International Journal of Information Technology and Language Studies (IJITLS)

Comparative Study on Fast Feature Selection

Ahood Naif Alharbi and **Mohamed Dahab**
aalharbi1482@stu.kau.edu.sa; mdahab@kau.edu.sa

King Abdul-Aziz University, Saudi Arabia

Abstract. No one can deny how feature selection became an important aspect of the machine learning field. Feature selection has proved its ability to overcome the problem that known as the curse of dimensionality, which raises when a number of features for a given data is large and required to be expressed in a space of high dimensions. In this paper, we are going to investigate the algorithms in feature selection as well as the enhanced ones in the acceleration of the feature selection process. We have shown the algorithms, their limitations, and how authors have enhanced them.

Keywords: Classifications, Solution Tree, HPC, Feature Selection

1. Introduction

The enormous data that is collecting every day is one of the challenges that faced data analysis field in order to get control and use of the given data. Among these data, there are lots of features some are relevant and some are irrelevant (noise). The irrelevant and redundant data may increase the training time and affect the predictive model accuracy. Therefore, narrowing the number of features or attributes is a solution for the problem that known by the curse of dimensionality, which has been studied in many types of research to enhance the performance in term of speed, predictive power, and reduce dimensionality (Liu & Motoda, 2007). In machine learning, such processing of the data falls into two categories namely feature selection and feature extraction.

Mainly feature selection is to select a small set of features (variables, predictors) that best capture the characteristics of the problem being addressed by removing irrelevant or redundant features (Liu & Motoda, 2007). Feature extraction is to create new features by transforming a space composing of many dimensions into a space of fewer dimensions (Jain & Zongker, 1997).

In this paper, our scope will focus only on Feature selection, which is highly demanded technique when a number of features are high compared to the number of samples; e.g., analysis of written text. Three general methods are typical in feature selection algorithms, which are filter method, wrapper method, and embedded method. In the ***filter*** method, features are selected independently of the predicted model while the ***wrapper*** method takes into account the model that will use them. Lastly, the ***embedded*** method is selecting features during the learning process (Liu & Motoda, 2007).

Considering the problem of the curse of dimensionality and its effect in many applications, lead many authors to investigate this problem and proposed many solutions. Some of these solutions were enhanced to be faster either by using High-Performance Computing (HPC) or modify the same algorithm for better performance.

The rest of the paper is organized as follows. In section 2, a comparison between the fast feature selection algorithms is presented, followed by the results of the comparison in section 3, and finally, in section 4, we present the conclusion.

2. Comparative Between Fast Feature Selection Algorithms

In this section, we represent a background about feature selection then exploring the algorithms that have been enhanced on the feature selection process.

2.1. Feature Selection

Feature selection generally consists of two basics components features search and features subset evolution measure (Liu & Motoda, 2007). There are many algorithms used to search for the suitable subset of features but some guarantee the optimal selection of features and some are not. Evolution measure is one of the three main categories of feature selection methods: wrapper, filter and embedded method, which are used to score different feature subsets.

A given taxonomy by Anil and Douglas for feature selection algorithms has divided them into two categories, one is the algorithms that based on statistical pattern recognition (SPR) and the second category is the algorithms that using artificial neural networks (ANN) (Jain & Zongker, 1997). Then, SPR algorithms divided into a single solution, which belongs to the algorithms that store single feature subset then does some modification to it and multiple solutions, which refer to the algorithms that maintain a population of subsets. Another distinguished between the algorithms is whether they give the same subset of features for a given problem in every run (**Deterministic**) or different result different subsets at each run (**Stochastic**) (Jain & Zongker, 1997).

2.2. Optimal Algorithms

This part illustrated the algorithms that guarantee the optimal features subsets selection, which enhance the predictive model performance in term of speed and accuracy. Exhaustive and branch and bound feature selection algorithms are exponential search methods but they were known for their ability to find the optimal required subset of features (Jain & Zongker, 1997). In the following subsections, both algorithms are shown.

