IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 22, 2018, accepted November 22, 2018, date of publication November 27, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883537

Feature Selection for High Dimensional Data Using Monte Carlo Tree Search

MUHAMMAD UMAR CHAUDHRY AND JEE-HYONG LEE
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jee-Hyong Lee (john@ skku.edu)

This work was supported by the National Research Foundation of Korea, Ministry of Science, ICT, through the Next-Generation Information Computing Development Program, under Grant NRF-2017M3C4A7069440.

ABSTRACT Feature selection is the preliminary step in machine learning and data mining. It identifies the most important and relevant features within a dataset by eliminating the redundant or irrelevant features. The substantial benefits may include an improved performance in terms of high prediction accuracy, reduced computational complexity, and simply interpretable underlying models. In this paper, we present a novel framework to investigate and understand the importance of Monte Carlo tree search (MCTS) in feature selection for very high-dimensional datasets. We construct a binary feature selection tree where each node represents one of the two feature states: a feature is selected or not. The search starts with an empty root node reflecting that no feature is selected. Then, the search tree is expanded by adding nodes in an incremental fashion through MCTS-based simulations. Following tree and default policy, every iteration generates an initial feature subset, where a filter is used to select the top k features forming the candidate feature subset. The classification accuracy is used as the goodness or reward of the candidate feature subset and propagated backward up to the root node following the active path. Finally, the candidate subset with highest reward is selected as the best feature subset. Experiments are performed on 30 real-world datasets, including 14 very high-dimensional microarray datasets, and results are also compared with state-of-the-art methods in the literature, which proves the efficacy, validity, and significance of the proposed method.

INDEX TERMS Dimensionality reduction, feature selection, filter-wrapper, hybrid, Monte Carlo tree search (MCTS), H-MOTiFS.

I. INTRODUCTION

In the present era of big data, most of the datasets are high dimensional ranging from few hundreds to thousands of features. A substantial amount of features are either redundant or irrelevant which makes the underlying model very complex and degrade the performance of the prediction task [1]–[3]. For predictive modeling, achieving high accuracy within acceptable amount of time is of prime importance. Feature selection then becomes handy as a preliminary step before any predictive task. The objective of a feature selection algorithm is to find the most significant features while maintaining the underlying structure of the dataset. This helps in making better predictive models in terms of performance by achieving high accuracy and reduced time complexity. Also the underlying structure becomes trivial to interpret and analyze. Feature selection is widely being applied in various application domains relating to machine learning [4]–[6], pattern recognition [7]–[9] and data mining [10], [11].

Various approaches for feature selection have been developed in the past decades [12]. Generally feature selection consists of two main components: a *search component* and an *evaluator component*, as depicted in Fig.1. A *search component* is responsible for generating candidate features subsets whereas *evaluator component* checks the goodness of the candidate subsets. Based on *evaluator component*, feature selection methods are generally categorized as filter, wrapper or hybrid methods. Considering search component, feature selection approaches are categorized as exhaustive, heuristic or meta-heuristic approaches. The details are provided in Section 2.

Recently, meta-heuristic approaches have gained much attention in feature selection domain. They are also referred as Evolutionary Algorithms. Meta-heuristic algorithms combine the exploitation of good solutions with the exploration of new ones, thus, trying to reach the optimum solutions. Genetic algorithms (GA) [13], [14], Particle Swarm

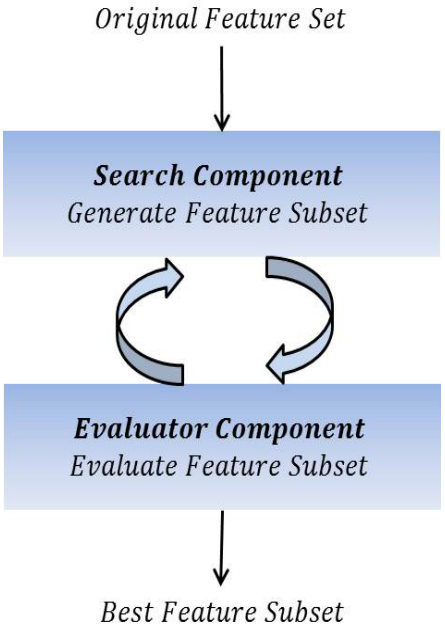

FIGURE 1. Key components of feature selection.

Optimization (PSO) [15]–[18], Bat Algorithms (BA) [19], [20], Ant Colony Optimization (ACO) [21], [22] and Multi-Objective Evolutionary Algorithms [23], [24] belong to this category. The use of these meta-heuristic approaches have opened up new horizons in feature selection, however, they are in infancy phases and need more thorough investigations and research. The one shortcoming which is common among all such algorithms is that they exhibit complex nature: how they deal with the meta-heuristics, the high enough time requirement for convergence and induction of many hyper-parameters [25]. Thus, a vast room for research exists and new feature selection approaches are immensely needed which can be as *efficient* as *accurate* along with *least complex* to model.

In this study, we come up with a novel framework where Monte Carlo Tree Search (MCTS) is deployed in conjunction with the hybrid of filter-wrapper methods to find the best feature subset for the efficient classification of the dataset. We referred the proposed algorithm as H-MOTiFS (Hybrid-Monte Carlo Tree search based Feature Selection). The term MCTS is referred to a heuristic search technique which is recently evolved in gaming AI and showed remarkable performance in huge search spaces [26]. It uses lightweight random simulations to find the best solutions [27]. The success of MCTS in gaming domain provoked us to investigate its effectiveness in feature selection and acted as a catalyst for this study.

The search starts with an empty root node reflecting that no feature is selected. As the search proceeds, nodes are added one by one reflecting one of the two feature states: either a feature is selected or not. Every iteration leads to a feature subset consisting of arbitrary number of features, referred as *initial feature subset*, based on tree search and random sampling. A filter is then applied on *initial feature subset* to select the top k features forming the *candidate feature subset*. The goodness of the *candidate feature subset* is measured in terms of the classification accuracy. It also serves as the reward of the current active path. This reward is propagated backwards up to the root node following the active path and the tree is updated. Lastly, the candidate feature subset holding the maximum classification accuracy is selected as the *best feature subset*. Experiments are performed on *30* different datasets, where *14* datasets are very high dimensional microarray datasets. The results are compared with the established and state-of-the-art approaches in literature. The promising results show the effectiveness and superiority of the proposed method. The key highlights of this study are as follows:

- The significance of MCTS is investigated and a novel hybrid framework is proposed for feature selection in very high dimensional datasets, referred as H-MOTiFS.
- H-MOTiFS performs the efficient exploration of the feature space and finds the top k best features in a limited number of iterations, relatively.
- H-MOTiFS is simple and flexible as compared to other complex evolutionary approaches. Only three hyper-parameters are involved namely: *Scaling factor*, *top-k selector* and *Termination criteria*.
- H-MOTiFS is experimented on *30* publically available benchmark datasets including *14* very high dimensional microarray datasets. The comparison with the state-of-the-art approaches established the significance of the proposed method.

The rest of the paper is organized as follows. In Section 2, a preliminary background is provided. Section 3 provides the details and explanation of the proposed method. The experimental details and results are discussed in Section 4. Finally, the summarized conclusions are provided in Section 5.

II. BACKGROUND

A. OVERVIEW OF EXISTING FEATURE SELECTION APPROACHES

The key components and basic feature selection process is illustrated in Fig. 1 above. The brief overview of literature is provided in the following text.

Considering the evaluation criterion, the literature divides the feature selection methods as filter, wrapper or hybrid methods. Filter based methods are classifier independent and use some proxy measure to evaluate the features. Using correlation of features with class variable based on various statistical tests, features are evaluated and ranked [28]–[30]. Filter methods are fast enough and can be generalized with any classifier, however, they lack in performance in the presence of redundant features and show low classification accuracy. Various information theoretic based approaches have been developed to overcome the issues with traditional filter based methods [31]–[33]. Wrapper methods are classifier dependent and use the classifier directly to score the

feature subsets. The classification accuracy serves as a scoring metric to measure the goodness of feature subsets [34]–[36]. They deliver high classification accuracy as the specific classifier is directly involved in the process. However, the major drawbacks associated are high computational complexity and prone to over fitting. Hybrid approaches combing filter and wrapper methods gain much attention in recent literature [37], [38]. They take advantage of both methods by using an independent metric to rank the features and using a learning algorithm to quantify the strength of feature subsets. Our focus is on the hybrid of filter-wrapper method in this study because of their superiority over filter or wrapper methods.

