IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 12, 2019, accepted January 1, 2020, date of publication January 7, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964726

A Literature Survey and Empirical Study of Meta-Learning for Classifier Selection

IRFAN KHAN[1], XIANCHAO ZHANG[1], MOBASHAR REHMAN[2], AND RAHMAN ALI[3]
[1]School of Software, Dalian University of Technology, Dalian 116620, China
[2]Department of Information Systems, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar 31900, Malaysia
[3]QACC, University of Peshawar, Peshawar 25120, Pakistan

Corresponding author: Mobashar Rehman (mobashar@utar.edu.my)

This work was supported by Universiti Tunku Abdul Rahman.

ABSTRACT Classification is the key and most widely studied paradigm in machine learning community. The selection of appropriate classification algorithm for a particular problem is a challenging task, formally known as algorithm selection problem (ASP) in literature. It is increasingly becoming focus of research in machine learning community. Meta-learning has demonstrated substantial success in solving ASP, especially in the domain of classification. Considerable progress has been made in classification algorithm recommendation and researchers have proposed various methods in literature that tackles ASP in many different ways in meta-learning setup. Yet there is a lack of survey and comparative study that critically analyze, summarize and assess the performance of existing methods. To fill these gaps, in this paper we first present a literature survey of classification algorithm recommendation methods. The survey shed light on the motivational reasons for pursuing classifier selection through meta-learning and comprehensively discusses the different phases of classifier selection based on a generic framework that is formed as an outcome of reviewing prior works. Subsequently, we critically analyzed and summarized the existing studies from the literature in three important dimensions i.e., meta-features, meta-learner and meta-target. In the second part of this paper, we present extensive comparative evaluation of all the prominent methods for classifier selection based on 17 classification algorithms and 84 benchmark datasets. The comparative study quantitatively assesses the performance of classifier selection methods and highlight the limitations and strengths of meta-features, meta-learners and meta-target in classification algorithm recommendation system. Finally, we conclude this paper by identifying current challenges and suggesting future work directions. We expect that this work will provide baseline and a solid overview of state of the art works in this domain to new researchers, and will steer future research in this direction.

INDEX TERMS Meta-learning, algorithm selection, classification, machine learning.

I. INTRODUCTION

Classification is the most widely studied machine learning paradigm. The standard approach of classification algorithms is to learn from labeled training examples and then use that learning for classification of new unseen instances of dataset. Classification tasks are common in real world and researchers over the years had developed numerous algorithms that have widespread useful applications in many domains e.g., engineering, finance, biology, just to name a few. Each algorithm is intrinsically optimized and its performance on a particular task depends on how well its embedded fixed bias match the problem. Hence, there is no single algorithm that can learn all the tasks efficiently and every algorithm can perform better only on limited number of tasks. This phenomenon is also called performance complementarity [1], and is also been confirmed by the well known No Free Lunch theorem [2]. Since no single algorithm can best learn all the tasks effectively, the question that which algorithm should be used from the large number of available algorithms for a given task has gain tremendous importance and attention [1], [3]. The selection of appropriate algorithm from the available large set of algorithms is a non-trivial and challenging task, also known as model selection or Algorithm Selection Problem (ASP) in literature [4], [5].

The usual conventional approaches for algorithm selection e.g., trial and error, theoretical analysis or expert knowledge, has several drawbacks [6]. The trial and error approach

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

involves evaluation of all the available set of algorithms for a particular task, however it could not be practically possible in most cases, since the computational cost will be very high, especially for datasets that have fairly large size. Similarly, the emphasis of theoretical methods is to obtain knowledge about appropriateness of algorithms by evaluating their representational biases. Nevertheless, applicability of all classifiers cannot be theoretically analyzed [6]. Practically in most situations, domain experts that have experience in dealing with similar problems are employed for assistance in selection of suitable algorithm for a problem, however, this approach also have several shortcomings, for example, acquiring expert knowledge is generally costly or not readily available in most situations and can also be prone to personal bias and preference [5].

Taking into account, the drawbacks of conventional approaches for algorithm selection and the continuous successful applications of classification algorithms in numerous domains, there has been an ever-growing demand of machine learning systems that can automate the process of algorithm selection, i.e. provide intelligent assistance to end users by recommending potentially appropriate algorithms for various different tasks [7], [8]. Such systems will not only overcome the drawbacks of conventional algorithm selection approaches but will also broaden the application of machine learning algorithms to new problems and enable non-experts to apply machine learning more independently [9], [10]. One salient approach for automated algorithm selection is meta-learning based algorithm recommendation. It overcome the limitations of conventional algorithm selection approaches by automating the process of algorithm selection. Domain experts use their knowledge regarding performance of machine learning algorithms on previous tasks. Meta-learning imitate this strategy by accumulating the knowledge obtained from the application of algorithms on similar tasks and then use that knowledge to recommend a set of potentially appropriate algorithms for a specific task [11]. The typical area of meta-learning for algorithm recommendation is classification [12], [13], however it has also demonstrated success for algorithm selection in a number of other domains that includes clustering [14]–[16], instance selection [17], [18] regression [19] optimization [20]–[24] time series [25]–[27] and dynamic ensemble selection for performance improvement of classification problems [28]–[32].

As a matter of fact, classification is the most widely studied paradigm in machine learning, likewise classifier selection is also the mostly focused and researched area in the domain of automated algorithm selection [9], [33]. Meta-learning have demonstrated considerable success in classifier selection and vast amount of literature is available in this regard, in which various methods are proposed for classification algorithm recommendation. Yet there is no survey on meta-learning based classification algorithm selection methods that critically analyze and summarize the existing works, identify current challenges and suggest potential future research directions. Moreover, there is also lack of extensive comparative study in the literature that evaluates the performance of the prominent methods for algorithm selection and highlight their strengths and weaknesses as each newly proposed method in literature for classifier selection is compared only with one or two previous methods. To fill these gaps, this study aims to survey classifier selection methods and perform extensive comparative evaluation study. To the best of authors' knowledge it is the first survey and comparative study in this regard.

The reminder of the paper is structured as follows. Section II is related to background in which the first seminal framework for ASP is presented. Moreover, a short overview of related surveys and contributions of this paper are also highlighted. In section III, based on literature survey, we presented a generic work-flow of meta-learning system for algorithm recommendation. In section IV, detailed critical analysis of three important dimensions (meta-features, meta-learner, meta-target) of classifier selection is presented. Subsequently in section V, we present the summary and answers six Research Questions that were formulated on the survey part of the paper. In section VI, the experimental study is briefly described and comparative analysis of various methods are presented. Finally, we conclude the paper by identifying current challenges and suggesting future work directions in section VII.

II. BACKGROUND

A. RICE MODEL FOR ALGORITHM SELECTION PROBLEM

Previous research works in literature on algorithm selection problem (ASP) suggest that mapping measurable characteristics (meta-features) of problems (datasets) with performance of algorithms is a promising approach for building algorithm recommendation system [3]. The first seminal model based on this idea was presented by Rice [34]. Figure 1 shows the four basic components of Rice framework. The problem space P corresponds to problems in a certain domain e.g. classification, x in P represents a single instance in the problem space. The algorithm space A include algorithms that can be applied to all the instances x in P. The feature space F represent measurable characteristics of instances which are calculated through a feature extraction procedure applied on every $x \in P$. The performance space Y represents the mapping of each algorithm to a set of performance metric(s) e.g. accuracy and runtime. Given these notations, then according to [33], ASP can formally be defined as,

"For a given problem instance $x \in P$ with features $f(x) \in F$, find the selection mapping $S(f(x))$ into the algorithm space A, such that the selected algorithm $\alpha \in A$ maximizes the performance mapping $y(\alpha(x)) \in Y$ ".

Theoretically this framework is easy to understand, flexible in terms of adding new algorithms and problems and also have the ability of improving the learned mapping [35]. Nevertheless, it has certain implementation issues due to the complexity and size of P and A. The first component comprising of problem space P is high dimensional and has infinite cardinality since practically it is impossible to

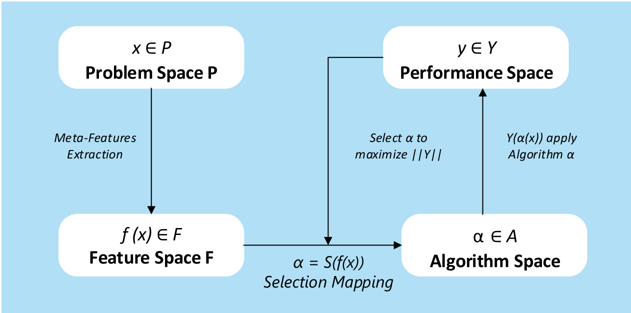


FIGURE 1. Rice representation of algorithm selection problem (From, Smith-Miles2008 [33]).

include all the problems in any algorithm recommendation system [20]. Similarly, the algorithm space A also has likely high cardinality, because it is also impossible to consider all the available candidate algorithms in algorithm recommendation system [20]. Another major problem with this framework was the lack of implementation guidelines on how to extract and map the characteristics of problems with the performance of algorithms. However the advent of meta-learning in the past two decades has favored as its implementation method with demonstrated success across many domains [16], [20], [36]. In section III, the Rice model is framed in the typical meta-learning work flow for algorithm selection.