2.2.1. Exhaustive Feature Selection Algorithm

In features search, the simplest algorithm is to try all the possible subset of features and choose the optimal one known as exhaustive search. Despite its ability to find the optimal subset of feature, this search method is insufficient and extremely computationally costly especially in a large number of features because it is growing exponentially (Liu & Motoda, 2007). For instance, using exhaustive search to solve knapsack problem, which defined as a given set of elements n, each one has a weight w and value v; determine number of elements to include in a collection, which satisfied the following condition; the total elements' weight is less or equal to a given limit and the total elements' value is as large as possible. Such problem in exhaustive search method would cost 2^N (where N is the number of features) to traverse all possible subsets of features, which is inefficient search method and impractical in term of the consumed time and computational cost (Levitin, 2012).

2.2.2. Branch and Bound (BB) Feature Selection Algorithm

The impractical of the exhaustive search has motivated many researchers to come up with a method that speeding up feature selection process and reduce computational cost. In 1977 a branch and bound feature selection algorithm was developed by Narendra and Fukunaga and has scored efficient results than the exhaustive search (Narendra & Fukunaga, 1977). It guarantees to find the optimal feature subset without evaluating all possible subsets. Basically, it is constructing an ordered tree whose root node consists of the original set of features, searching for the optimal features subset k among N features with respect of a certain criterion function. For each tree level, a limited number of sub-trees is generated by deleting one feature from the set of features from the parent node. The following figure (1) illustrated the constructing tree in branch and bound algorithm.

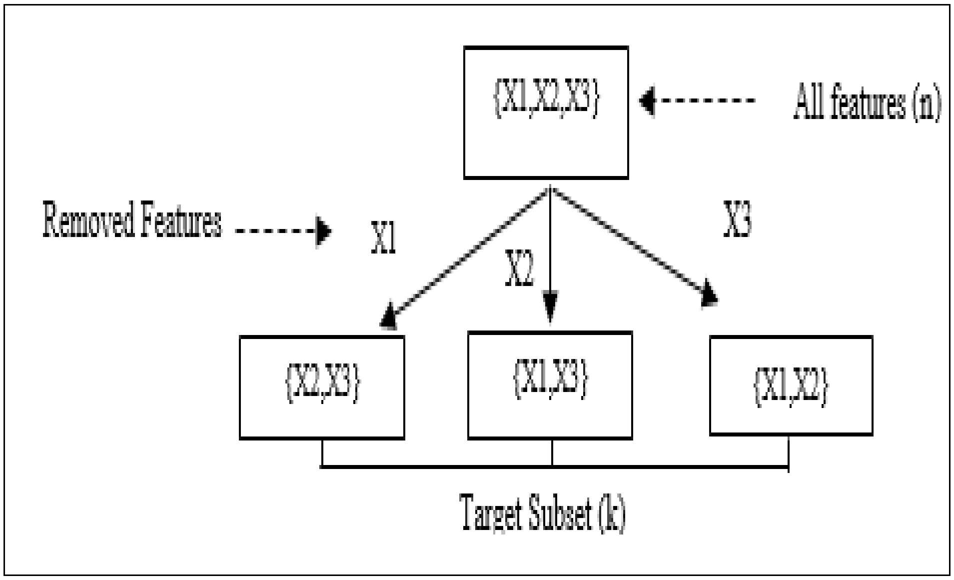

Figure 1. Branch and Bound Tree

One important aspect of BB algorithm is assuming that the criterion function is monotonically increasing. So, for a given two feature subset k={2,4} and N={2,4,5} and feature criterion function (J);

$$K \subset N \text{ then } J(k) < J(N)$$

BB algorithm traverses the tree from right to left in the depth-first search pattern. If the value of the criterion is less than the threshold (relevant to the most recent best subset) at a given node, all its successors will also have a value less than a criterion value. Anytime the criterion value J(Nm) in some internal node is found to be lower than the current bound, due to the monotonicity condition, the whole subtree may be cut off and many computations may be omitted. To prevent unnecessary repetition in the calculation, features only allowed to be removed in increasing order. The leaves nodes of the tree in BB algorithm is related to all subsets of k features while the number of levels denoted by N-k +1. Branch and Bound algorithm considered to be more efficient than the exhaustive search because it constructed a tree with all possible feature subset but exploring only promising branches. In the worst case, the BB algorithm computed the criterion function in every tree node same as exhaustive search (Narendra & Fukunaga, 1977).