Irrespective of the evaluator used, an efficient search strategy is mandatory as it is practically impossible to exhaustively check the goodness of each and every possible subset (2^n subsets for n number of features). Considering this fact, researchers developed and adopted several heuristic approaches. Among the heuristic methods best first search and greedy hill climbing are commonly used in literature [28], [39]. Sequential Forward Selection (SFS), Sequential Backward Selection (SBF) and bi-directional approaches are categorized as greedy hill climbing approaches. They assess the local changes in the search space in order to find the dominant features. The major shortcoming associated is referred as the nesting effect. Whenever a potential change occurs in the candidate feature subset; a feature is included in the candidate subset (i.e. in SFS) or eliminated from the candidate subset (i.e. in SBF) once, then this particular feature is never re-evaluated and the search becomes highly prone towards local optimum.

To tackle the issues in traditional heuristic approaches, researchers attempted to use meta-heuristic approaches in feature selection. Meta-heuristic algorithms combine the exploitation of good solutions with the exploration of new ones, thus, trying to reach the optimum solutions. Most dominant approaches include Genetic algorithms (GA) [13], [14], Particle Swarm Optimization (PSO) [15]–[18], Bat Algorithms (BA) [19], [20], Ant Colony Optimization (ACO) [21], [22] and Multi-Objective Evolutionary Algorithms [23], [24]. These approaches have shown decent performance in feature selection, however, they suffer from the problem of *parameters overload*. That is, excessive hyper-parameters are involved and it becomes complex to tune the model for optimized performance [25]. For instance, in Genetic Algorithm based approaches, high enough generations with large population size is essential to achieve the required results. Thus, GA based approaches tend to be computationally expensive. Moreover, the induction of many hyper-parameters like no. of generations, population size, permutation and crossover probabilities, etc. makes it highly complex and challenging to fine tune the model for desired results.

There are some recent researches which used MCTS for feature selection. FUSE algorithm formalized the feature selection as a reinforcement learning problem and employed MCTS for approximating the optimal policy [40]. The algorithm used the exhaustive search tree where the state space is exponential in the number of features. Various heuristics were tried to tackle the challenging issue of the huge branching factor. Another algorithm, FSTD, used the temporal difference to traverse the huge state space and selected the best features subset [41]. Ashtiani *et al.* [42] have proposed a strategy for local feature subset selection. The algorithm simultaneously partitions the sample space into localities and selects features for them, and uses MCTS to learn near-optimal feature trees. Recently, Chaudhry and Lee [43] have proposed a novel binary feature selection tree with less branching factor and developed a wrapper based approach, MOTiFS, which used MCTS to select the optimal features subset. The algorithm showed promising classification performance for relatively small dimensional datasets. However, it tends to select high proportion of features, relatively. One possible reason might be the selection of some noisy features because of the impact of randomness. This problem may become worse when dealing with very high dimensional datasets within the allowed number of simulations.

The main contribution of this work is to propose a novel framework based on MCTS, to deal with very high dimensional datasets with an objective to achieve high accuracy with reduced dimensions. In our proposed framework, MCTS is deployed in conjunction with the hybrid of filter-wrapper methods. We construct a binary feature selection tree where the exploitation and exploration are properly balanced, and the induction of the filter aids in removing the noisy features from the features subset during each simulation. The classifier is then used to evaluate the candidate features subset. Experiments are performed extensively on many high dimensional benchmark datasets. The comparison with various methods proves the significance of the proposed method.

B. MONTE CARLO TREE SEARCH (MCTS)

The term MCTS is referred to a search technique which is recently evolved in gaming AI and showed remarkable performance in huge search spaces [26]. It performs random lightweight simulations in order to find optimal solutions [27]. Each MCTS simulation constitutes four ordered steps, namely, Selection, Expansion, Simulation and Backpropagation. During Selection, a search tree is traversed from root node to a node which is non-terminal and has unexpanded children. The nodes are selected with highest approximated value based on UCT algorithm. A new child node is then added to expand the tree (according to the actions available) during Expansion step. During Simulation, a random simulation is run from the new child node until the terminal node is reached. The simulation reward is also approximated at this stage. Finally, the simulation reward is backpropagated through the selected nodes to update the tree. The *tree policy* is used to perform the *selection* and *expansion* steps, whereas *default policy* is responsible for the *simulation* step.

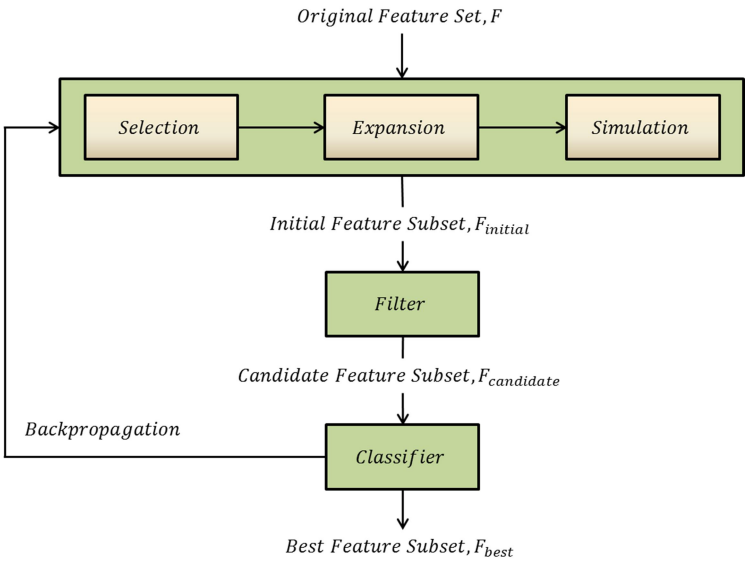

FIGURE 2. Graphical illustration of the proposed method.

C. UPPER CONFIDENCE BOUNDS FOR TREES (UCT) ALGORITHM

The nodes selection is performed by the *tree* policy during each simulation. The *tree policy* uses the UCT algorithm to rank the competing nodes at each level. The importance of each node is approximated using Equation (1). The node with the highest approximated value is selected at each level. It keeps balancing between exploring new solutions and exploiting good ones.

$$UCT_v = \frac{W_v}{N_v} + C \times \sqrt{\frac{2 \times ln\left(N_p\right)}{N_v}} \tag{1}$$

Where, N_v and N_p denotes the number of visits performed on nodes v and its parent p, respectively. C is the constant to balance between exploration and exploitation. W_v holds the count of wining simulations (in gaming context) at node v.

III. H-MOTiFS (HYBRID-MONTE CARLO TREE SEARCH BASED FEATURE SELECTION)

We propose a novel framework based on MCTS, to deal with very high dimensional datasets with an objective to achieve high accuracy with reduced dimensions. In our proposed framework, MCTS is deployed in conjunction with the hybrid of filter-wrapper methods. We construct a binary feature selection tree where the exploitation and exploration are properly balanced, and the induction of the filter aids in removing the noisy features from the features subset during each simulation. The classifier is then used to evaluate the candidate features subset. The details are provided below.

In our proposed framework, we search the feature space using MCTS and evaluate the feature subsets in a hybrid setting. We traverse the feature selection tree using MCTS to look for the best path (constitutes best performing features) in order to select the best feature subset. Following *tree* and *default* policies, each MCTS iteration generates an initial feature subset, $F_{initial}$. The filter is then applied to select the top k features forming the candidate feature subset, $F_{candidate}$. The goodness of $F_{candidate}$ is measured in terms of classification accuracy according to the classifier applied. Then, the search tree is updated through the active path. This procedure repeats until the stopping criterion is met. Fig. 2 shows the graphical illustration of the proposed method.