B. META-LEARNING

The essence of meta-learning is "learning to learn" [37]. Meta-learning is defined as "the study of principled methods that make use of meta- knowledge to improve or construct self-improving learning systems through adaptation of machine learning and data mining processes" [38]. A meta-learning system is composed of a learning subsystem that adapts with accumulation of experience over time. The experience is acquired through knowledge gained from prior learning episodes. It is a broad field and has many important dimensions. Automated algorithm selection is one of the key research area in meta-learning [24]. In automated algorithm recommendation the prior experience is obtained through the learning of instances $x \in P$ by candidate algorithms $a \in A$. In meta-learning, the ASP is considered a usual learning problem in which dataset characteristics (meta-features) represent independent variables and target variable is the estimation of candidate algorithms performance. The issues of high cardinality of P and A space in the Rice model are tackled in the meta-learning studies by choosing problems of varying complexity, while selecting various complementary algorithms with distinct inherent biases for the algorithm space A [33]. One of the main leverage of meta-learning based algorithm recommendation is that once the system is built, it can learn and adapt with experience over time with the addition of more algorithms and problems. Meta-learning has also shown success in the dynamic ensemble selection of classifiers in Multiple Classifier Systems (MCS).Typically, MCS includes an ensemble of classifiers and a function for parallel combination of classifier outputs in order to increase accuracy on a classification task. Meta-learning in dynamic ensemble selection provide support by assessing the competence level of each base classifiers and only choose the competent ones from the available pool for a classification task [28]. Prior works in literature on dynamic ensemble selection in MCS demonstrate that it is an efficient method for problems in which the number of training examples are small and it is hard to train a single model on the dataset [29]. Beside this, meta-learning has also shown its viability in ensemble of noise filters for improving performance of noise filters [39]. Recently, a promising research direction in meta-learning is few shot learning, it refers to the learning of classification task by a classifier that generalizes well though trained on small number of training instances [40]. One salient approach to few-show classification is the meta-learning paradigm in which transferable knowledge is extracted and propagated from a collection of tasks for avoiding over fitting and improving generalization e.g., model initialization based methods [41], metric learning methods [42] and hallucination based methods [43]. Few shot learning has demonstrated success in classifying very complex tasks, e.g. in a recent work [44], the authors proposed a meta-learning based learning to learn approach for classification of videos.

C. RELATED SURVEYS AND OUR CONTRIBUTION

Meta-learning based algorithm selection has demonstrated success in many domains e.g., clustering [14]–[16], instance selection [17], [18] regression [19] optimization [20]–[24] time series [25]–[27]. Hence enormous amount of literature is available in this regard. Discussion of meta-learning surveys in other domains is outside the scope of this work, therefore in this paper, we only focus and discussed related surveys which are more directly linked to our work. To the best of authors knowledge, this is the first literature survey and extensive empirical study that is specifically focused on meta-learning based classifier selection. The summary of contributions of previous related surveys and our work is presented in Table 1.

The theoretical survey part of this paper aims to answer the following research questions (RQ's).

- RQ1 Is the problem space P properly explored in previous studies and are the datasets employed are representative and sufficient?
- RQ2: Are the pool of the candidate algorithms adequate and enough?
- RQ3: Are the feature space F properly explored and are the meta-features employed are diversified and complete?
- RQ4: What is the scope of the types of meta-targets employed in previous studies? What are their merits and demerits?
- RQ5: What is the coverage of approaches used for mapping the problem space P into the performance space Y at the meta-level and which approaches mostly suits classifier recommendation?

TABLE 1. Summary of contribution of previous surveys and the current paper.

Ref	Title	Focus	Method	Contributions
[45]	A perspective view and survey of meta-learning	Meta-Learning Review and Self Adaptive Learners	Literature Survey	It is the first detailed survey in meta-learning,mainly comprised of two parts. In the first part, the authors present their own perspective of meta-learning regarding self adaptive learners, i.e. learning algorithms that improve their bias dynamically through experience by accumulating meta-knowledge. In second part, the authors reviews various research directions of meta-learning in literature and argues that regardless of the various definitions and research directions, the primary objective of meta learning remains the same, i.e. how to use prior knowledge for performance improvement of learning algorithms that is obtained from the application of algorithms to similar problems .
[33]	Cross-disciplinary perspectives on meta-learning for algorithm selection	Integration of various approaches for algorithm selection in different domains	Literature Survey	It presents an insightful overview on cross disciplinary perspectives of meta-learning in algorithm selection. This paper highlighted the issue of limited generalization of meta-learning beyond classification and discussed several relevant attempts in other disciplines for solving algorithm selection problem. The author introduced a unified framework that coupled the cross-disciplinary developments in various domains regarding algorithm selection. Moreover, the author also draw attention towards expansion of meta learning concepts to new domains and discussed its application to algorithms that focus on problems like optimization, forecasting, sorting, constraint satisfaction and bioinformatics.
[38]	Metalearning: a survey of trends and technologies	Broad overview of various research direction within the scope of meta-learning	Literature Survey	The contribution of this survey is that it reconciled different definitions of meta-learning from the literature, present a broader overview of various research direction that are carried out within the scope of meta-learning. Moreover, the authors identified and discussed major research challenges in the domain of meta-learning.
Current paper	A literature Survey and Empirical Study of Meta-Learning for Classifier Selection	Meta-learning based classification algorithm selection	Literature survey + Evaluation	In the current paper, we present a literature survey of classification algorithm recommendation methods. It comprehensively discusses the different phases of classifier selection based on a generic framework that is formed as an outcome of reviewing prior works. Subsequently, we critically analyzed and summarized the existing studies from the literature in three important dimensions, i.e. meta-features, meta-learner and meta-target and answered six research questions that were formulated on the three important dimensions. In the second part of this paper, we perform extensive comparative evaluation study of all the prominent methods for classifier selection on 17 classification algorithms and 84 benchmark datasets. It present the comparative analysis that quantitatively assess the performance of various classifier selection methods and highlight their limitations and strengths in a classification algorithm recommendation system. Moreover, we discussed the current challenges and suggested future work directions. To the best of authors knowledge, it is the first detailed survey and comparative study that specifically focus on meta-learning based classification algorithm recommendation. We hope that this work will provide guidelines and prove to be an important source for researchers. Furthermore, We also expect that the shortcomings and challenges highlighted in this paper would further be addressed in upcoming works that will steer future research in this direction.

- RQ6: What type of evaluation measures are used in the meta-level and are they suitable?

We summarized, analyzed and discussed each of the RQ in section V.

Regarding the rationale behind extensive comparative study, the literature survey show that each of the newly proposed methods in prior works was only compared with one or two methods. Moreover, the use of same group of meta-features in the related works does not necessarily mean that the authors had exactly used the same measures for extracting the meta-features. Furthermore, the type and number of candidate algorithms and datasets used in previous studies are also diverse. Consequently, the lack of a detailed comparative study that evaluate the promising methods and the above mentioned impediments make it hard to quantitatively assess the performance of each method. Hence we performed a detailed extensive comparative evaluation study, presented in section VI.

III. META-LEARNING FRAMEWORK FOR CLASSIFIER SELECTION

Through a detailed literature review of meta-learning based algorithm recommendation studies, a generic framework for algorithm selection is presented in Figure 2. In meta-learning literature, the algorithm selection problem is considered a usual learning problem in which the dataset characteristics (meta-features) represents independent variables and target variable is the estimation of performance of candidate algorithms on datasets. The framework has three main basic components as shown in Figure 2 and described below in detail.

The first step in the meta learning framework is the construction of meta-knowledge. As shown in Figure 2, the construction of knowledge base consist of two sub modules (a) Performance evaluation (b) Data set Characterization. First of all, a subset of diverse range of classification problems referred to as problem space P are selected for building the meta-knowledge database. Similarly various complementary

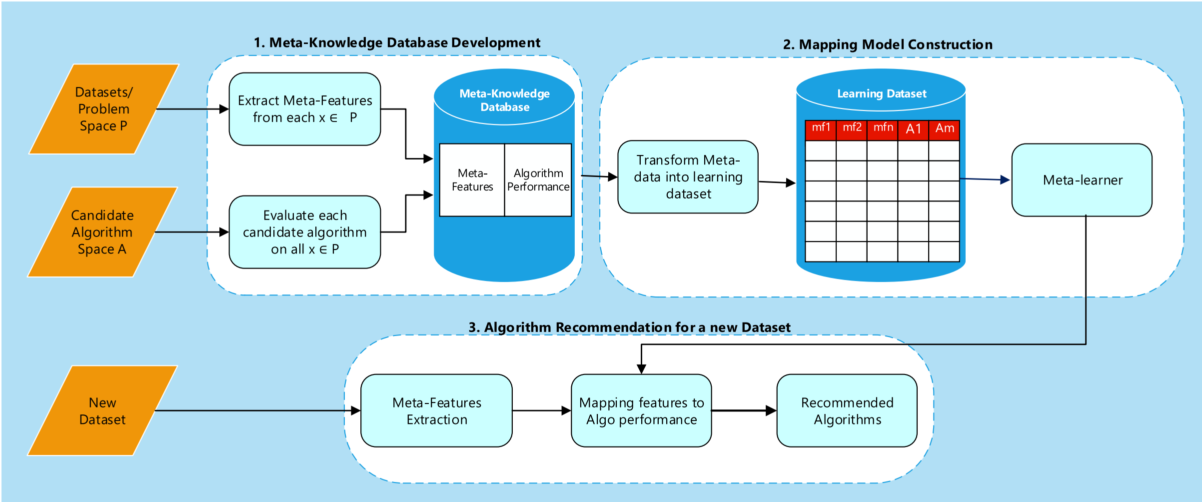


FIGURE 2. Generic framework for algorithm(s) recommendation.

algorithms referred to as algorithm space A (candidate algorithms) are selected such that they can potentially solve the problem instances x in P. The first submodule involves the extraction of a set of measurable characteristics of the problem instances, also known as meta features or data characterization. The meta-features are used to uniformly characterize datasets. There are various groups of meta-features which are described in detail in section IV-A. Details about the number of candidate algorithms and classification problems used in previous meta-learning studies for algorithm recommendation can be found in Table 3. In the second submodule all the candidate algorithms at algorithm space A are applied on the datasets at the problem space P and their performance evaluation measures (accuracy and runtime) are noted in the database. The final outcome of the first component is the Meta-Knowledge Database, which contain meta-features extracted from each dataset and the performance measures of all candidate algorithms.

After the construction of the meta-knowledge database that contains information about dataset characteristics and performance evaluation measures of candidate algorithms, next step is the construction of mapping model. The objective of mapping model construction is to model the relationship between dataset characteristics (meta-features) and performance information of candidate algorithms. In latest approaches [6], [12], [13], [46], the meta-knowledge is first transformed into a learning dataset so that a model can be train on them. Transformation of the meta-data into a learning dataset involves determining the set of best performing algorithms for each dataset in the knowledge-base. According to the recommendation of Demsar [47] and frequently followed procedure in previous similar studies in literature [6], [12], [13], [46], the multiple comparison procedure (MCP) is recommended for identifying the best performing algorithms for each dataset. The non parametric MCP, Friedman test followed by Holm procedure test are performed at significance level of 0.05 in order to identify best performing algorithms among the candidate algorithm for each dataset such that the difference in performance of identified algorithms are not statistically significant. The advantage of MCP test is that it allows to compare two or more set of metrics e.g., accuracy and runtime, while controlling type I error [6]. In algorithm recommender systems, minimizing the effect of type I error is very important because high type I error means that probability of falsely excluding algorithms which are not significantly different from the best one(s) will be high. Consequently, if the meta-knowledge is not appropriately transformed into learning dataset then the meta-learner cannot adequately model relationship between dataset-characteristics and performance of candidate algorithms, which will degrade performance of the system. After applying MCP, now each problem in the learning dataset is represented by meta-features and their corresponding best performing algorithms. Any learning algorithm like ML-KNN [48] can now be trained on the learning dataset to model the intrinsic relationship between meta-features and performance of algorithms. The training examples in the learning dataset are also called meta-examples.