2.2.3. Adapted Branch and Bound Algorithms

For its efficiency in many applications, Branch and Bound has attracted many researcher's attention. By 1993, Yu and Yuan suggested more efficient Branch and Bound algorithm named BB+ (Yu & Yuan, 1993). It relied on minimizing the solution tree by removing single branching intermediate nodes for less calculation of redundant criterion function (J). Also, in 2003, an improved method was applied to the BB algorithm and lead to better results compared to original BB and BB+ (Chen, 2003). Their method mainly uses top-down and right-left search strategies together with backtracking. So, they get use for the information that gained in the previous search. Such way will definitely reduce the number of evolution nodes and speed up the selection process.

Another research has been done by Somol and Pudil in 2004 to reduce the calculation in branch and bound algorithm using simple prediction mechanism to estimate J rather than compute the true J of each node (Somol et al., 2004). Calculating the estimated J costs fewer operations and therefore speed up the search than the calculation of the true J. The true criterion function J for a node is only computed if and only if the estimated J for a node is less than the current bound. Their method guarantees the optimal subset selection of features because the cut off for any node is based on the true J criterion function and never cut off based on J estimation value. However, due to the using of estimated J in ordering the nodes, inaccurate J value can lead to inefficient node ordering, which cost more calculation using true J calculation.

Moreover, a new adaptive method to BB algorithm, which is slightly different from the previous versions were done by Nakariyakul and Casasent (Nakariyakul & Casasent, 2007). They aimed to speed up the search for the optimal subset of features by ignoring redundant criterion function calculations; taking into account several prosperities. Before searching the tree for the optimal subset of features, they have constructed the tree based on the most significant features and defined the initial bound using sequential forward floating selection (SFFS) method. Selecting good initial bound would cut off many branches early and therefore minimize J calculations. Another property of the algorithm is defining a new strategy to jump among levels in which at least one successor node to be less than the defined bound. The new adaptive method showed promising results across four different databases compared to the previous adaptive methods.

In 2012 (Reis, 2012), a PhD thesis has improved the use of U-curve assumption to find the feature set of the minimum cost that proposed earlier by Reis (Ris et al., 2010). Their proposed algorithm U-CURVE-BRANCH-AND-BOUND (UBB) use U-curve assumption to find the feature set of minimum cost without going throw all elements. UBB is using a recursive enumeration scheme, constructing the tree and then use it as the search space. The tree pruning process occurs when the cost of an element in the search chain starts increasing. In the early iterations of the UBB algorithm, finding a minimum element in a chain leads to eliminate many nodes in the tree. However, the pruning became slower as long as going toward later iterations because the search chain is not the best possible chains in the search space. Recently in 2018, a paper has proposed a fast Branch and bound algorithm with the assumption of U-curve for the cost function and investigated its application in the design of imaging W-operators and in feature selection for classifier design (Atashpaz-Gargari et al, 2018). In IUBB algorithm, they have used a different search structure, which focuses on the optimal chain in the search space at each step of the algorithm. Unlike the UBB algorithm, they have used the U-curve assumption not only to prune the nodes but also for finding the minimum element of each chain. Their experiments were applied to both algorithms (UBB and IUBB), its showing that the IUBB need a smaller number of function evolution and have used the U-curve assumption efficiently than UBB.

2.3. Non-Optimal, Deterministic, Single-Solution Methods

On the other hand, there are many algorithms serving in feature selection field, but they do not guarantee the optimal subset of features. Those algorithms start with a single solution then add or eliminate features until a certain criterion is satisfied (single solution) and known to be deterministic (give the same subset of features for a given problem in every run). The most popular and used algorithms in this category are known by sequential search methods (Liu & Motoda, 2007). Sequential search methods are Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS). SFS is introduced by Whitney (Whitney, 1997) in 1971 and it mainly begins with an empty subset of features and sequentially adds one feature at a time due to its most contribution to the criterion function (Liu & Motoda, 2007) (Jain & Zongker, 1997). While SBS is defined by Marill and green (Marill & Green, 1963) in 1963; it starts with all features and sequentially discards features that least contribute to the criterion function (Liu & Motoda, 2007) (Jain & Zongker, 1997). Both SFS and SBS have similar performance and do not consider all possible subset of features, so they do not ensure the optimal solution (Jain & Zongker, 1997). SFS and SBS are suffering from a nesting effect, which means once the feature is selected in SFS or eliminated in SBS, it can't be deleted or reselected respectively (Liu & Motoda, 2007). To overcome this problem, Plus-l-Minus-r (l-r) search method was developed in 1976 by Stearns (Strearns, 1976) where the value of l and r are predefined and can't be changed.