A. FEATURE SELECTION TREE

We assume the feature selection as a single player game tree where one has to pick the best performing nodes (features) having maximum accumulative reward. Each node represents one of the two corresponding feature states: a feature is selected or not selected.

Definition 1: For a feature set, $F = \{f_1, f_2, \ldots, f_i, \ldots, f_n\}$, *the feature selection tree is a tree satisfying the following conditions:*

1) *The root is* $\emptyset_0$, *which represents no feature is selected.*
2) *Any node at level* $i-1$ *has two children,* f_i *and* $\emptyset_i$, *where* $0 < i < n$.

In the feature selection tree, node f_i represents feature f_i is selected and $\emptyset_i$ represents feature f_i is not selected. Any path from the root node to one of the leaves constitutes a feature subset. So, the objective is to find a path that offers the best reward (i.e. accuracy). The algorithm traverses the feature selection tree using MCTS and selects one of the paths.

The search starts with an empty root node reflecting that no feature is selected. The nodes are then added in an incremental fashion during each simulation. Following *tree* and *default* policy, every iteration generates an initial feature subset, $F_{initial}$, where a filter is applied to select the top k features forming the candidate feature subset, $F_{candidate}$. The classifier is then applied on $F_{candidate}$ for evaluation.

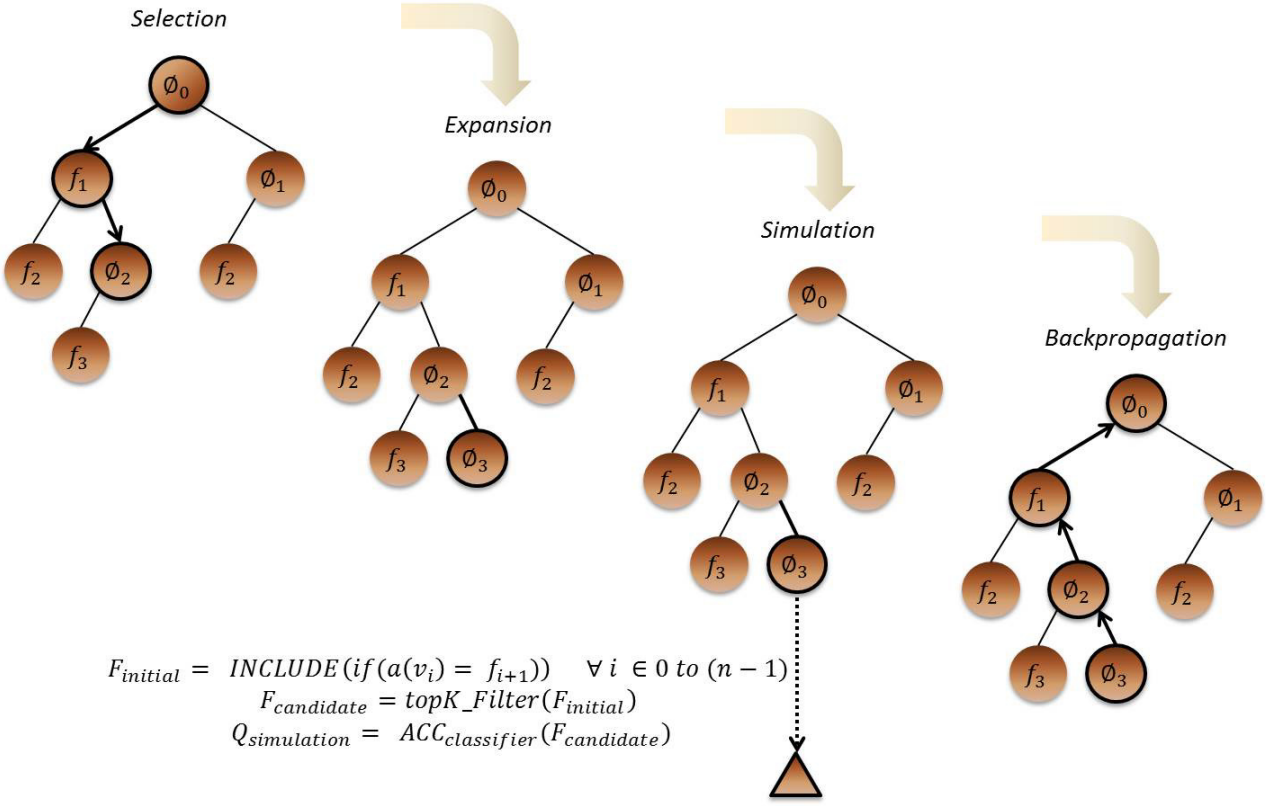

FIGURE 3. Feature selection tree and the search procedure.

The classification accuracy is used as the goodness of the candidate subset. It also serves as a reward of the current simulation and propagated backwards up to the root node following the active path. Finally, $F_{candidate}$ with maximum accuracy is selected as the best feature subset, F_{best}. The proposed feature selection tree and the four step MCTS procedure are shown in Fig. 3.

B. SELECTION

During selection, the algorithm traverses the already expanded tree and selects one of the possible paths. The selected path constitutes on nodes whose inclusion used to give the high reward in previous iterations. The features in the selected path are added in the initial feature subset, $F_{initial}$, of the current iteration.

Following *tree* policy, the *UCT* algorithm traverses the already expanded tree, from the root node to the urgent node (a non-terminal node with a child to expand). At each tree level, it selects the node which gives high score obtained by using (2). If the selected node is f_i at level i, it provides the intuition that feature f_i contributed well towards high rewards in previous iterations, hence, feature f_i is added in the feature subset, $F_{initial}$. On the other hand, if UCT algorithm selects the node $\emptyset_i$, feature f_i is not added in the $F_{initial}$. The intuition is, feature f_i did not contribute towards high reward and needs not to be added in the current feature subset.

The vanilla UCT algorithm is best suitable for gaming scenarios where the reward is binary; either a player wins the game or loses it. One has to find the best wining moves in the least number of visits. Therefore, the reward at each node is estimated by penalizing with the number of visits as shown in (1). The nature of feature selection problem is, however, different where the core objective is to find the path that holds the highest reward (i.e. accuracy in our case). This intuition leads us to use the maximum reward achieved at each node, as shown in (2).

$$UCT_{v_j} = \max(Q_{v_j}) + C \times \sqrt{\frac{2 \times ln\left(N_{v_i}\right)}{N_{v_j}}} \tag{2}$$

Where, $\max(Q_{v_j})$ represents the maximum reward at the node v_j. $C > 0$ is an exploration constant. N_{v_j} and N_{v_i} counts the number visits performed on nodes v_j and its parent v_i, respectively. During tree traversal, the nodes with the highest scores are selected at each tree level.

C. EXPANSION

A search tree is expanded by adding a new child node. The child is added to the urgent node (the last node selected during selection). Again, the decision is based on the UCT score. The child with the highest score is added in the search tree. Let's assume the urgent node is v_i at some moment in time. If the algorithm decides to add the child f_{i+1} in the search

tree, then the feature f_{i+1} is also included in the initial feature subset, $F_{initial}$. Conversely, the tree is expanded by adding child node $\emptyset_{i+1}$ and feature f_{i+1} is not included in $F_{initial}$.

D. SIMULATION

Simulation step prompts randomness in the process and is controlled by the *default* policy. It randomly selects a path from expanded child, say v_i, to the leaf node v_n. Hence, the remaining unexpanded features are included or excluded in the current feature subset, $F_{initial}$, with uniform probability.

Let's assume, the expanded child node is v_i in the current iteration. Features from f_1 to f_i are included in initial feature subset, $F_{initial}$, during the selection and expansion steps (*tree policy*). However, the remaining features from f_{i+1} to f_n are randomly included based on simulation step (*default policy*). A tree search is supported by the random sampling in the generation of a feature subset. It provides the fair chance to find the best feature subset in a fewer simulations, rather full expansion of the tree.