Once a model is trained on the learning dataset then the recommendation of potentially suitable algorithms for any new problem instance $x_{new} \in$ P involves extraction of meta-features $f(x) \in F$ as input to the learned model that recommend set of appropriate algorithms a $\in$ A such that it maximize the performance mapping $y(\alpha(x)) \in Y$.

IV. DIMENSIONS OF META-LEARNING WORKFLOW

The main difference in literature regarding the methods proposed for classifier selection lies in: (i) Variation among the group of meta-features used (ii) Variation in the Procedures used for mapping feature space F to performance space Y (Meta-learner) (iii) Type of method used for estimating the relative performance of candidate algorithms on the datasets

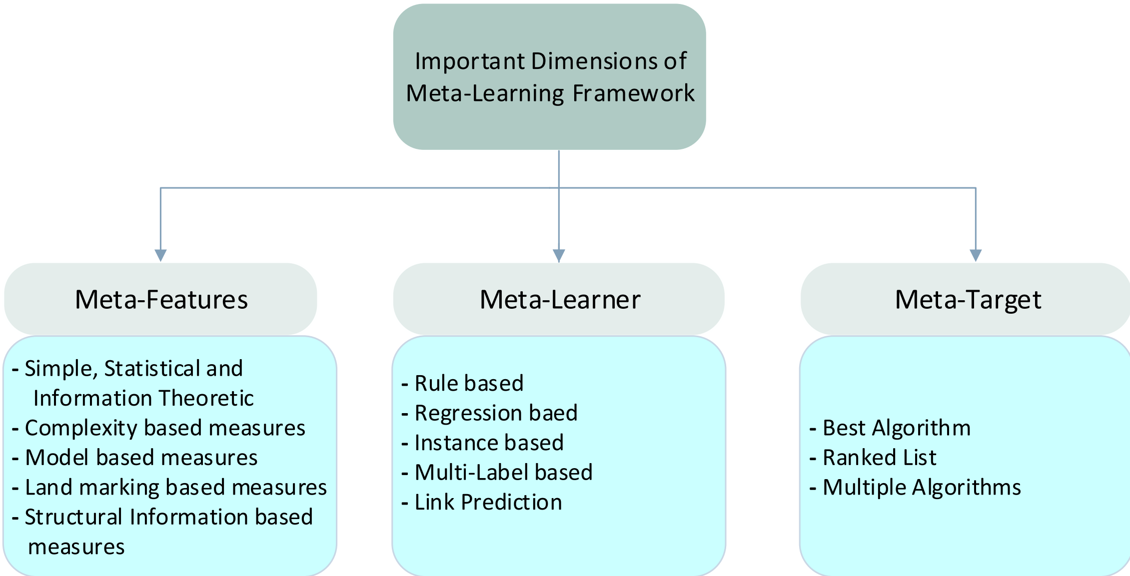


FIGURE 3. **Important dimensions of meta-learning Framework.**

(Meta-target). Hence we categorized and analyzed existing papers from literature survey in these three important dimensions as shown in Figure 3 (a) Meta-Features (b) Meta-Learner (c) Meta-Target.

A. META-FEATURES

Meta features are the collection of measurable characteristics of datasets that are able to estimate the performance of learning algorithms on problems. In the context of algorithm recommendation, meta-features must have the following properties and any new measure with these properties can potentially be added to the existing groups of meta-features. They are: (i) The measures should be efficiently and uniformly computable for wide range of problems in a particular domain (ii) The measures should low computational cost, easy to calculate and must take less time (iii) There must be intrinsic relationship between meta features and the performance of algorithms.

The measures and methods for extraction of meta-features varies across different domains. Thus specialized groups of meta-features and methods for its extraction are required for every different domains. The development of domain specific meta-features is a difficult task and therefore is considered an impediment in applying meta-learning for algorithm selection to new domains [35]. Nevertheless, various groups of meta-features are proposed for several domains e.g., clustering [14], instance selection [17], regression [19] and optimization [20]. In his paper, we only discussed meta-features that are used for classifier selection.They are grouped into five subcategories and every group represents a number of subset of measures that share similarities among them. The specific measures that are extracted by each of the five groups are shown in Table 2. Following are two examples of these measures that how are they calculated. More information on how to calculate each group of these measures can be found in [6], [49], [50].

- Maximum Fisher's discriminant ratio (F1)

$$F1 = \frac{1}{1 + \max_{i=1}^{m} r_{f_i}}$$

It measure overlap within the values of features in different classes. Here r_{f_i} is a discriminant ratio for every feature f_i.

- Entropy of Class Proportions (C1)

$$C1 = \frac{1}{\log\left(n_{c_i}\right)} \sum_{i=1}^{n_c} p_{c_i} \log\left(p_{c_i}\right)$$

This measure is used to capture the imbalance in a dataset, where p_{c_i} is the proportion of examples in each of the classes.

1) SIMPLE, STATISTICAL AND INFORMATION-THEORY-BASED MEASURES

This group represents the largest and diversified group of meta-features, also called general measures. It has low computational cost and comprised of three sub-categories as shown in table 2. The simple measures represent basic information like the number of instances or attributes. The statistical measures extract information regarding the distribution of numerical attributes of a dataset like dispersion and central tendency, while the information-theoretic measures are based on entropy which describes variability and redundancy of discrete attributes [50]. This group of meta-features is the most widely used in classifier algorithm selection [9], [51], [52]. It were first used in the Statlog project [53]–[55]. These meta-features are also used in the recommendation of feature selection algorithms [56] and noise filters [57] for classification tasks.

TABLE 2. Dataset characterization techniques(Meta-features).

Category	Dataset characterization measures (Meta-features)
	1. Simple, Statistical & Information Theoretic
Simple	Number of instances, Number of Attributes, Number of target concept values, Proportion of minority target, Proportion of majority target, Proportion of binary attributes, Proportion of nominal attributes, Proportion of numeric attributes
Statistical	Proportion of instances with missing values, Proportion of missing values, Geometric mean, Harmonic mean, Trim mean excluding the highest and lowest 5 %, Mean absolute deviation, Variance, Standard deviation, Interquartile range, Proportion of numerical attributes with outliers,
Info theo	Skewness of data based on numerical attributes, Kurtosis of data based on numerical attributes, Maximum eigenvalue
	2. Complexity Based Measures
Features based	Maximum Fisher discriminant ratio, Directional vector maximum Fisher discriminant ratio,
Linearity	Volume of overlapping region, Maximum individual feature efficiency, Collective feature efficiency
	Sum of the error distance by linear programming, Error rate of linear classifier, Non linearity of linear classifier
Neighborhood	Fraction of borderline points, Ratio of intra/extra class NN distance, Error rate of NN classifier, Non linearity of NN classifier
	Fraction of hyperspheres covering data, Local set average cardinality
Dimensionality	Average number of points per dimension, Average number of points per PCA dimension,
	Ratio of the PCA dimension to the original dimension
Class Balance	Entropy of classes proportions, Imbalance ratio
	3. Model Based Measures
Model Based	height of tree, width of tree, Number of nodes in tree, number of leaves in tree, maximum number of nodes at one level, mean of the number of nodes on levels, length of the longest branch, length of the shortest branch, Mean of the branch lengths, Standard deviation of the branch lengths, Minimum occurrence of attributes, Maximum occurrence of attributes, Mean of the number of occurrences of attributes, Standard deviation of the number of occurrences of attributes
	4. Landmarking Based Measures
Landmarking	Naive Bayes, ii) 1-NN (Nearest Neighbor), iii) Elite 1-NN, iv) a decision node learner, v) a random chosen node learner and vi) the worst node learner. The last three learners can be achieved based on the well-known learning algorithm C4.5.
	5. Structural Information Based Measures
Structural info based	First, the two feature vectors, one-item feature vector and two-item feature vector are generated from a dataset. The two vectors are comprised of frequencies of one-item sets and two-item sets. Then the minimum, 1/8 quantile; 2/8 quantile; 3/8 quantile; 4/8 quantile; 5/8 quantile; 6/8 quantile; 7/8 quantile and maximum are calculated for the two vectors, which are employed as meta-features.

2) PROBLEM-COMPLEXITY-BASED MEASURES

These measures examine the spatial distribution of data and evaluate the source of difficulty in a dataset by explaining its geometrical complexity. Moreover, these measures compute the approximate size and shape of decision boundary that separate the classes [49], [58]. It includes the measures presented by Ho and Basu in [59]. It define complexity of the boundary that separate binary classification problems. Afterwards, these measures were extended in other studies to multi-class classification problems [58]. It includes the following measures

(i) Feature based: It characterize to what extent are the available features providing information in differentiating the classes.

(ii) Linearity measures, It estimates that whether the classes are linearly separable or not. (iii) Neighborhood measures, It describe the existence and density of different or same classes in local neighborhoods (iv) Dimensionality measures, It evaluates the data sparsity in accordance with the number of samples relative to the data dimensionality. (v) Class balance measures, It takes into account the ratio of the number of examples among classes. These measures are also used in the formulation of novel data driven preprocessing [49]. Moreover, The authors in [57], [60], has shown the these measures can successfully estimate the performance of noise filters on datasets. In addition, recently the results of a study have provided evidence of significance of these measures in predicting the performance of classifiers that are usually used in micro-array data analysis [61].

3) MODEL-STRUCTURE-BASED MEASURES

Contrarily to the previous data characterization methods that were calculated from the data distribution, the model based is indirect characterization method and are calculated by inducing a decision tree model on a dataset to get information about the hidden structures of the data [62]. The properties of the tree are used as meta-features. Its advantage is that it does not only rely on the distribution of the data but consider the representation of the data set in a special structure for getting information about the learning complexity. However its drawback is that it has relatively high computational cost associated with it.

