, T. Bhadra, U. Maulik, and P. Mitra., "Integration of Dense Subgraph Finding with Feature Clustering for Unsupervised Feature Selection," Pattern Recognition Letters, vol. 40, pp. 104–112, (2018).

[3] S. Yu, "Human resources management and evaluation system based on fuzzy analytic hierarchy process", *Journal of Interdisciplinary Mathematics*, vol. 20, no. 4, pp. 951–964, (2017).

[4] T. Bhadra and S. Bandyopadhyay., "Unsupervised Feature Selection Using an Improved Version of Differential Evolution," Expert Systems with Applications, vol. 42, no. 8, pp. 4042–4053, (2020).

[5] H. Liu, L. J. Latecki, and S. Yan., "Fast Detection of Dense Subgraphs with Iterative Shrinking and Expansion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 9, pp. 2131–2142, (2021).

[6] S. R. Kawale, S. P. Diwan and D. G. V, "Intelligent Breast Abnormality Framework for Detection and Evaluation of Breast Abnormal Parameters," 2022 International Conference on Edge Computing and Applications (ICECAA), pp. 1503-1508, (2022).

[7] L. Faivishevsky and J. Goldberger., "Unsupervised Feature Selection Based on Non-Parametric Mutual Information," in IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, (2020).

[8] P. Pavankumar, N. K. Darwante, "Performance Monitoring and Dynamic Scaling Algorithm for Queue Based Internet of Things," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp. 1-7, (2022).

[9] Saxena, N. R. Pal, and M. Vora., "Evolutionary Methods for Unsupervised Feature Selection Using Sammon's Stress Function," Fuzzy Information and Engineering, vol. 2, no. 3, pp. 229–247, (2017).

[10] Bahmani, R. Kumar, and S. Vassilvitskii., "Densest Subgraph in Streaming and Mapreduce," Proceedings of VLDB Endowment, vol. 5, no. 5, pp. 454–465, (2020).

[11] K. S and M. R. Arun, "Priority Queueing Model-Based IoT Middleware for Load Balancing," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 425-430, (2022).

[12] M. Mandal and A. Mukhopadhyay., "Unsupervised Non-Redundant Feature Selection: A Graph-Theoretic Approach," in Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), 2013, pp. 373–380,(2019).

[13] J. Quan, "Influential factors and case analysis of economic performance of tourism industry based on regression analysis", *Journal of Interdisciplinary Mathematics*, vol. 20, no. 4, pp. 965–977, (2017).

[14] M. Nagabushanam, H. G. Govardhana Reddy & K. Raghavendra. "Vector space modelling-based intelligent binary image encryption for secure communication," *Journal of Discrete Mathematical Sciences and Cryptography*, 25:4, pp.1157-1171, (2022).

[15] Arun Kumar, Pragati Tripathi, M. A. Ansari & Alaknanda Ashok, "Novel scheme of k-SVM analysis using PCA and NN for detection of MRI brain images," *Journal of Interdisciplinary Mathematics*, 23:5, 967-976, (2020).

[16] S. Bandyopadhyay and S. Saha., Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications, 1st ed. Berlin, Germany: Springer Science & Business Media, (2018).

[17] Guyon and A. Elisseeff., "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, no. Mar, pp. 1157–1182, (2019).

[18] Hua, W. Tembe, and E. R. Dougherty., "Feature Selection in the Classification of High-dimension Data," in IEEE International Workshop on Genomic Signal Processing and Statistics, pp. 1–12, (2018).

[19] Rahul Adhao & Vinod Pachghare, "Feature selection using principal component analysis and genetic algorithm," *Journal of Discrete Mathematical Sciences and Cryptography*, 23:2, 595-602, (2020).

[20] P. A. Est´evez, M. Tesmer, C. A. Perez, and J. M. Zurada., "Normalized Mutual Information Feature Selection," IEEE Transactions on Neural Networks, vol. 20, no. 2, pp. 189–201, (2021).