E. CANDIDATE FEATURE SUBSET GENERATION

Each iteration generates a feature subset which we referred above as initial feature subset, $F_{initial}$. Due to fully random nature of *default* policy, it is highly probable that high proportion of features can be selected in $F_{initial}$. Along with arbitrary feature dimensions, it may induce the inclusion of noisy features in $F_{initial}$. To overcome the issue, we deploy the filter on $F_{initial}$ which returns the top k features forming the candidate feature subset, $F_{candidate}$, as shown in (3).

$$F_{candidate} = topK\_Filter\ (F_{initial}) \tag{3}$$

The induction of filter at this stage has two advantages. First, it helps in removing the noisy features from random selection. Secondly, it aids the objective function to control the selected feature dimensions along with achieving high accuracy.

F. REWARD CALCULATION AND BACKPROPAGATION

At this step, classifier is applied on candidate feature subset, $F_{candidate}$. The classification accuracy on $F_{candidate}$ is used as the simulation reward, $Q_{simulation}$, of the current expanded node. The search tree is updated by propagating the reward backwards, following the active path.

$$Q_{simulation} = ACC_{classifier}\ (F_{candidate})$$

Where, $ACC_{classifier}\ (F_{candidate})$ is the classification accuracy on $F_{candidate}$. If $F_{candidate}$ gives the improved accuracy then the best feature subset is updated. The process continues until stopping criteria is met.

For experimental purposes, we employed *K-NN* classifier for reward (accuracy) calculation and *ReliefF* as a filter measure to rank the features according to class labels. *K-NN* is characterized as a non-parametric, simplest, and efficient learning algorithm and proved its significance in many similar studies [44]–[46]. ReliefF evaluates the importance of a feature by continuously sampling the instances to separate from the nearest neighbors (neighbors from the same and different classes) and assigning a weight according to the ability of separation [47]–[49]. However, any other classifiers and filter measures can be used within the proposed framework. The algorithm of H-MOTiFS is provided below as Algorithm 1.

TABLE 1. Summary of the small dimensional datasets

#	Dataset	# Features	# Instance	# Classes
1	Spambase	57	4701	2
2	Ionosphere	34	351	2
3	Arrhythmia	195	452	16
4	Multiple ft.	649	2000	10
5	Waveform	40	5000	3
6	WBDC	30	569	2
7	German no.	24	1000	2
8	DNA	180	2000	2
9	Sonar	60	208	2
10	Hillvalley	100	606	2
11	Musk 1	166	476	2
12	Coil20	1024	1440	20
13	Orl	1024	400	40
14	Lung_Discrete	325	73	7
15	Kr-vs-kp	36	3196	2
16	Spect	22	267	2

IV. EXPERIMENT AND RESULTS

This section provides the details of extensive experimentation and comparative analysis on various benchmark datasets.

A. DATASETS

We use *30* benchmark datasets from various domains. Purposefully, we divide the datasets into two broad categories; the small dimensional datasets and the microarray datasets (also referred as a very high dimensional datasets because of thousands of feature dimensions). We considered it inappropriate to experiment on very small dimensional datasets (where the number of features are less than 20) and discarded from the scope.

In our experiments, 16 datasets are small dimensional datasets and downloaded from UCI [50] and Lib-SVM [51]. The details are provided in Table 1.

For very high dimensional datasets, we experimented on 14 microarray (gene selection) datasets. The datasets are downloaded from [52] and [53]. The details of datasets are summarized in Table 2.

B. EXPERIMENTAL SETUP AND PARAMETERS SETTING

We perform 10 fold cross validation where 1 fold is used as a test set and remaining 9 folds are used for the training and validation purpose. Thus, each fold is used exactly once as a test set.

The hyper-parameters values used in the experiments are given in Table 3. We fix the *Scaling factor, C,* at 0.1 for all the datasets. The other two parameters, *Termination criteria (# simulations)* and the *top-k* filter are used in two different settings. We empirically set the *# simulations* at 1000 and 10,000 for the small dimensional datasets and microarray

Algorithm 1 The Proposed feature Selection Algorithm

```
Load dataset and preprocess
Initialize SCALAR, BUDGET, k
function FEATURE_SELECTION (featuresList)
    create rootNode
    maxReward, bestFeatureSubset ← UCTSEARCH (rootNode)
    return(maxReward, bestFeatureSubset)
function REWARD (featureSubset)
    candidateFeatureSubset← FILTER(featureSubset, k)
    simulation_reward ← CLASSIFIER(candidateFeatureSubset)
    return(simulation_reward)
function UCTSEARCH (rootNode)
    Initialize maxReward, bestFeatureSubset
    while within computational budget do
        frontNode ← TREEPOLICY (rootNode)
        reward, candidateFeatureSubset← DEFAULTPOLICY (frontNode.state)
        BACKUP (frontNode, reward)
        if reward is greater than maxReward then
            maxReward←reward
            bestFeatureSubset←candidateFeatureSubset
    return(maxReward, bestFeatureSubset)
function TREEPOLICY (node)
    while node is non-terminal do
        if node not fully expanded then
            return EXPAND (node)
        else
            node ← BESTCHILD (node, SCALAR)
    return node
function EXPAND (node)
    choose a ∈ untried actions from A(node.state)
    add a newChild with f (node.state, a)
    return newChild
function BESTCHILD (v, C)
    return max (Q_v') + C sqrt( 2×ln(v.visits) / v'.visits )
function DEFAULTPOLICY (state)
    while state is non-terminal do
        choose a ∈ A(state) uniformly at random
        state← f (state, a)
    traverse state.path
        if a_i is equal to f_{i+1} then
            featureSubset ← INCLUDE (f_{i+1})
    reward← REWARD (featureSubset)
    return(reward, featureSubset)
function BACKUP (node, reward)
    while node is not null do
        node.visits←node.visits + 1
        if reward > node.reward then
            node.reward←reward
        node←node.parent
    return
```

datasets, respectively. For small dimensional datasets, we are intended to select and analyze the top 10%, 20% and 30% of the total number of features (*# F*). However, microarray datasets are different in nature and a very small proportion of genes are relevant but not redundant. Therefore, we limit *top-k* to select top (10, 20, 30) genes only.

TABLE 2. Summary of very high dimensional (microarray) datasets.

#	Dataset	# Features	# Instance	# Classes
1	Colon	2000	62	2
2	Leukemia 1	7129	72	2
3	Lymphoma	4026	62	3
4	SRBCT	2308	82	4
5	Breast 2	4869	77	2
6	Prostate	6033	102	2
7	Leukemia 2	3051	38	2
8	Brain	5597	42	5
9	Adenocarcinoma	9868	76	2
10	Breast 3	4869	95	3
11	NCI	5244	61	8
12	9 Tumors	5726	60	9
13	DLBCL	7070	77	2
14	CNS	7129	60	2

TABLE 3. Parameters values used in the experiments.

Data Type	*C*	*#simulations*	*top-k filter*
Small	0.1	1000	(*k* / *#F*) %
Microarray	0.1	10,000	*k*

**k = (10,20,30)*

TABLE 4. Methods used for comparison

Method	Description
SFS, SBS	Traditional Sequential Approaches
MOTiFS	A wrapper approach based on MCTS [43]
IWSSr	Incremental Wrapper-Based Subset Selection with replacement [54]
SFSW	An Evolutionary Multi-Objective Approach [23]
PSO (4-2)	PSO with Novel initialization and Update [18]
RF	Random Forest Based Approach [55]
FS-FS	Feature Similarity Technique [56]
FR-FS	Fuzzy Rule Based Method [57]
DEMOFS	Differential Evolution based Multi-Objective Approach[23]
PSO	Particle Swarm optimization
ENORA	Multi-Objective Evolutionary Method [58]
SRKNN	Ensemble of Sequential Random KNN [44]
HPSO-LS	PSO Based Hybrid Approach [59]
ACO	Ant Colony Optimization [59]
E-FSGA	Ensemble approach using bi-objective GA [60]
WOA, WOA-T	Whale Optimization Approaches [61]

C. METHODS USED FOR COMPARISON

We compare our results with significant and state-of-the-art methods in literature. While choosing the comparison methods, we try to maintain the diversity and quality of the works reported. The comparison methods include the traditional wrapper and filter based approaches, multi-objective and hybrid methods, evolutionary approaches and random forests based approach. The details and references are provided in Table 4.