4) LANDMARKING-BASED MEASURES

In this group of meta-features, the performance information from the application of few simple and fast learners on datasets are used for recommendation of appropriate algorithms for a particular task [63]. The basic idea behind landmarking is that each problem has specific characteristics that associate it to an area of expertise of certain algorithms. In this context, the candidate algorithms that have nearest inductive bias to landmarkers that achieve high performance on the given task are preferred. For example, for two landmarkers A and B, if landmarker A outperform landmarker B on a data set then the candidate algorithm that has the nearest inductive bias to landmarker A is considered suitable for the dataset. The learners which are used as landmarkers must have different inherent bias and should have low computational cost [64].

5) STRUCTURAL-INFORMATION-BASED MEASURES

Based on Tatti work [65] of measuring similarity between datasets through summary statistics, the author in [66], extended the set of meta-features by introducing structural and information based method to characterize binary datasets. These measures were improved and extended to ordinary classification problems in later studies [6]. This group of meta-features has shown its significance in algorithm recommendation systems [12], [67]. It adopt frequencies of item-sets with respect to the parity function to characterize a dataset [66]. First a given dataset is converted to its relevant binary dataset. After that one-item-set V_I and two-item V_{II} set are generated, the V_I acquire the distribution of values in a given attribute and V_{II} represent the correlation among two features. The V_I and V_{II} are sorted in ascending order to compute the statistical summary of the items to attain a unified representation of dataset. The statistical summary includes the minimum, maximum and seven octile's of the item-sets.

B. META-LEARNER

Meta-learner refers to the algorithm that model the relationship between dataset characteristics (meta-features) and candidate algorithms. For any given problem, the meta-learner receives meta-features as input and recommends appropriate algorithms according to the learned model.

In the first preliminary work on meta-learning for classifier selection in StatLog project [53], the decision tree algorithm C4.5 was used as meta-learner for modeling the relationship between meta-features and 22 classification algorithms. Likewise the authors in [68], employed the advanced rule based learning algorithm (C5.0) to generate recommendation rules for 8 candidate algorithms. The output of both of these works was set of recommendation rues which were to be checked manually for selection of appropriate algorithm. Although the recommendation rules were not precise enough but they still were able to narrow down the set of candidate algorithms and the user had to evaluate and choose algorithm from limited set of algorithms for a particular problem.

The work of the StatLog project was further extended in another European METAL (Meta-learning assistant for providing user support in machine learning and data mining) project [69], in which a fully automated web based tool called Data Mining Advisor (DMA) was designed. The DMA employed K-Nearest Neighbors (KNN) algorithm as meta-learner and the meta-knowledge was built on 67 datasets and 10 classification algorithms. In order to recommend potentially suitable algorithms for a dataset, the proposed system first identifies its K most similar data sets in the meta-knowledge database through the distance measure calculated upon meta-features. Then the performance information of the identified similar datasets are aggregated on the candidate algorithms in order to produce a ranked list of algorithms. Usually, the top 3 ranked algorithms are considered appropriate for a dataset. The advantage of using instance based meta learner like k-Nearest Neighbor is that the training data available in the start of a meta learning system is usually small, which makes it difficult to induce models that are general and frequently produce crisp threshold like decision trees and rule induction algorithms [70]. The instance based methods has also the added advantage of providing flexibility to easily extend the meta learning system. Initially in meta-learning systems for algorithms recommendation the meta-knowledge database is built upon small set of meta examples and new meta examples are added to the meta-knowledge database later upon their availability [71]. With the use of KNN as meta-learner, the addition of meta examples from new experimental results can quickly be integrated into the existing meta examples in the meta-knowledge database without the requirement of remodeling the relationship between meta-features and performance of candidate algorithms. This approach is used in majority of the studies for classifier selection, examples includes [39], [46], [51], [66], [67], [71]–[73]. However, the drawback of this approach is the selection of optimal value for the parameter K because the number of similar datasets vary for each problem in the meta-knowledge database [12].

Furthermore, the authors in [18], [39], [74], [75], employed regression algorithms to learn the relationship between meta-features and performance of candidate algorithms. In order to recommend algorithms for any given problem, the proposed method first extract meta-features from the dataset and which are then given as input to the learned regression model that estimate the performance of candidate algorithms on the problem. This approach estimates the predictive performance of each candidate algorithm independently rather than a ranked list of recommended algorithms. Hence the user directly gets information about the expected accuracy of candidate classifiers for a particular task. However, its drawback is that the meta-learner (regression algorithm) must be trained separately on each of the candidate algorithm. Moreover, it also requires retraining of the meta-learner every time when new meta-examples are added to the meta-knowledge database. Hence, the computational cost of this method is comparatively high.

In addition the authors in [6], presented a multi-label based methodology for algorithm selection. In the proposed method, the authors framed the algorithm selection problem in a meta-learning setup as a multi-label learning problem. In this regard the meta-examples at the meta-knowledge database are transformed into a multi-label classification data set. The meta-examples are first transformed into multi-label dataset by identifying appropriate candidate algorithms for each dataset through MCP. In each record of the multi-label dataset, the meta-features represents independent variables while the corresponding multiple appropriate algorithms are the dependent variables. After transforming the meta-examples into a multi-label learning dataset then the multi-label lazy learning algorithm ML-KNN [48] is trained on the meta-data to model the relationship between meta-features and performance of algorithms. The proposed method showed efficacy on experiments performed on 84 datasets and 13 candidate classification algorithms on five groups of meta-features. However, this method also suffers from the drawback of selection of the optimal value for parameter K.

Moreover, the authors in [6], [76], [77] investigated unsupervised learning for modeling the relationship between meta-features and performance of algorithms. In [6], the authors employed clustering technique on meta-features in order to cluster similar datasets in the meta-knowledge database. Subsequently all the candidate algorithms are evaluated on every cluster and appropriate algorithms for each of the cluster are identified through MCP. The recommendation of algorithm for a new dataset involves the extraction of meta-features and identifying its nearest similar cluster through the distance between its meta-features and the center of each cluster. Once the nearest similar cluster of a dataset is identified then the already identified appropriate algorithms for that cluster are recommended for the given problem.

More recently the authors in [12], proposed a new method in which the algorithm recommendation is framed as a link prediction problem in a meta-learning setup. In this method the meta-knowledge base is imitated in a heterogeneous network and then a link prediction algorithm is employed as meta-learner for algorithm recommendation. In the proposed method first a heterogeneous network comprised of 131 datasets and 21 candidate classification algorithms is constructed. Then each dataset is linked with the best performing candidate algorithms, which are identified through the statistical test procedure (Friedman test followed by Holm procedure test). For each dataset in the network its nearest 5 similar datasets are identified through the KNN approach and a link is made for each dataset with its 5 nearest similar datasets. Then the recommendation of algorithms for a new problem instance involves the extraction of its meta-features and linking it to its nearest similar dataset in the network. Afterwards the similarity based link prediction methods LRW (Local Random Walk) and SRW (Superposed Random Walk) are employed on the network and the algorithms with highest link probability are recommended as appropriate algorithms.

C. META-TARGET

As described earlier that meta-learning view algorithm selection as a regular learning problem in which the independent variables are represented by meta-features and the learning target is the meta-target. In meta-learning setup for algorithm recommendation, the meta-target corresponds to the type of output that the system produces in the form of estimated relative performance of candidate algorithms for any given problem. It can be represented in many distinct ways, currently three types of meta-target for algorithm selection based on meta-learning are reported in the literature i.e. (i) Best Algorithm (ii) Ranked List (iii) Multiple Algorithms.

The methods in which the meta-target is the prediction of best algorithm for a dataset, then only one algorithm is recommended that are predicted to perform best among the candidate algorithms by the meta-learning system for the dataset. However, it is important to note that there is a certain level of dependency to choose from several combinations of meta-learner and meta-targets in developing a meta-learning system for algorithm recommendation. Specifically, the choice of meta-target to be employed for algorithm recommendation also affects the choice of meta-learner. For example, Usually singe label algorithms are employed as meta-learners when the meta-target is the prediction of single best algorithm. The authors in [53], [68], employed rule based learners while the authors in [18], [39], [74], [75], used regression algorithms. The drawback in algorithm recommendation systems in which the meta-target is the recommendation of single best algorithm is that the end user has no choice to choose from a set of few potentially appropriate algorithms for a classification task.

The second type of meta-target (Ranked List) in the literature corresponds to the recommendation of a ranked list of candidate algorithms for a problem. Normally the user chooses from the top three ranked algorithms of his choice. In algorithm recommendation systems in which ranked list are employed as meta-target, the instance based learner KNN is used as meta learner. Ranking based meta-target is employed in most of the studies e.g. [39], [46], [51],

TABLE 3. Summary of meta-learning based algorithm recommendation studies in classification domain.

	RQ1	RQ2	RQ3	RQ4					RQ5	RQ6
Ref	Datasets	Criteria	Algo	Meta Features					Meta learner	Meta target
				SI	PC	MS	LM	STI		
[72]	77	Accuracy	3	✓					KNN	Ranked List
[73]	47	Accuracy	10	✓		✓	✓		KNN	Ranked List
[71]	53	Accuracy+Run Time	10	✓			✓		KNN	Ranked List
[68]	100	Accuracy+Run Time	8	✓					C5.0(Rule Based)	Multiple Algorithms
[13]	84	Accuracy+Run Time	13	✓	✓	✓	✓	✓	ML-KNN	Multiple Algorithms
[66]	84	Accuracy+Run Time	17	✓	✓	✓	✓	✓	KNN	Ranked List
[46]	115	Accuracy+Run Time	22	✓	✓	✓	✓	✓	KNN	Ranked List
[6]	84	Accuracy+Run Time	17	✓	✓	✓	✓	✓	Cluster	Multiple Algorithms
[12]	131	Accuracy+Run Time	21	✓	✓	✓	✓	✓	Link Pred	Ranked List
[51]	39	Accuracy+Run Time	18	✓			✓		KNN	Ranked List
[67]	80	Accuracy+Run Time	11	✓	✓	✓	✓	✓	KNN	Ranked List
[18]	40	Accuracy+Run Time	6	✓	✓				Regression	Best Algorithm
[39]	90	Accuracy+Run Time	5	✓	✓	✓	✓		KNN	Ranked List
[57]	53	Accuracy	6	✓	✓	✓	✓		Regression	Best Algorithm
[77]	85	Accuracy	15	✓			✓		Cluster	Multiple Algorithms
[74]	54	Accuracy	9	✓		✓	✓		Regression	Best Algorithm
[75]	65	Accuracy	8	✓			✓		Regression	Best Algorithm
[69]	67	Accuracy+Run Time	10	✓		✓	✓		KNN	Ranked List
[53]	22	Accuracy	22	✓					C4.5(Rule Based)	Multiple Algorithms
[76]	57	Accuracy	6	✓					Cluster	Multiple Algorithms

[66], [67], [71]–[73]. Its advantage is that it provides more options to the end user's in order to choose algorithm of their choice from the top 3 ranked algorithms that can potentially perform better on a given task. Whereas, its drawback is that the end user's don't have any information regarding the potential significant statistical difference in performance of the top 3 ranked algorithms. Beside that, there could be other candidate algorithms that are computationally cheaper and are competitive in terms of performance to the top 3 ranked algorithms but still the end user's will not consider them for a given problem.