In 1994, Pudil, Novovičová, and Kittler have introduced an adapted sequential method called floating search methods, which allow the value of l and r to float (Pudil et al., 1994). The floating methods depending to the search procedures it is either SFFS or Sequential Backward Floating Selection (SBFS). They allowed the values of l and r to be more flexible for change, inclusion and excluding until approximate the optimal solution. So, both floating methods SFFS and SBFS are free to modify the false decision in previous steps for more efficient results near to the optimal subsets unlike SFS and SBS.

2.4. Non-Optimal, Deterministic, Multiple-Solution Methods

Another type of the algorithms is the one that generates many candidate population subsets that best met the criterion function (multiple solutions) and known to be deterministic (give the same subset of features for a given problem in every run). In 1988, a paper has investigated one of these algorithms named best first strategy taken from Artificial intelligence used as a search method in a weighted tree for the best subset of features. Despite the algorithm not give the optimal subset for any criterion that satisfies monotonicity, it is computationally less than the branch and bound (Xu et al, 1988). Also, a beam search algorithm has been investigated in feature selection, which is a type of best first search. Beam search reduces the number of calculation by pruning the unpromising node in each level of the tree and only keep the best features for further branching. Since no recovering if pruning decision was wrong, beam search does not guarantee the optimal subset of features as best first search (Gupta, 2002).

2.5. Non-Optimal, Stochastic, Multiple-Solution Methods

Another type of algorithms is the one that generates a different subset of features in each run due to its random selection in every run (multiple solutions) and known to be stochastic (give a different subset of features for a given problem in every run). Genetics algorithm (GA) is introduced by Siedlecki and Sklansky in feature selection field (Siedlecki & Sklansky, 1993). The feature subset in GA represented by a binary string with length n called chromosome. The chromosome has either 1 or 0 value in a certain position i that determine the presence and absence of a feature i respectively. The fitness of each chromosome is determined whether this chromosome will survive to next generation or discarded. Crossover process is used to create a new child (offspring) from the parents by mixing two chromosomes where the mutation is creating a new child by changing a single pit randomly in a single survived chromosome. The algorithm showed better results to find the near optimal subset of features compared to all non-exhaustive search methods (Siedlecki & Sklansky, 1993).

Recently, a research has been investigated in feature selection using a genetic algorithm and HPC (Alwadei et al, 2018). Their main contribution was to accelerate feature selection on the BCI dataset (the P300 based system). Reducing the consumed time with acceptable accuracy would enhance the BCI performance. They have used Genetic Algorithm (GA) and Differential Evolution (DE) as a search algorithm while Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) are used as a classifier. DE-SVM has promising results with an accuracy of 80% selecting 42% of the original features only.

Also, among the most popular machine learning methods in this type of algorithms is Random Forest (RF). Random forest is a growing forest of random trees on bagged samples that lead to excellent results compared with the best-known classifiers (Liu & Motoda, 2007). It is computationally powerful and lead to highly accurate classifiers for large datasets as well as handle huge numbers of input features and estimate the relevance of features in determining the best split of the tree. A recent paper has an ensemble classification framework using RF on the basis of estimation of most relevant input features (Chutia et al, 2017). To define the optimal subset of features that will improve the learning accuracy and minimize the execution time for the RF classifier, they have used Correlation-based feature selection (CFS). An experiment was carried out on different types of images from the Landsat Enhanced Thematic Mapper Plus (Landsat ETM1) and Quick Bird sensors. They have noticed that the performance of the RF classifier was noticeably improved while using the optimal set of relevant features compared with a few of the most advanced supervised classifiers like Naïve Bayes and Support Vector Machine.