D. RESULTS AND COMPARISONS

For every dataset, we evaluate our model five times and report the mean results. We report the mean accuracy with standard deviation and the number of features selected.

TABLE 5. Mean accuracy and standard deviation on small dimensional datasets at different values of top-*k* features. Bold face values in each row indicate the best result.

Dataset	H-MOTiFS		
	top-k = (10 / #F)%	*top-k = (20 / #F)%*	*top-k = (30 / #F)%*
Spambase	0.877±0.002	0.901±0.002	**0.907**±0.001
Ionosphere	0.870±0.009	**0.892**±0.013	0.887±0.007
Arrhythmia	0.630±0.007	**0.640**±0.007	0.630±0.009
Multiple ft.	0.969±0.003	0.976±0.002	**0.983**±0.003
Waveform	0.750±0.004	0.812±0.005	**0.823**±0.005
WBDC	0.926±0.005	**0.964**±0.003	0.961±0.002
German no.	0.717±0.001	0.705±0.007	**0.728**±0.014
DNA	**0.905**±0.004	0.889±0.003	0.881±0.004
Sonar	0.767±0.012	0.830±0.006	**0.836**±0.004
Hillvalley	**0.566**±0.004	0.561±0.006	0.559±0.008
Musk 1	0.789±0.009	0.832±0.006	**0.850**±0.01
Coil20	0.962±0.008	0.985±0.007	**0.989**±0.007
Orl	0.835±0.012	0.855±0.010	**0.883**±0.009
Lung_discrete	0.769±0.037	0.809±0.013	**0.823**±0.006
Kr-vs-kp	0.941±0.002	**0.976**±0.004	0.962±0.005
Spect	0.767±0.008	0.786±0.008	**0.817±0.009**

1) SMALL DIMENSIONAL DATASETS

Small dimensional datasets are evaluated using the *5-NN* classifier as used by the baseline methods; SFSW [23] and PSO(4-2) [18] for the fair comparison.

Table 5 shows the results obtained at different values of *top-k* filter. Experiments are performed independently to select the top 10%, 20% and 30% features in the whole feature space. Ten datasets ''Spambase'', ''Multiple ft.'', ''Waveform'', ''German no'', ''Sonar'', ''Musk 1'', ''Coil20'', ''Orl'', ''Lung_Discrete'', and ''Spect'' show their best when top 30% features are selected. Four datasets ''Ionosphere'', ''Arrhythmia'', ''WBDC'', and ''Kr-vs-kp'' are best classified when top 20% features are selected. However, ''DNA'' and ''Hillvalley'' are well classified at top 10% features. This diversity indicates the significance and impact of the *top-k* selector component. The standard deviation shows the stability of the proposed method.

The comparison of the proposed method with other approaches on *16* small dimensional datasets is provided in Table 6. For the summarized comparisons we pick the best results (in terms of mean accuracy) from Table 5 against each dataset. Comparing with MOTiFS in terms of classification accuracy, our proposed method outperforms on 9 out of *14* datasets and stands equal on *1* dataset. The superiority of the proposed method is clearly evident in terms of the number of selected features. Our proposed method selects very less number of features with improved or nearly equivalent classification performance on all the datasets. The results of relatively large dimensional datasets, ''Multiple ft.'', ''DNA'', ''Hillvalley'', ''Musk 1'', ''Coil20'' and ''Orl'' are worth mentioning where our proposed method selects very small number of features as compared to MOTiFS with improved accuracy.

Comparing with SFSW, our proposed method outperforms on *8* out of *11* datasets in terms of both the classification accuracy and the dimensional reduction. Comparing with E-FSGA, our proposed method overtakes on *6* out

TABLE 6. Comparison (w.r.t Avg. Accuracy & no. of Sel. Feat) of H-MOTiFS with other methods, for small dimensional datasets. Best results in each row are bold and underlined. The second best results in each row are in bold face. "-" are placed wherever information is not available.

Dataset	Avg. Acc. (# Sel. Feat.)											No. Feat. Sel.
	H-MOTiFS	MOTiFS [43]	SFSW [23]	SFS [23]	SBS [23]	FS-FS [56]	FR-FS [57]	E-FSGA [60]	WOA [61]	WOA-T [61]	PSO (4-2) [18]	DEMOFS [23]	
Spambase	**0.907** (18)	**0.907** (31.5)	0.885 (26)	0.874 (35.7)	0.870 (37.3)	0.900 (29)	-	**0.922**	-	-	-	-	0.903
Ionosphere	**0.892** (7)	**0.889** (12.3)	0.883 (11.5)	0.887 (1.2)	0.859 (9.1)	0.788 (16)	0.844 (4.3)	0.862	0.890 (21.45)	0.884 (20.20)	0.873 (3.3)	-	0.849
Arrhythmia	0.640 (40)	**0.650** (94.4)	**0.658** (100)	0.599 (89.4)	0.580 (49.2)	0.589 (100)	-	-	-	-	-	-	0.615
Multiple ft.	**0.983** (195)	**0.980** (321.8)	0.979 (270)	0.903 (210)	0.912 (305)	0.783 (325)	-	0.945	-	-	-	-	0.978
Waveform	**0.823** (12)	0.816 (19.42)	**0.837** (16)	0.778 (18.4)	0.785 (18.3)	0.752 (20)	-	-	0.713 (33.2)	0.710 (33.72)	-	-	0.789
WBDC	0.964 (6)	**0.967** (15.4)	0.941 (13.5)	0.901 (13.9)	0.898 (17.8)	-	0.936 (2.1)	**0.969**	0.955 (20.76)	0.950 (20.55)	0.940 (3.5)	-	0.970
German no.	**0.728** (8)	**0.725** (11.5)	0.713 (10.5)	0.682 (12.2)	0.658 (10.8)	-	-	-	-	-	0.685 (12.8)	0.701 (1)	0.724
DNA	**0.905** (18)	0.810 (89.3)	**0.831** (71.8)	0.822 (18.8)	0.823 (20.6)	-	-	-	-	-	-	-	0.829
Sonar	0.836 (12)	0.850 (28.9)	0.827 (20)	0.808* (12)	-	-	0.729 (5.8)	0.808	**0.854** (43.38)	**0.861** (38.22)	0.782 (11.2)	0.786 (10)	0.846
Hillvalley	0.566 (10)	0.535 (45.2)	0.575 (40)	0.563* (10)	-	-	-	-	-	-	**0.578** (12.2)	**0.605** (26)	0.531
Musk 1	**0.850** (50)	**0.852** (81.3)	0.815 (59.3)	0.838* (14)	-	-	-	-	-	-	0.849 (76.5)	0.835 (58)	0.832
Coil20	**0.989** (308)	**0.980** (505.4)	-	-	-	-	-	0.892	-	-	-	-	0.976
Orl	**0.883** (308)	**0.862** (498.3)	-	0.835* (23)	-	-	-	0.622	-	-	-	-	0.882
Lung_Discre	**0.823** (98)	**0.810** (154.82)	-	0.795* (24)	-	-	-	0.713	-	-	-	-	0.836
Kr-vs-kp	**0.975** (8)	0.961 (20.1)	-	**0.973** (12)	-	-	-	-	0.915 (27.90)	0.896 (26.71)	-	-	0.960
Spect	**0.817** (7)	**0.809** (10.28)	-	0.794* (4)	-	-	-	-	0.788 (12.10)	0.792 (11.51)	-	-	0.809

*Evaluated using Weka library

of *8* datasets in terms of accuracy. It shows the dominance of H-MOTiFS over GA based approaches where the tuning of plenty of parameters is a huge challenge for the optimized performance, as the number of features grows. H-MOTiFS shows the best accuracy on *5* out of *6* datasets in comparison with both WOA and WOA-T. Moreover, H-MOTiFS selects a very less number of features than WOA and WOA-T for all the datasets. Comparing with all other approaches, SFS, SBS, FS-FS, FR-FS, PSO(4-2) and DEMOFS, our proposed method outperforms on all datasets, except 1 dataset "Hillvalley" where PSO(4-2) and DEMOFS show better classification performance.