Contrary to the previous approaches of meta-targets, when the meta-target is multiple algorithms then the system recommends set of algorithms that potentially have no significant difference in performance on a given problem. The advantage of this approach is that it recommends multiple appropriate instead of predicting the single best or a ranked list of algorithms. The user can select any algorithm of his choice from the recommended set of algorithms. The approach is used in [6], [13], [77].

V. SUMMARY AND DISCUSSION

The following sub sections presents summary of the literature survey and discussion.

A. SUMMARY

Research papers from the detailed literature survey are summarized in Table 3. Data from relevant related papers are extracted and presented in terms of the Research Questions (RQs) introduced in section II-C. For better understanding and readability of Table 3, we have sub categorized few of the RQs and assigned them notations as follows:

- (RQ4) The meta-features are categorized into groups that represents similar measures, they are: (i) Statistical and Information-Theory-Based Measures (SI) (ii) Problem-Complexity-Based Measures (PC) (iii) Model-Structure-Based Measures (MS) (iv) Landmarking-Based Measures (LM) (v) Structural-Information-Based Measures (STI)
- (RQ5) The meta-targets are organized into (i) Best algorithm (ii) Ranked list (iii) Multiple algorithms
- (RQ6) The measures for the base-evaluation are organized into single criteria that measures only accuracy and multi-criteria that measures accuracy and run time.

1) DATASETS AND META-LEVEL EVALUATION MEASURES

In literature, majority of the studies used less than 100 datasets with few exceptions that used more than

100 datasets [12], [46]. There is no clear indication in literature regarding the adequate number of datasets at the meta-level, however, reasonable number of datasets must be considered that can appropriately map the feature space into the performance space Y [78], [79]. Moreover, Luengo [58], argue that the inclusion of datasets at the meta-level requires caution which can induce bias in meta-learning systems. The author recommends to incorporate datasets that are standard, diverse and representative in the particular domain. In fact, recent studies have used fairly reasonable number of diverse representative datasets from the standard UCI Machine Learning Repository. This ascertains moderate exploration of the problem space P and conforms to the trend in classification research area in which these datasets are regarded as benchmarks. However, few studies from the past had used relatively low number of datasets.

Regarding the RQ related to the evaluation measures, two types of evaluation metrics are involved in the meta-learning. One is related to the performance evaluation of the candidate algorithms on the benchmark datasets at the meta-level and the second is evaluating the performance of algorithm recommender system. The literature show that all of the recent studies used the multi-criteria performance evaluation metric Adjusted Ratio of Ratios (ARR, Def 1) at the knowledge base. Whereas, earlier studies considered only accuracy for the performance evaluation of candidate algorithms at the meta-knowledge database. Regarding the metrics concerning performance measure of the algorithm recommender system two types of metrics are considered standard and suitable (i) Recommendation Accuracy (Def 2) (ii) Hit Ratio (Def 3). These metrics are used in most of the studies [6], [12], [13], [66]–[68].

2) ALGORITHMS

Candidate algorithms in the literature that are frequently employed at the meta-level are distributed into the following categories: (i) Instance-based algorithms (ii) Probability-based Bayes Network and Naive Bayes, (iii) Tree-based C4.5, (iv) Rule-based Ripper and PART (v) Support vector machines. These represents the most commonly used algorithms for classification tasks and hence are used in majority of the algorithm recommender systems. Beside these algorithms some studies [6], [12], [13], have also employed ensemble of classifiers i.e. RandomForest, RandomTree, Bagging and Boosting with the simple classifiers like Instance based, Naive Bayes, C4.5 and PART. Latest approaches like neural networks could not be found in the meta-learning recommender systems. It is a major limitation in the existing studies because new approaches tend to perform better.

3) META-FEATURES

Initially, most of the studies used limited group of meta-features with the SI and LM are the most widely used in literature. Although the trend has changed and the recent studies has used all five groups of meta-features. However, it is not necessarily a leverage because it involves additional cost of calculating the meta features. It is therefore important to make this cost as minimal as possible thereby making sure that the automatic algorithm recommendation is not surpassed by computational cost incurred for calculating the meta features. Moreover, it also gives rise to the problem of high dimensionality in the feature space F which may result in a large though redundant set of meta-features. The findings of [80], suggest that the use of same group of meta-features in various studies does not necessarily mean that they had used exactly the same measures. Alternatively, the author proposed a framework and developed a standard R library [50] for the uniform and standardized generation of meta-features.

B. META-LEARNER AND META-TARGET

Concerning RQ 6, the literature survey show that researchers have used three types of meta-target. In initial studies the meta-target was the prediction of single best algorithm for a particular problem. While the most commonly used meta-target is the recommendation of a ranked list of algorithms, which is used in 10 out of 20 studies, however, the trend in recent studies has changed and the researchers are more interested in using multiple algorithms as meta-target.

Regarding and RQ 5, the literature shows that any algorithm like Rule based, Regression, Instance based or Multi label can be used meta as meta-learner, however more practical conditions are taken into consideration for the selection of suitable meta-learner. The choice of meta-target also limits the choice of meta-learner. Table 3 shows that for ranking based meta-target only KNN based algorithm is applied as meta-learner. It is also the most widely practiced approach in literature. The reasons are two folds. (i) One of the primary concern in the selection of meta-learner is the ease in extensibility of the system because a meta-learning system accumulate knowledge and evolves with experience as more meta-examples are added to the knowledge base. Hence, the addition of new meta-examples to the meta-knowledge database without the requirement of remapping the relationship of datasets and performance measures of candidate algorithms makes KNN a good choice for meta-learner. (ii) Contrarily to predicting the single best algorithm it recommends a ranked list of algorithms which provide more choices to the end user to choose from the top three algorithms. For Best Algorithm as meta-target, rule based and regression algorithms are used as meta-learner in literature. For Multiple Algorithms as meta-target, ML-KNN [13], clustering [6], [76], [77] are used as meta-learners in previous studies.

While there is no consensus in the literature regarding, which algorithm to use as meta-learner, every approach has its own merits and demerits [71]. The disadvantage of using ML-kNN and regression algorithms as meta-learner is that it is difficult to extend the system when new meta examples from different experiments became available. With the addition of new examples the system has to relearn the relationship between meta-features and performance of algorithms in meta-knowledge database.

VI. EMPIRICAL EVALUATION

As noted in the literature survey that each of the newly proposed methods in prior works was only compared with one or two previous methods. Moreover, the use of same group of meta-features in the related works does not necessarily mean that the authors had exactly used the same measures for the extraction of meta-features. In addition, the type and number of candidate algorithms and datasets also varies in previous studies. Hence, the above mentioned impediments and the lack of a detailed comparative study make it hard to quantitatively assess the performance and draw generic conclusion from literature regarding each method. The current comparative study aims to address these shortcomings by: (a) Comparing and assessing the performance of all the prominent methods for classifier selection. (b) Generating the measurable properties (meta-features) of datasets in a standard and unified manner such that the drawback of variation among the measurable properties in previous methods could be covered. Hence, ensuring a more fair comparison. (c) Keeping the problem and candidate algorithm space same for all the methods in order to ensure impartiality. (d) Assessing performance of the algorithm recommendation methods on each group of meta-features, and to evaluate the competence of each group in capturing the inherent properties of datasets that make specific learning algorithms to perform better on particular tasks.

As described earlier, algorithm recommendation has three important dimensions i.e., meta-features, meta-learner, meta-target. All these dimensions are briefly described in section section IV and summarized in Table 3. The main difference among prior studies in literature is due to the variation of these three important dimensions. In this comparative study, we have compared all the prominent methods for classifier selection that vary in these three dimensions. For better presentation of results and ease of readability, in this section we denote each method from literature with the following notations.

- Method A
  It denotes the multi-labeled based ML-KNN method for algorithm recommendation, in which the meta-learner is ML-KNN and the meta-target is a set of multiple appropriate recommended algorithms.
- Method B
  It denoted the instance based method for algorithm recommendation, in which KNN is used as meta-learner and the meta-target is a ranked list of algorithms.
- Method C
  It denote the method in which the algorithm recommendation is framed in a heterogeneous network and then link prediction technique is employed as meta-learner to recommend appropriate set of algorithms.
- Method D
  It denotes the method in which similar datasets are clustered and appropriate algorithms for each cluster are identified.
- Method E
  It denote the method in which regression algorithm is used as meta-learner and the meta-target is the recommendation of single best algorithm for a dataset.

In order for our comparative study to be reproducible and independently verified, We have provided details of every step.

A. KNOWLEDGE BASE CONSTRUCTION

According to the graphical representation of the methodology for algorithm recommendation based on meta-learning presented in Figure 2, the first major component is the construction of meta-knowledge database. Following are the details of every step involved in the construction of meta-knowledge base.

1) BENCHMARK DATASETS

For the construction of meta-knowledge database, 82 representative classification datasets of varying complexity from the standard UCI repository were used. Details of datasets regarding the number of instances, features and classes are given in Table 4.

2) CANDIDATE ALGORITHMS

Regarding the candidate algorithms, 17 well known representative classification algorithms having different induction biases are employed as candidate classifiers that are been used in prior studies in literature. They are: two rule-based (Ripper and PART), two probability based (Bayes Network and Naive Bayes), tree-based C4.5, Support Vector Machine and the Instance-based Learner (K-Nearest Neighbors). Similarly, we employed classifier ensembles, which include RandomTree and Tree Based RandomForest, Boosting and Bagging with four simple classifiers i.e., IBL, Naive Bayes, PART and C4.5.