2.6. Node Pruning

Node pruning in feature selection is similar to the process of removing least contribution features from large feature set. Dashti & Wijs (Dashti & Wijs, 2007) proposed an approach using the feedforward network with a back propagation learning method and compared their method with Whitney's feature selection method. They pruned the least salient nodes in the network using node saliency measure that defined as the sum of the increase in error to all the training patterns. They have noticed that the pruning node method is outperformed other classical selection methods because it optimizes both the feature set and the classifier and achieves higher accuracy result (86.3 %) compared to Whitney's accuracy result (81.8%) (Dashti & Wijs, 2007).

3. Discussion

From the previous studies, we have found that exhaustive search is guaranteed the optimal subset of features, but it is computationally unfeasible. On the other hand, the BB algorithm is also guaranteed to find the optimal feature subset without evaluating all possible subsets and consider to be more efficient than the exhaustive search. For its efficiency, many adapted BB algorithms have been suggested to modify the original one for more better results. Another selection algorithms described to be non-optimal due not consider all possible subset of features like forward and backward selection, but it is computationally less than the branch and bound. Also, Random algorithms have played a significant role in the selection process like GA showed better results to find the near optimal subset of features compared to all non-exhaustive search methods but it lacks to guarantee the optimal subset of features. Considering the optimal subset of features, the branch and bound algorithm have scored promising results across different datasets but none of the mentioned studies have used HPC to accelerate the selection process. From this point of view, we aim to implement BB using HPC in order to find the optimal feature subsets within shorten time. In Table 1, we have summarized the comparison results.

Paper Name	Algorithm used	Dataset	How to select features	Results and Limitations
Computational methods of feature selection (Liu & Motoda, 2007)	Exhaustive search	-	Try all the possible subset of features and choose the optimal one	▪ Guarantee the optimal subset of features ▪ Computationally unfeasible
A Branch and Bound Algorithm for Feature Subset Selection (Narendra & Fukunaga, 1977)	Branch and Bound (BB)	Multispectral data acquired from airborne remote sensing scanners	Constructing An ordered tree, searching for the optimal features subset k among N features with respect of a certain monotonically criterion function	▪ It guarantees to find the optimal feature subset without evaluating all possible subsets ▪ More efficient than the exhaustive search ▪ In the worst case, the BB algorithm is an exhaustive search
A more efficient branch and bound algorithm for feature selection (Yu & Yuan, 1993)	Adaptive branch and bound (BB+)	Two groups of samples from two types of symbols in the circuit diagram.	Minimize the solution tree by removing single branching intermediate nodes for less calculation of redundant criterion function (J)	▪ More efficient than original BB algorithm ▪ It does guarantee that the selected feature subset is globally optimum as BAB does
An improved branch and bound algorithm for feature selection (Chen, 2003)	Adaptive branch and bound (IBB)	Three data sets from the UCI repository, two letter image recognition and one mammogram data	Use top-down and right-left search strategies together with backtracking. So, they get use for the information that gained in the previous search.	▪ Lead to better results compared to original BB and BB+ ▪ Reduce the number of evolution nodes and speed up the selection process.
Fast branch & bound algorithms for optimal feature selection (Somol et al., 2004)	Adaptive branch and bound	Using three real data sets using traditional probabilistic distance criteria.	Using simple prediction mechanism to estimate J rather than compute the true J of each node. Calculating of the estimated J cost less operations and therefore speed up the search than calculation of the true J.	▪ Guarantee the optimal subset selection of features ▪ Due to the using of estimated J in ordering the nodes, inaccurate J value can lead to inefficient node ordering which cost more calculation