Overall summarizing Table 6, our proposed method, H-MOTiFS, shows the best performance on 9 datasets, "Ionosphere", "Multiple ft.", "German no.", "DNA", "Coil20", "Orl", "Lung_Discrete", "Kr-vs-kp", and "Spect". On 3 datasets, "Spambase", "Waveform", and "Musk 1", our proposed method stands the 2nd best among the list. On 4 datasets "Arrhythmia", "WBDC", "Sonar", and "Hillvalley", our proposed method takes the 3rd position or less. Collectively, H-MOTiFS shows the outstanding performance in terms of accuracy and dimension reduction. Specially, for "Multiple feat.", "DNA", "Coil20", and "Orl" datasets (relatively large dimensional datasets), the results are worth mentioning where H-MOTiFS selects very small number of features with highly improved classification accuracy as compared to all other methods. The last column in Table 6 shows the accuracy results without feature selection.

2) HIGH DIMENSIONAL DATASETS

For microarray datasets, we use *1-NN* classifier for evaluation as reported in the random forests based method [55] and SRKNN [44]. We select the top 10, 20, and 30 genes and report the average results in Table 7, along with the standard deviation of 5 independent runs. Two datasets, "Prostate" and "DLBCL", show their best when top *10* features are selected. Six datasets, "Colon", "Lymphoma", "Breast 2", "Leukemia 2", "Adenocarcinoma", and "CNS", are the best at top *k=20*. Whereas, six datasets, namely "Leukemia 1", "SRBCT", "Brain", "Breast 3", "NCI", and "9 Tumors" perform the best at top *k=30*. This indicates the significance of *top-k* selector component.

TABLE 7. Mean accuracy and standard deviation on microarray datasets at different values of top-*k* features. Bold face values in each row indicate the best result.

Dataset	H-MOTiFS		
	top-k = 10	*top-k = 20*	*top-k = 30*
Colon	0.807±0.022	**0.816**±0.021	0.810±0.026
Leukemia 1	0.920±0.007	0.892±0.018	**0.930**±0.012
Lymphoma	0.950±0.012	**0.987**±0.011	0.983±0.012
SRBCT	0.951±0.028	0.960±0.023	**0.983**±0.016
Breast 2	0.613±0.026	**0.665**±0.052	0.657±0.053
Prostate	**0.904**±0.025	0.896±0.006	0.886±0.010
Leukemia 2	0.900±0.022	**0.963**±0.021	0.960±0.034
Brain	0.691±0.043	0.735±0.039	**0.792**±0.016
Adenocarcinom	0.818±0.002	**0.822**±0.006	0.806±0.001
Breast 3	0.510±0.025	0.515±0.037	**0.539**±0.043
NCI	0.598±0.042	0.610±0.020	**0.675**±0.029
9 Tumors	0.433±0.024	0.477±0.030	**0.521**±0.041
DLBCL	**0.963**±0.030	0.930±0.028	0.934±0.020
CNS	0.600±0.050	**0.610**±0.022	0.592±0.030

Comparing with ensemble based approaches, SRKNN [44], our method shows the best performance on *8* out of *11* datasets namely, ''Lymphoma'', ''SRBCT'', ''Breast 2'', ''Prostate'', ''Leukemia 2'', ''Brain'', ''NCI'' and ''9 Tumors''.

Comparing with MCTS based wrapper approach, MOTiFS, our method outperforms on *12* out of *14* datasets in term of both accuracy and number of selected features. The comparison of the number of selected features shows the dominance and significance of our proposed method. H-MOTiFS selects a very small number of features with highly improved accuracy as compared to MOTiFS.

While comparing H-MOTiFS with traditional wrapper based (SFS and IWSSr), multi-objective (ENORA) and evolutionary (HPSO-LS, PSO (4-2), ACO, PSO) approaches, our method outperforms on all datasets, except one dataset ''Leukemia 2'' where PSO shows the best result as compared to our method. However, it tends to select a large number of features.

Summarizing overall results on *14* microarray datasets, H-MOTiFS stands the best on *8*(''Leukemia 1'', ''Lymphoma'', ''SRBCT'', ''Breast 2'', ''Brain'', ''9 Tumors'', ''DLBCL'' and ''CNS'') datasets. On *3* datasets ''Prostate'', ''Adenocarcinoma'' and ''NCI'', our method shows the 2^{nd} best performance in a row. Only on *3* datasets, ''Colon'', ''Leukemia 2'' and ''Breast 3'', H-MOTiFS stands 3^{rd} or less. The last column in Table 8 shows the accuracy results without performing any feature selection. The highly improved performance of H-MOTiFS with very small number of selected features, as compared to no feature selection, shows its significance in feature selection for very high dimensional datasets.

3) NON-PARAMETRIC STATISTICAL TESTS

Comparing mean accuracy scores, our proposed method shows superior performance in most of the cases (as presented above). However, we perform the statistical tests to validate whether the results achieved by the proposed method are statistically significant. We perform the Wilcoxon Signed-Ranks and Friedman tests [62] with p value of 0.05 to verify whether H-MOTiFS outperforms the comparison methods in the experiments.

For pair-wise comparison between the H-MOTiFS and the other methods, we conduct the Wilcoxon Signed-Ranks test for small and high dimensional datasets and report the results in Tables 9 and 10, respectively. We perform the test where the number of datasets is at least 8. The high values of R^+ and low values of R^- indicate that H-MOTiFS outperforms all the other methods for both the small and high dimensional datasets. Further we test the significance of results by observing the p value. Observing Table 9 reveals that the p values against the MOTiFS, SFS, SBS, and E-FSGA are lower than the significant level of 0.05, thus, indicating the rejection of null hypotheses. However, the p value against the SFSW is greater than 0.05, indicating that H-MOTiFS is not significantly better than SFSW. Table 10 for high dimensional datasets shows the p values against MOTiFS, SFS, IWSSr, ENORA, and PSO are less than 0.05, thus, revealing the significance of H-MOTiFS. The p values against RF and SRKNN reveal that the null hypothesis is not rejected. These results suggest that H-MOTiFS is significantly dominated over most of the other methods. Specially, on high dimensional datasets the dominance of H-MOTiFS is amplified and a very small values of p are observed in most of the cases.

The overall impact of H-MOTiFS among the multiple methods is drawn using the Friedman test. We report the results of Friedman test in Table 11 and Table 12, for small dimensional and high dimensional datasets, respectively. In both the tables, H-MOTiFS stands 1^{st} with the lowest rank value among the other methods. Moreover, $p < 0.05$ in both the tables shows the significant difference in results. It also indicates the dominance and significance of H-MOTiFS over all the other methods.

E. DISCUSSION

We proposed a novel hybrid framework to investigate and understand the importance of MCTS in feature selection for very high dimensional datasets. Because of the less branching factor, our algorithm traverses the feature selection tree efficiently and finds the best feature subset by incorporating tree search with random sampling. The induction of the filter reduces the impact of randomness and aids in removing the noisy features during each simulation.

We test the effectiveness of our proposed method by experimenting on 30 datasets including very high dimensional (microarray) datasets. The promising results show that our proposed method is able to select the top k best features. The diversity in the results among different datasets indicates the significance of the *top-k* filter. The nominal standard deviation shows the stability of the search procedure at a fix value of k in a *top-k* filter. The remarkable performance, especially on very high dimensional datasets, makes our approach a perfect candidate to be considered as a standard feature selection approach and for future researches.