The experiments for performance estimation of algorithm recommendation methods are performed in R version 3.5.1 [81]. The Java-based open source data-mining software WEKA version 3.8.2 [82] is used for the performance evaluation of candidate classifiers at the meta-knowledge database and the interface from WEKA to R was provided through RWeka Package [83]. Like the prior studies, the candidate classifiers were used with their default parameters in WEKA version 3.8.2, Support Vector Machine having Polynomial Kernal and K-Nearest Neighbors having Linear Search.

3) META-FEATURES

The five types of meta-features that are reported in the literature for classification algorithm recommendation were

TABLE 4. Details of datasets.

Dataset	#Instances	#Features	#Classes	Dataset	# Instances	#Features	#Classes
Abalone	4177	8	19	Led7digit	500	8	10
Acute-Inflammations	120	6	2	Lymphography	148	18	4
Acute-urinary	120	7	2	Mammographic-mass	961	6	2
anneal	898	26	6	Meta-data	528	22	24
audiology	226	69	24	Mines-vs-rocks	208	60	2
Australian statlog	690	14	2	Molecular-promotor	106	58	2
autos	205	25	7	monks-problems-1_test	432	7	2
Balance scale	625	4	3	monks-problems-1_train	124	6	2
Banknote-authentication	1372	5	2	monks-problems-2_test	432	7	2
Blood-transfusion-service	748	5	2	Movement-libras	360	91	15
Breast Cancer	286	9	2	mushroom	8124	22	2
Breast-cancer-wisconsin	699	10	2	Page-blocks	5473	10	5
bridges_version1	108	13	6	Parkinsons	195	23	2
bridges_version2	108	13	6	Planning-relax	182	13	2
Car Evaluation	1728	6	4	postoperative-patient-data	90	8	3
Cardiotocography	2126	23	10	qualitative-bankruptcy	250	7	2
Climate-simulation	540	18	2	seeds	210	7	3
CMC	1473	9	3	Image Segmentation	2310	19	7
colic	368	27	2	shuttle.landing.control	15	6	2
column2C	310	6	2	solar-flare_1	323	12	6
column3C	310	6	3	Connectionist Bench	208	60	2
Statlog (Australianl)	690	14	2	soybean	307	35	19
Statlog (German)	1000	20	2	spectf_test	267	44	2
cylinder-bands	512	39	2	Molecular Biology	3190	61	3
Dermatology	366	33	6	sponge	76	45	3
diabetes	768	20	2	Statlog-australian-credit	690	14	2
ecoli	336	8	8	Statlog-german-credit	1000	20	2
Fertility	100	10	2	Statlog-heart	270	13	2
Flags	194	30	5	Teaching Assistant Evaluation	151	5	3
Glass	214	10	7	Thoracic-surgery	470	17	2
Habermans-survival	306	3	2	Tic-tac-toe	958	9	2
Hayes-roth	160	5	3	trains	10	32	2
Heart Disease	303	13	5	User Knowledge Modeling	403	5	5
Statlog (Heart)	270	13	2	Statlog (Vehicle Silhouettes)	946	19	4
heart-statlog	270	13	2	Congressional Voting Records	435	16	2
Hepatitis	155	19	2	vowel	13	990	11
Horse-colic-surgical	368	27	2	Waveform Database Generator	5000	40	3
Indian-liver-patient	583	10	2	Wholesale-customers	440	8	2
Ionosphere	351	34	2	Wine	178	13	2
Iris	150	4	3	Yeast	1484	8	10
Chess	3196	36	2	Zoo	101	17	7

used for extracting the measurable characteristics of datasets at the meta-knowledge database. They are (i) Statistical and Information-Theory-Based measures (ii) Problem-Complexity-Based measures (iii) Model-Structure-Based measures (iv) Landmarking-Based measures (v) Structural-Information-Based measures. We used the standardized meta-feature generation framework [49], [50] for the extraction of meta-features.

B. METRICS FOR PERFORMANCE EVALUATION OF CANDIDATE ALGORITHMS AT META-LEVEL

Regarding the performance evaluation of candidate algorithms at meta-knowledge database, we used the multi-criteria evaluation measure ARR (Adjusted Ratio of Ratios), which combines information about the classification accuracy and total execution time of the learning algorithms. We acquired ARR from the literature which is consistently been used for performance evaluation at meta-level in similar studies, for instance in [6], [12], [13], [71]. Likewise, following similar studies, we performed 5×10-folds cross-validation for estimating the performance, in order to get stable performance. i.e., for each candidate classifier applied on every problem, the 10 fold cross validation is repeated 5 times by randomizing the order of instances. It controls the variation imputed by different choices of training and test instances [84]. Moreover, the performance metrics obtained from the cross validation is further used in the MCP for identifying appropriate algorithms for each dataset.

According to literature [12], [71], ARR is defined as

Definition 1 (ARR):

$$ARR^{d_k}_{A_i,A_j} = \frac{acc^k_i/acc^k_j}{1+\alpha.log(t^k_i/t^k_j)} \quad (1 \leq i \neq j \leq x, 1 \leq k \leq n) \tag{1}$$

$$ARR^{d_k}_{A_i} = \frac{1}{x-1}\sum\nolimits_{j=1 \wedge j \neq i}^{x} ARR^{D_k}_{A_i,A_j} \quad (1 \leq i \neq j \leq x, 1 \leq k \leq n) \tag{2}$$

The x and n in the above equation represent the number of candidate algorithms and datasets respectively. The acc^k_i denotes the classification accuracy and t_i represent the runtime of algorithm A_i on a dataset d_k respectively. The ARR value for each candidate algorithm on a dataset is calculated using equation 2. Here α represent the user-defined trade-off coefficient for relative importance of accuracy and runtime of candidate algorithms. It corresponds to the amount of accuracy that the end-user is willing to compromise for choosing algorithms that have low run time [6]. When the value of α is set to 0, it means that only accuracy is considered for the evaluation of algorithm, also called single criteria. In this work we have set the α value to 0 and 0.05% in order to evaluate the methods in both the single and multi-criteria performance evaluation metrics.

C. RECOMMENDATION MODEL CONSTRUCTION

After the construction of meta-knowledge database for meta-learning based algorithm system, the next step is recommendation model construction as shown in Figure 2. For method A [13], method C [12] and method D [6], the meta-knowledge database is first transformed into learning dataset according tho the procedure (MCP) presented in section III and then their corresponding meta-learners are trained on them as described in section (Meta-learner) IV-B. Likewise, meta-learners for method B and E are trained on the meta-data as described in section (Meta-learner) IV-B of the literature survey.

Following the literature, we have used the corresponding meta-learners that are employed for each method in the literature. For method A the multi-labeled (ML-KNN) [48] is used as meta-learner. Regarding method B, the well known Instance base learner (KNN) [85] is used as meta-learner that is been used in previous algorithm recommendation studies [66], [71] as a benchmark. For method C, the similarity based link prediction [86] algorithm is used as meta-learner. Regarding method D, the well known Expectation Maximization algorithm [87] and for method E, we used linear regression [85].

D. METRICS FOR EVALUATION ALGORITHM RECOMMENDATION METHODS

Regarding the performance evaluation of algorithm recommendation methods, we adopted metrics from the literature that are used in majority of the prior works [6], [12], [13], [66]–[68]. First we present some basic notations in order to better understand the metrics.

Let $A = \{a_1, a_2, a_3, a_x\}$ denotes the set of all candidate algorithms and $D = \{d_1, d_2, d_3 \ldots .d_n\}$ represents the N number of benchmark datasets. The meta-examples obtained from the N number of datasets are denoted by $M = \{m_1, m_2, m_3 \ldots m_n\}$. Every m_i in M represent a dataset d_i and is a row $< X_i, Y_i >$ $(1 \leq i \leq n)$. Here X_i is the set of meta-features generated from d_i and $Y_i \subseteq A$ represents the set of actual appropriate algorithms on d_i. Then for any problem d_i, suppose Z_i be the set of algorithms recommended by a meta-learning system. Then based on these notations the metrics can be defined as

Definition 2 (Recommendation Accuracy):

$$RecAcc(m_i) = \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}(1 \leq i \leq n) \tag{3}$$

$$RecAcc(M) = \frac{\sum_{i=1}^{n} RecAcc(m_i)}{N} \tag{4}$$

The recommendation accuracy for each method is calculated through equation 4.

Definition 3 (Hit Rate):

$$HitRate(m_i) = \begin{cases} 1, & if\ Z_i \cap Y_i \neq \emptyset \\ 0, & Else \end{cases} \tag{5}$$

$$HitRate(M) = \frac{\sum_{j=1}^{n} HitRate(m_i)}{N} \tag{6}$$

The Hit Rate matrix corresponds to the probability of appropriateness of recommended set of algorithms Z_i. When the value of Hit Rate matrix for a dataset $\{d_i\}$ is equal to 1, it indicates that the set of recommended algorithms Z_i contains at least one algorithm from the set of appropriate algorithms Y_i. The performance on the Hit Rate matrix for each method is calculated on equation 6. The higher values of these metrics indicate better performance of the algorithm recommendation systems. It could be noted that the definition of Recommendation Accuracy in equation 3 is quite tough and it penalize the performance on each inappropriate algorithm that is recommended by the system. This definition of recommendation accuracy metric is very tough condition for method E. Moreover, this method in the literature is also not evaluated on the same metric. Hence, for fair comparison we have not considered method E for comparison on Recommendation Accuracy metric.