Adaptive branch and bound algorithm for selecting optimal features (Nakariyakul & Casasent, 2007)	Adaptive branch and bound	Four different datasets from the UCI machine learning repository, handwritten letter recognition , mammogram data, SPECTF heart and a sonar databases	Constructed the tree based on the most significant features and defined the initial bound using sequential forward floating selection (SFFS) method. Selecting good initial bound would cut off many branches early and therefore minimize J calculations. Defining new strategy to jump among levels in which at least one successor node to be less than the defined bound	▪ Showed promising results across four different databases compared to the previous adaptive methods.
Minimization of decomposable in U-shaped curves functions defined on poset chains (Reis, 2012)	Adaptive branch and bound (UBB)	Simulated and real-world data.	Use U-curve assumption to find the feature set of minimum cost without going throw all elements. UBB is using a recursive enumeration scheme, constructing the tree and then use it as the search space. The tree pruning process occurs when the cost of an element in the search chain starts increasing.	▪ In the early iterations of the UBB algorithm, finding a minimum element in a chain leads to eliminate many nodes in the tree ▪ The pruning is become slower as long as going toward later iterations
A fast Branch-and-Bound algorithm for U-curve feature selection (Atashpaz-Gargari et al, 2018)	Adaptive branch and bound (IUBB)	Investigated the algorithm in the design of imaging W-operators and in classification feature selection, using the average mean conditional entropy (MCE) as the cost function for the search.	Using a different search structure which focuses on the optimal chain in the search space at each step of the algorithm. Unlike the UBB algorithm, they have used the U-curve assumption not only to prune the nodes but also for finding the minimum element of each chain.	▪ IUBB need a smaller number of function evolution and have used the U-curve assumption efficiently than UBB.
A Direct Method of Nonparametric Measurement Selection (Whitney, 1997)	Sequential Forward Selection (SFS)	-	begin with an empty subset of features and sequentially adds one feature at a time due its most contribution to the criterion function	▪ Do not consider all possible subset of features, so they do not ensure the optimal solution ▪ Suffering from *nesting effect*

On the effectiveness of receptors in recognition systems (Marill & Green, 1963)	Sequential Backward Selection (SBS)	The raw data were obtained by asking subjects to print letters A, B, C ,D by hand. Each of the 40 subjects produced 5 examples of each of the four letters. The sample of 800 individual characters. Each printed within a two-inch square.	Starts with all features and sequentially discard features that least contribute to the criterion function	▪ Do not consider all possible subset of features, so they do not ensure the optimal solution ▪ Suffering from *nesting effect*
Floating search methods in feature selection (Pudil et al., 1994)	Sequential Forward Floating Selection (SFFS) and Sequential Backward Floating Selection (SBFS).	Various types of data	Allowed the values of l and r to be more flexible for change, inclusion and excluding until approximate the optimal solution. So, both floating methods SFFS and SBFS are free to modify the false decision in previous steps	▪ More efficient than SFS and SBS ▪ Do not guarantee the optimal subset of features.
Best first strategy for feature selection (Xu et al, 1988)	Best first strategy	-	Search in a weighted tree for the best subset of features.	▪ It is computationally less than the branch and bound ▪ Do not guarantee the optimal subset of features
Beam search for feature selection in automatic SVM defect classification (Gupta, 2002)	Beam search algorithm	The data set is comprised of about 3000 images, with 13 defect classes.	Reduce the number of calculation by pruning unpromising node in each level of the tree and only keep the best features for further branching.	▪ No recovering if pruning decision was wrong, beam search do not guarantee the optimal subset of features as best first search
A Note On Genetic Algorithms For Large-Scale Feature Selection (Siedlecki & Sklansky, 1993)	Genetics algorithm (GA)	Data was provided by the U.S. Army. It consisted of 150 30-dimensional feature vectors, each vector belonging to one of two classes.	The feature subset in GA represented by binary string with length n called chromosome. The chromosome has either 1 or 0 value in a certain position i that determine the presence and absence of a feature i respectively. The fitness of each chromosome is determined whether this chromosome will survive to next generation or discarded.	▪ Showed better results to find the near optimal subset of features compared to all non-exhaustive search methods ▪ Do not guarantee the optimal subset of features

A DE-SVM Feature Selection Model Based on High-Performance Computing (HPC) Technique for P300 Based Brain-Computer Interface (BCI) Data (Alwadei et al., 2018)	Genetics algorithm (GA)	BCI dataset (P300 based system).	Used Genetic Algorithm (GA) and Differential Evolution (DE) as search algorithms while Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) are used as a classifier.	▪ DE-SVM has promising results with an accuracy of 80% selecting 42% of the original features only.
An effective ensemble framework using random forests and a correlation-based feature selection technique (Chutia et al., 2017)	Random Forest (RF)	Different types of images from the Landsat Enhanced Thematic Mapper Plus (Landsat ETM1) and Quick Bird sensors.	To define the sub-optimal subset of features that will improve the learning accuracy and minimize the execution time for RF classifier, they have used Correlation-based feature selection (CFS).	▪ The performance of the RF classifier was noticeably improved while using the optimal set of relevant features compared with a few of the most advanced supervised classifiers like Naïve Bayes and Support Vector Machine.
Pruning State Spaces with Extended Beam Search (Dashti & Wijs, 2007)	Adapted Beam search to the state space generatio n setting	Using a case study Clinical Chemical Analyzer (CCA)	Using a feed-forward network with a back propagation learning method and compared their method with Whitney's feature selection method. They pruned the least salient nodes in the network using node saliency measure that defined as the sum of the increase in error to all the training patterns.	▪ The pruning node method outperforms other classical selection methods because it optimizes both the feature set and the classifier and achieves higher accuracy result (86.3 %) compared to Whitney's accuracy result (81.8%)