In our proposed method, we use the deterministic number of simulations, s. For n number of features, the complexity

TABLE 8. Comparison (w.r.t Avg. Accuracy & no. of Sel. Feat) of H-MOTiFS with other methods, for microarray datasets. Best results in each row are bold and underlined. The second best results in each row are in bold face. "-" are placed wherever information is not available.

Dataset	Avg. Acc. (# Sel. Feat.)											No. Feat. Sel.
	H-MOTiFS	MOTiFS [43]	SFS*	IWSSr* [54]	ENORA* [58]	PSO*	RF [55]	SRKNN [44]	HPSO-LS [59]	PSO (4-2) [59]	ACO [59]	
Colon	0.816 (20)	0.760 (973)	0.724 (4.5)	0.790 (7)	0.806 (36)	0.774 (442)	**0.841** (22)	**0.82** (83)	0.802 (30)	0.676 (-)	0.752 (-)	0.774
Leukemia 1	**0.930** (30)	0.877 (3509.5)	0.874 (2)	**0.903** (5)	0.847 (84)	**0.903** (2465)	-	-	0.899 (100)	0.850 (-)	0.795 (-)	0.875
Lymphoma	**0.987** (20)	**0.983** (1895.4)	0.919 (2)	0.919 (7)	0.952 (11)	0.968 (298)	0.953 (73)	0.980 (20)	0.877 (50)	0.664 (-)	0.706 (-)	0.984
SRBCT	**0.983** (30)	0.830 (1118.6)	0.843 (5.5)	0.940 (8)	0.880 (46)	0.880 (666)	0.961 (101)	**0.980** (35)	-	-	-	0.843
Breast 2	**0.665** (20)	0.614 (2414.4)	0.558 (7)	0.608 (16)	0.545 (29)	0.571 (1731)	**0.663** (14)	0.540 (15)	-	-	-	0.584
Prostate	**0.904** (10)	0.827 (2967.7)	0.882 (4)	0.902 (8)	0.824 (46)	0.843 (2530)	**0.939** (18)	0.900 (23)	-	-	-	0.853
Leukemia 2	0.963 (20)	**0.993** (1430.9)	0.954 (1)	0.842 (3)	0.895 (4)	**0.974** (189)	0.913 (2)	0.960 (25)	-	-	-	1.000
Brain	**0.792** (30)	0.785 (2712.3)	0.619 (5)	0.714 (7)	0.738 (35)	**0.786** (1294)	0.784 (22)	0.780 (23)	-	-	-	0.762
Adenocarcinoma	**0.822** (20)	0.814 (4875)	0.803 (4)	0.816 (8)	0.816 (40)	0.820 (3177)	0.815 (6)	**0.950** (13)	-	-	-	0.816
Breast 3	0.539 (30)	0.496 (2405)	0.558 (11)	0.495 (24)	0.526 (25)	0.495 (1462)	**0.654** (110)	**0.570** (44)	-	-	-	0.505
NCI	**0.675** (30)	**0.705** (2565.9)	0.541 (13)	0.607 (17)	0.607 (75)	0.656 (1655)	0.673 (230)	0.640 (33)	-	-	-	0.688
9 Tumors	**0.521** (30)	0.400 (2857.1)	0.333 (19)	0.400 (14)	0.417 (57)	**0.483** (1872)	-	0.430 (25)	-	-	-	0.367
DLBCL	**0.963** (10)	0.841 (3480.7)	0.805 (3.7)	**0.857** (7)	0.831 (13)	0.831 (1697)	-	-	-	-	-	0.831
CNS	**0.610** (20)	0.567 (3516.7)	0.530 (4.1)	0.533 (8)	0.550 (191)	**0.600** (2133)	-	-	-	-	-	0.567

*Evaluated using Weka library

TABLE 9. Results of Wilcoxon test for the small dimensional datasets.

H-MOTiFS vs.	R^+	R^-	p value
MOTiFS	106.5	28.5	0.043
SFSW	51	15	0.109
SFS	120	0	0.001
SBS	36	0	0.012
E-FSGA	33	3	0.036

TABLE 10. Results of Wilcoxon test for high dimensional (microarray) datasets.

H-MOTiFS vs.	R^+	R^-	p value
MOTiFS	96	9	0.006
SFS	103	2	0.002
IWSSr	105	0	0.001
ENORA	105	0	0.001
PSO	101	4	0.002
RF	31	24	0.721
SRKNN	44.5	21.5	0.306

of one simulation is given as, $O(nl + f + c)$, where l is the complexity of node selection, f and c are the complexities of the filter and the classifier, respectively. As node selection l is a constant, the complexity can be stated as, $O(n + f + c)$. Including the complexities of the filter (ReliefF), $O(nm)$, and the nearest neighbor classifier, $O(nm)$, the complexity of one simulation becomes $O(n + nm + nm)$, where m represents the number of instances. Hence, the overall complexity of the proposed algorithm $O(snm)$ is linear to the number of features.

For comparison, the complexity of MOTiFS is stated as $O(snm)$. The complexities of SRKNN and random forests are stated as $O(n^{\frac{3}{2}}bm)$ and $O(n^{\frac{1}{2}}bm\log m)$, where b represents the number of base classifiers [44]. The complexity

TABLE 11. Results of Friedman test for the small dimensional datasets.

Methods	Rank
H-MOTiFS	**1.64**
MOTiFS	2.09
SFSW	2.55
SFS	3.64

$p = 0.004$

TABLE 12. Results of Friedman test for high dimensional (microarray) datasets.

Methods	Rank
H-MOTiFS	**2.00**
MOTiFS	4.50
SFS	6.55
IWSSr	5.70
ENORA	5.95
PSO	4.5
RF	3.2
SRKNN	3.6

$p = 0.0003$

of SFSW is given as $O(n + m^2)$, where m represents the data points for inter-class and intra-class distance computations [23]. The complexity of FSFS is $O(n^2m)$. The complexity comparisons show that our proposed method is better or equally efficient as compared to various comparison methods.

Future research directions may include the detailed sensitivity analysis and/or experimenting with different reward functions for the improved performance.

V. CONCLUSIONS

In this study, we proposed a novel framework based on MCTS, to deal with very high dimensional datasets with an objective to achieve high accuracy with reduced dimensions. In our proposed framework, MCTS is deployed in conjunction with the hybrid of filter-wrapper methods. Our proposed method efficiently searched the feature space by exploiting the good solutions along with exploring the new ones. Our proposed method was able to find the top k best features in fewer simulations, relatively. The simplicity and the less model complexity are the key characteristics of our method, as only three hyper-parameters are associated. Experiments were performed on 30 publically available datasets including the very high dimensional (microarray) datasets. Comparison with the state-of-the-art methods showed the significance of our proposed method.

REFERENCES

[1] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1205–1224, 2004.

[2] E. Gasca, J. S. Sánchez, and R. Alonso, "Eliminating redundancy and irrelevance using a new MLP-based feature selection method," *Pattern Recognit.*, vol. 39, no. 2, pp. 313–315, 2006.

[3] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, Jan. 2010.

[4] S. Adams, A. P. Beling, and R. Cogill, "Feature selection for hidden Markov models and hidden semi-Markov models," *IEEE Access*, vol. 4, pp. 1642–1657, 2016.

[5] Z. Si, H. Yu, and Z. Ma, "Learning deep features for dna methylation data analysis," *IEEE Access*, vol. 4, pp. 2732–2737, 2016.

[6] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, 2003.

[7] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, and M. Buckland, "A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis," *IEEE Access*, vol. 4, pp. 9145–9154, 2016.

[8] R. M. Mehmood, R. Du, and H. J. Lee, "Optimal feature selection and deep learning ensembles method for emotion recognition from human brain EEG sensors," *IEEE Access*, vol. 5, pp. 14797–14806, 2017.

[9] M. Reif and F. Shafait, "Efficient feature size reduction via predictive forward selection," *Pattern Recognit.*, vol. 47, no. 4, pp. 1664–1673, 2014.