E. RESULTS AND DISCUSSION

1) HIT RATE

The performance of algorithm recommendation methods on each group of meta-features in terms of Hit Rate are shown in Figure 4 ($\alpha = 0$, Accuracy) and 5 ($\alpha = 0.05$, Accuracy + Runtime). One other reference value in the results is also presented i.e. mean performance of every method on each group of meta-features, represented by the dotted line in the figures. It can be observed from Figure 4 and 5 that Method C (link prediction) and D (Clustering based) showed good performance. The mean Hit Rate (92%) of these methods

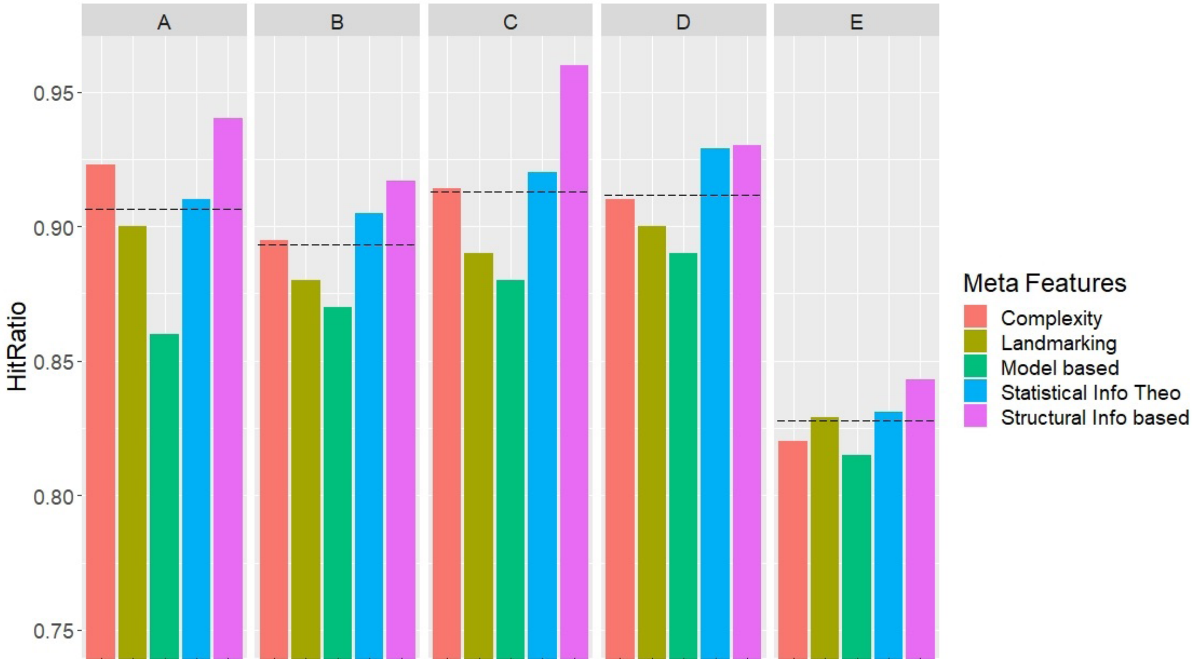


FIGURE 4. Comparison of algorithm recommendation methods on five groups of meta-features in terms of Hit Rate matrix on single criteria (Accuracy, $\alpha = 0$ in ARR).

are almost equal and high from the rest of the methods. It means that in the set of recommended algorithm by these methods for any classification problem, there is 92% probability that the set of recommended algorithms will contain at least one really appropriate algorithm which can achieve best performance on the given problem. Regarding method A (ML-KNN), which consider algorithm recommendation as a multi-label learning problem, the Hit Rate is also above 90%, closely followed by method B (KNN) but slightly lower than method C and D, which shows that these methods are also competent for classification algorithm recommendation. Hit Rate of method B (KNN) is almost 89%, while that E is the lowest among all. As described earlier, that it is easy to extend and add new meta-examples in method B, which is one of the meta-learning based algorithm recommendation systems i.e., to adapt and learn with experience, that is why this approach is been used in most of the prior studies. Moreover, It can be noted from Figure 5, that the performance of all the methods slightly decreased on the multi-criteria ARR ($\alpha = 0.05$). The reason is due to the fact that increasing the value of α in the multi-criteria ARR metric, according to the Definition 2, the set of appropriate algorithms decreases because the end users only wants those algorithms that have low runt time and high performance. As a whole the results of Hit Rate matrix on strategy A,B,C, and D demonstrate that they are adequate for classification algorithm recommendation. Regarding meta-features it can be noted from Figure 4 and 5 that usually the Hit Rate of all the methods is better on structural information theoretic measures. Overall the five groups of meta-features showed that they have the ability in identifying the hidden pattern in datasets and relating it to the inherent biases of different candidate classifiers for the recommendation of appropriate algorithms.

2) RECOMMENDATION ACCURACY

The performance of algorithm recommendation methods in terms of recommendation accuracy metric is shown in Figure 6 ($\alpha = 0$, Accuracy) and 7 ($\alpha = 0.05$, Accuracy + Run Time). The findings show that method C performs better with recommendation accuracy of 52%. The recommendation accuracy of method A is slightly lowers than C, closely followed by B. The performance of method D is poor on Recommendation Accuracy matrix as compared to its performance on Hit Rate matrix. As described in section IV-B, the reason is that in this method the datasets at the meta-level are clustered into various clusters based on similarity and then relevant suitable algorithms with nearest cluster are recommended for a dataset. The number of similar datasets in each cluster identified through the clustering approach are usually high. Hence the number of appropriate algorithms for each cluster is also high which are recommended as appropriate algorithm for a classification task. The higher number of algorithms recommended for a task increases its probability of containing at least one algorithm that perform best on the given task, which boost its performance on Hit Rate matrix. Nevertheless it also increases the chances of imputing unsuitable algorithms, which degrade its performance on Recommendation Accuracy matrix. The same pattern of Hit Rate and Accuracy regarding method D is also observed in another study [13]. As a matter of fact, the definition of recommendation accuracy (Definition 2) is very tough in the context of algorithm recommendation as it penalize the performance

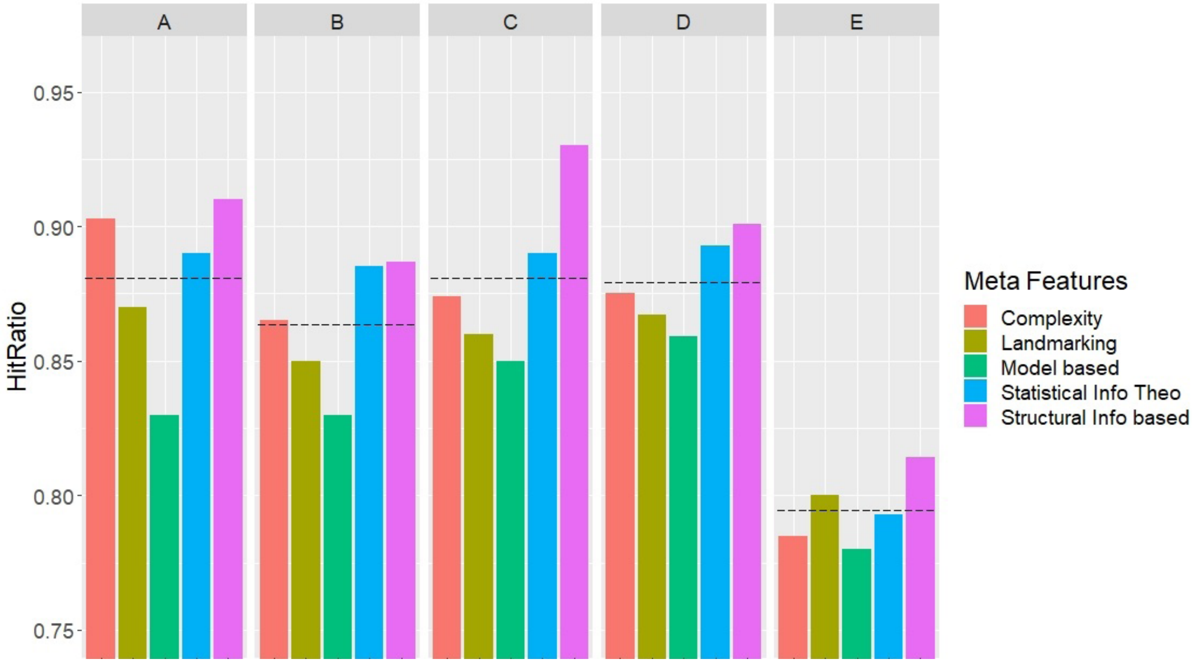


FIGURE 5. Comparison of algorithm recommendation methods on five groups of meta-features in terms of Hit Rate matrix on multi-criteria (Accuracy+Runtime, $\alpha = 0.05$ in ARR).

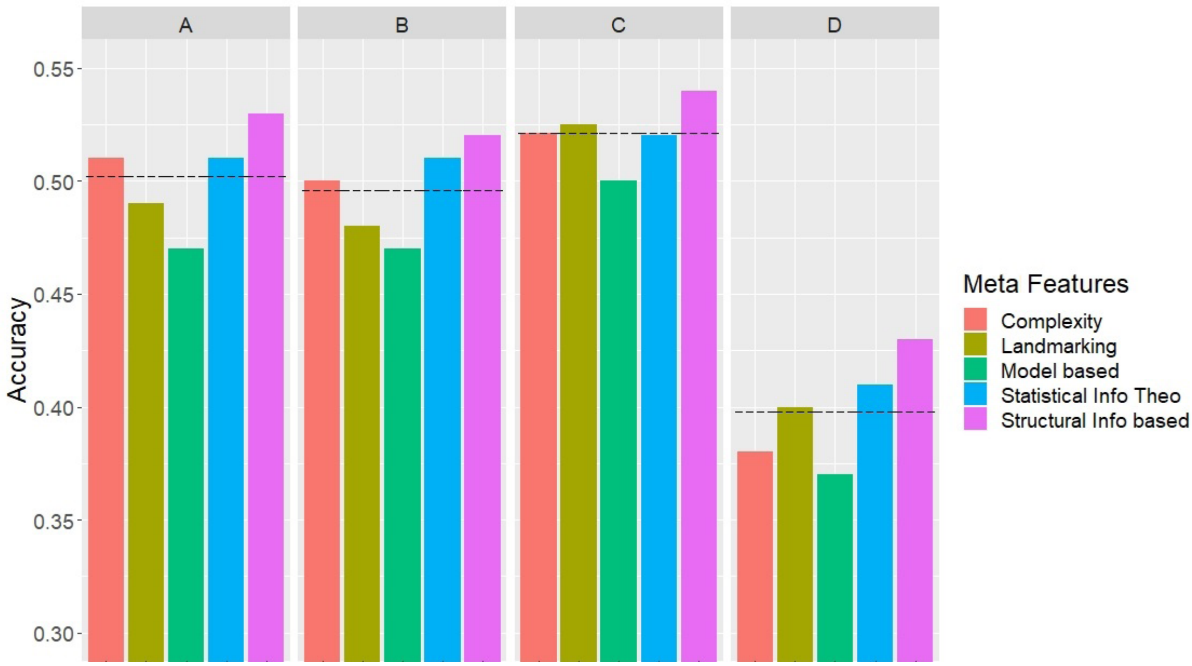


FIGURE 6. Comparison of algorithm recommendation methods on five groups of meta-features in terms of recommendation accuracy matrix on single-criteria (Accuracy+Runtime, $\alpha = 0$ in ARR).

by recommending inappropriate algorithms. Despite the strict definition of algorithm recommendation accuracy, the performance of method A,B, and C have demonstrated its adequacy on Recommendation Accuracy matrix.