Table 1. Comparison Results

4. Conclusion and Future Work

By studying the papers that have been published in this field, it is obvious that many types of research have spotted the lite on feature selection algorithms to improve its performance. Despite that optimal algorithms are guarantee the optimal feature subset, they are computationally cost a lot and in the worst case the algorithms going through all possible subset of features. Therefore, many papers have suggested an adaptive method for more efficient and rapid algorithms but none of them have accelerated the BB algorithm specifically using HPC. From this point of view, as a future work, we are suggesting a fast BB algorithm using HPC and build an optimization feature selection model that can be verified across different dataset types to be applicable for many applications use.

References

Alwadei, S.A. & Dahab, M. & Kamel, M. (2017). A Feature Selection Model based on High-Performance Computing (HPC) Techniques. *International Journal of Computer Applications,* 180(7):11-16.

Atashpaz-Gargari, E., Reis, M. S., Braga-Neto, U. M., Barrera, J., & Dougherty, E. R. (2018). A fast Branch-and-Bound algorithm for U-curve feature selection. *Pattern Recognition*, 73, 172-188.

Chen, X. W. (2003). An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24(12), 1925-1933.

Chutia, D., Bhattacharyya, D. K., Sarma, J., & Raju, P. N. L. (2017). An effective ensemble classification framework using random forests and a correlation based feature selection technique. *Transactions in GIS, 21*(6), 1165-1178.

Dashti, M. T., & Wijs, A. J. (2007, October). Pruning state spaces with extended beam search. In *International Symposium on Automated Technology for Verification and Analysis* (pp. 543-552). Springer, Berlin, Heidelberg.

Gupta, P., Doermann, D., & DeMenthon, D. (2002). Beam search for feature selection in automatic SVM defect classification. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (Vol. 2, pp. 212-215). IEEE.

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, 19(2), 153-158.

Levitin, A. (2012). *Introduction to the design & analysis of algorithms*. Boston: Pearson,.

Liu, H., & Motoda, H. (Eds.). (2007). *Computational methods of feature selection*. CRC Press.

Marill, T., & Green, D. (1963). On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1), 11-17.

Nakariyakul, S., & Casasent, D. P. (2007). Adaptive branch and bound algorithm for selecting optimal features. *Pattern Recognition Letters*, 28(12), 1415-1427.

Narendra, P. M., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*, (9), 917-922.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern recognition letters*, 15(11), 1119-1125.

Reis, M. S. (2012). *Minimization of decomposable in U-shaped curves functions defined on poset chains–algorithms and applications* (Doctoral dissertation, Ph. D. thesis Institute of Mathematics and Statistics, University of Sao Paulo, Brazil.(in Portuguese)).

Ris, M., Barrera, J., & Martins Jr, D. C. (2010). U-curve: A branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices applied to the feature selection problem. *Pattern Recognition*, 43(3), 557-568.

Siedlecki, W., & Sklansky, J. (1993). A note on genetic algorithms for large-scale feature selection. In *Handbook of Pattern Recognition and Computer Vision* (pp. 88-107).

Somol, P., Pudil, P., & Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7), 900-912.

Strearns, S. D. (1976). On selecting features for pattern classifiers. *Proc. ICPR*, 1976, 71-75.

Whitney, A. W. (1971). A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9), 1100-1103.

Xu, L., Yan, P., & Chang, T. (1988, November). Best first strategy for feature selection. In *Pattern Recognition, 1988., 9th International Conference on* (pp. 706-708). IEEE.

Yu, B. I. N., & Yuan, B. (1993). A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, 26(6), 883-889.