[10] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering—A filter solution," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2002, pp. 115–122.

[11] Y. Kim, W. N. Street, and F. Menczer, "Feature selection in unsupervised learning via evolutionary search," in *Proc. 1th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 365–369.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[13] T. M. Hamdani, J. M. Won, A. M. Alimi, and F. Karray, "Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2501–2509, 2011.

[14] J.-H. Hong and S.-B. Cho, "Efficient huge-scale feature selection with speciated genetic algorithm," *Pattern Recognit. Lett.*, vol. 27, no. 2, pp. 143–150, 2006.

[15] A. Unler, A. Murat, and R. B. Chinnam, "mr^2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification," *Inf. Sci.*, vol. 181, no. 20, pp. 4625–4641, 2011.

[16] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015.

[17] B. Xue, M. Zhang, and W. N. Browne, "Single feature ranking and binary particle swarm optimisation based feature subset ranking for feature selection," in *Proc. 35th Australas. Comput. Sci. Conf.*, vol. 122, 2012, pp. 27–36.

[18] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.

[19] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proc. Brazilian Symp. Comput. Graphic Image Process.*, Aug. 2012, pp. 291–297.

[20] D. Rodrigues *et al.*, "A wrapper approach for feature selection based on bat algorithm and optimum-path forest," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2250–2258, Apr. 2014.

[21] M. M. Kabir, M. Shahjahan, and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012.

[22] S. Tabakhi and P. Moradi, "Relevance–redundancy feature selection based on ant colony optimization," *Pattern Recognit.*, vol. 48, no. 9, pp. 2798–2811, 2015.

[23] S. Paul and S. Das, "Simultaneous feature selection and weighting—An evolutionary multi-objective optimization approach," *Pattern Recognit. Lett.*, vol. 65, pp. 51–59, Nov. 2015.

[24] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[25] M. Montazeri, "HHFS: Hyper-heuristic feature selection," *Intell. Data Anal.*, vol. 20, no. 4, pp. 953–974, 2016.

[26] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[27] C. Browne and E. Powley, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI in Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[28] M. Hall, "Correlation-based feature selection for machine learning," *Methodology*, 1999. [Online]. Available: https://www.cs.waikato.ac.nz/~mhall/thesis.pdf
[29] A. Senawi, H.-L. Wei, and S. A. Billings, "A new maximum relevance-minimum multicollinearity (MRmMC) method for feature selection and ranking," *Pattern Recognit.*, vol. 67, pp. 47–61, Jul. 2017.
[30] G. Zhao, Y. Wu, F. Chen, J. Zhang, and J. Bai, "Effective feature selection using feature vector graph for classification," *Neurocomputing*, vol. 151, pp. 376–389, Mar. 2015.
[31] W. Gao, L. Hu, and P. Zhang, "Class-specific mutual information variation for feature selection," *Pattern Recognit.*, vol. 79, pp. 328–339, Jul. 2018.
[32] W. Gao, L. Hu, P. Zhang, and F. Wang, "Feature selection by integrating two groups of feature evaluation criteria," *Expert Syst. Appl.*, vol. 110, pp. 11–19, Nov. 2018.
[33] W. Gao, L. Hu, P. Zhang, and J. He, "Feature selection considering the composition of feature relevancy," *Pattern Recognit. Lett.*, vol. 112, pp. 70–74, Sep. 2018.
[34] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006.
[35] S. S. Durbha, R. L. King, and N. H. Younan, "Wrapper-based feature subset selection for rapid image information mining," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 43–47, Jan. 2010.
[36] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.
[37] P. Bermejo, J. A. Gámez, and J. M. Puerta, "A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets," *Pattern Recognit. Lett.*, vol. 32, no. 5, pp. 701–711, 2011.
[38] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new hybrid filter–wrapper feature selection method for clustering based on ranking," *Neurocomputing*, vol. 214, pp. 866–880, Nov. 2016.
[39] F. Li, Z. Zhang, and C. Jin, "Feature selection with partition differentiation entropy for large-scale data sets," *Inf. Sci.*, vol. 329, pp. 690–700, Feb. 2016.
[40] R. Gaudel and M. Sebag, "Feature selection as a one-player game," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 359–366.
[41] S. M. H. Fard, A. Hamzeh, and S. Hashemi, "Using reinforcement learning to find an optimal set of features," *Comput. Math. Appl.*, vol. 66, no. 10, pp. 1892–1904, 2013.
[42] M.-H. Z. Ashtiani, M. N. Ahmadabadi, and B. N. Araabi, "Bandit-based local feature subset selection," *Neurocomputing*, vol. 138, pp. 371–382, Aug. 2014.
[43] M. U. Chaudhry and J.-H. Lee, "MOTiFS: Monte Carlo tree search based feature selection," *Entropy*, vol. 20, no. 5, p. 385, 2018.
[44] C. H. Park and S. B. Kim, "Sequential random k-nearest neighbor feature selection for high-dimensional data," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2336–2342, 2015.
[45] L. Devroye, L. Gyorfi, A. Krzyzak, and G. Lugosi, "On the strong universal consistency of nearest neighbor regression function estimates," *Ann. Statist.*, vol. 22, no. 3, pp. 1371–1385, 1994.
[46] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.
[47] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.
[48] J. Novaković, P. Strbac, and D. Bulatović, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugosl. J. Oper. Res.*, vol. 21, no. 1, pp. 119–135, 2011.
[49] M. Moradkhani, A. Amiri, M. Javaherian, and H. Safari, "A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm," *Appl. Soft Comput. J.*, vol. 35, pp. 123–135, Oct. 2015.
[50] Retrieved from University of California, Irvine. *Machine Learning Repository*. Accessed: Jul. 18, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html
[51] C. Chang and C. Lin. (2001). *Retrieved from LIBSVM—A Library for Support Vector Machines*. Accessed: Jul. 18, 2017. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
[52] *Gene Selection and Classification of Microarray Data Using Random Forest*. Accessed: May 1, 2018. [Online]. Available: https://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html
[53] *Gems-system*. Accessed: May 1, 2018. [Online]. Available: https://www.gems-system.org/
[54] P. Bermejo, J. A. Gamez, and J. M. Puerta, "Improving incremental wrapper-based subset selection via replacement and early stopping," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 5, pp. 605–625, 2011.
[55] R. Díaz-Uriarte and S. A. de Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinf.*, vol. 7, p. 3, Jan. 2006.
[56] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.
[57] Y.-C. Chen, N. R. Pal, and I.-F. Chung, "An integrated mechanism for feature selection and fuzzy rule extraction for classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 683–698, Aug. 2012.
[58] F. Jiménez, G. Sánchez, J. M. García, G. Sciavicco, and L. Miralles, "Multi-objective evolutionary feature selection for online sales forecasting," *Neurocomputing*, vol. 234, pp. 75–92, Apr. 2017.
[59] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft Comput.*, vol. 43, pp. 117–130, Jun. 2016.
[60] A. K. Das, S. Das, and A. Ghosh, "Ensemble feature selection using bi-objective genetic algorithm," *Knowl.-Based Syst.*, vol. 123, pp. 116–127, May 2017.
[61] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput. J.*, vol. 62, pp. 441–453, Jan. 2018.
[62] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

MUHAMMAD UMAR CHAUDHRY received the B.S. degree in computer engineering from Bahauddin Zakariya University, Multan, Pakistan, in 2009. From 2010 to 2014, he was an Instructor with the Virtual University of Pakistan, Lahore, Pakistan. He is currently pursuing the Ph.D. degree with Sungkyunkwan University, Suwon, South Korea. His research interests include recommender systems, feature selection, and machine learning.

JEE-HYONG LE received the B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1993, 1995, and 1999, respectively. From 2000 to 2002, he was an International Fellow at SRI International, USA. In 2002, he joined Sungkyunkwan University, Suwon, South Korea, as a Faculty Member. His research interests include fuzzy theory and applications, intelligent systems, and machine learning.