As demonstrated in the results and discussed earlier, the generalization and adapting with experience ability of meta-learning systems for algorithms recommendation had compelled researchers to build stand-alone tools for algorithm recommendation and also to integrate these prominent methods into the current open source tools for machine learning, e.g. WEKA [9]. Moreover, the current focus is also to integrate these methods into famous machine learning

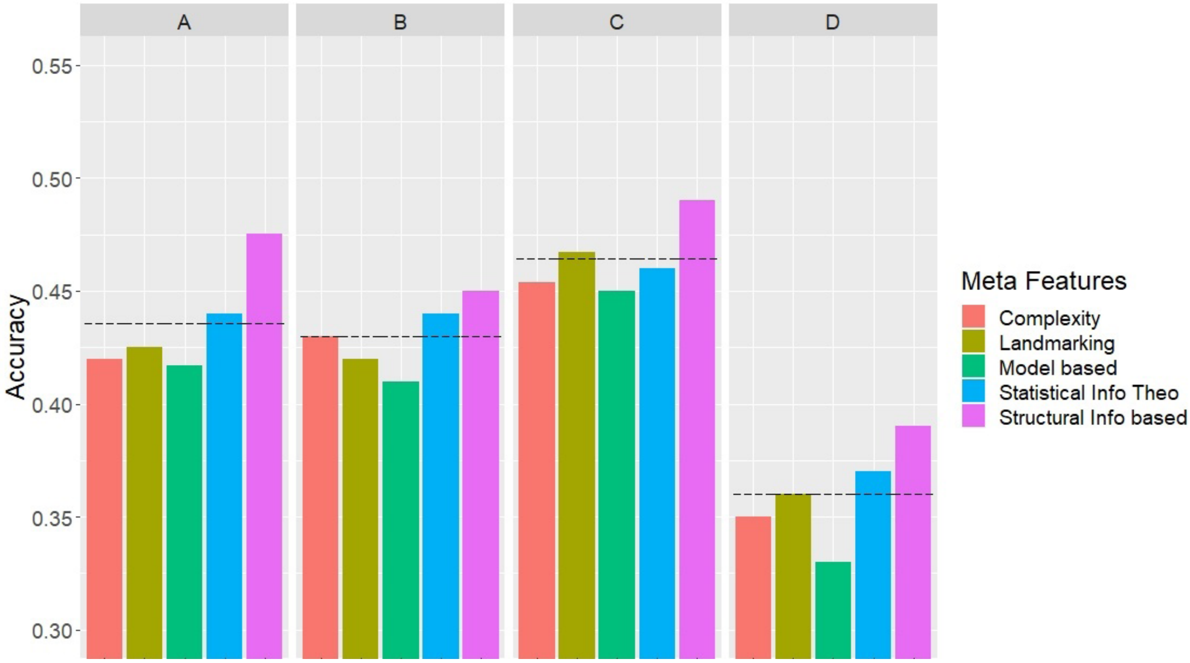


FIGURE 7. Comparison of algorithm recommendation methods on five groups of meta-features in terms of recommendation accuracy matrix on multi-criteria (Accuracy+Runtime, $\alpha = 0.05$ in ARR).

libraries in Python and R in order to provide intelligent assistant to end users regarding algorithm selection e.g. AUTO-SKLEARN [24].

VII. CONCLUSION AND FUTURE PROSPECTS

In this section, we present conclusion of our work and highlighted the current challenges along with suggesting future research directions.

A. CONCLUSION

In this work, we have deeply analyzed the problem of meta-learning based classification algorithm recommendation. The survey part of this paper present a thorough overview of important dimensions of meta-learning for classifier selection and answer six research questions that were formulated on three important dimensions i.e., meta-features, meta-learner and meta-target. Related work from literature is summarized and critically analyzed in this regard. The performance of algorithm recommendation methods is largely dependent on the quality of meta-features. It determines the intrinsic properties of dataset which are used by meta-learner to associate it with the inherent biases of various candidate algorithms. Though much progress has already been made in this direction and five groups of meta-features are proposed in literature, but still development of more efficient and low computational cost meta-features is a challenge and active research area.

After considering the results form extensive empirical study regarding the performance of various meta-learning methods on unified groups of meta-features, measured by recommendation accuracy and hit rate matrix, we reached the following conclusions. The Link Prediction, Multi-labeled (ML-KNN) and Instance based (KNN) based method offer the best performance for classifier selection. However, technical issues that affect the performance and increase the computational cost must be taken into consideration when developing a meta-learning system for algorithm recommendation. For instance, systems that are easy to extend and manage in future for incorporating more meta-examples and candidate classifiers are generally preferred because such systems better utilize the core concept of learning and adapting with experience overtime.

Overall, our deep analysis show that meta-learning have demonstrated success in automatic algorithm recommendation and apart from the experimental research studies, practical tools have also been developed that assist users in automatic algorithm selection e.g., the meta-learning based AUTO-SKLEARN, which is build on the famous Python Scikit-learn library won first phase of ChaLearn(Challenges in Machine Learning) AutoML challenge in 2015 [88]. However, there are still many challenges that needs to be addressed in the realm of algorithm selection based on meta-learning. In the following subsection, we present some of the takeaways from our work in terms of challenges and future research direction.

B. CHALLENGES AND FUTURE DIRECTIONS

Following are some of the takeaways from the survey in terms of challenges and future research direction.

- One of the limitations of developing algorithm recommender system is the computational cost associated with the evaluation of all the candidate algorithms in the algorithm space A on

all the datasets in the problem space P. Furthermore, increasing the number of candidate algorithms or datasets increase the computational cost of developing the meta-knowledge database. This hinders the rigorous exploration of the algorithm and problem space. Although the relevant studies in literature have used reasonable number of candidate algorithms and datasets, still the initial computational cost is high for building the system.
This drawback can be cover in future works by incorporating data in the meta-knowledge database from platforms like OpenML (Open Machine Learning) [89]. It is a collaborative science platform having an on-line data repository in which scientists share their experimental results from the application of algorithms on datasets. It also support advance searching and querying that enables researchers to reuse that information in meta-learning studies. It would not only reduce the computational cost but will also provide the opportunity to rigorously explore the problem and algorithm space and will ensure that no bias is induced into the system at meta-level.

- One important issue in meta-learning based algorithm recommendation is the additional computational cost associated with the extraction of meta-features. keeping this additional cost minimal is still a challenge. Moreover, there is still a need for developing more optimal techniques for the extraction of meta-features. The current measures like statistical, information-theoretic and structural information based present a global overview of distribution of the problems, as they are extracted by averaging the results of the measures on the entire dataset, typically smoothing the real distribution. Hence, alternatively more dataset characterization techniques should be developed that take into account the problem distribution in such a way that is more related to the learning performance.
- The literature also lacks detailed analysis of the meta-features. There is a need for a detailed analysis of the combination of individual measures across various groups of meta-features in order to remove redundancy. It will not only lower the computational cost of extracting the meta-features but can also improve the accuracy of the recommender system.
- The latest approach of imitating the meta-knowledge database in a heterogeneous network and then employing link prediction technique for algorithm recommendation shows better performance [12]. However, it still suffers from the drawback of the optimal value for the parameter K for identifying similar datasets through the KNN approach. Moreover, the value of K remain fixed for all the datasets. However, the number of similar datasets could not be same for all task at the meta-level, which may degrade performance of the system. In future the use of automatic clustering techniques for identifying the similar datasets should be investigated to check if it can boost the performance of algorithm recommender system.
- Although the existing studies had used reasonable number of standard classification algorithms yet there is a need to include more state of the art algorithms in future studies e.g., XGBOOST, Gradient Boosting Machines and LightGBM.
- For algorithm recommendation, meta-learning utilize knowledge obtained from the experience of application of algorithms to similar tasks. It would be intriguing to investigate, if the information among the candidate classifiers can appropriately be leveraged in a meta-learning setup for performance improvement, as prior evidence suggest that generalization performance is improved by leveraging valuable information contained in several related tasks [90].

Contrary to the prior surveys, by doing this detailed analysis which specifically focus on classification algorithm recommendation based on meta-learning, we expect that the shortcomings and challenges highlighted in this paper would further be addressed in upcoming works. Furthermore, we hope that this work will also provide a guideline and prove to be an important source for researchers, who wish to apply meta-learning to algorithm selection.

ACKNOWLEDGMENT

This work was supported by Universiti Tunku Abdul Rahman.

REFERENCES

[1] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection: Survey and perspectives," *Evol. Comput.*, vol. 27, no. 1, pp. 3–45, Mar. 2019.
[2] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[3] M. Lindauer, J. N. Van Rijn, and L. Kotthoff, "The algorithm selection competitions 2015 and 2017," *Artif. Intell.*, vol. 272, pp. 86–100, Jul. 2019.
[4] B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchette, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, and J. Vanschoren, "ASlib: A benchmark library for algorithm selection," *Artif. Intell.*, vol. 237, pp. 41–58, Aug. 2016.
[5] R. Ali, S. Lee, and T. C. Chung, "Accurate multi-criteria decision making methodology for recommending machine learning algorithm," *Expert Syst. Appl.*, vol. 71, pp. 257–278, Apr. 2017.
[6] G. Wang, Q. Song, and X. Zhu, "An improved data characterization method and its application in classification algorithm recommendation," *Appl. Intell.*, vol. 43, no. 4, pp. 892–912, Dec. 2015.
[7] M. Lindauer, F. Hutter, H. H. Hoos, and T. Schaub, "AutoFolio: An automatically configured algorithm selector," in *Proc. Int. Jt. Conf. Artif. Intell. (IJCAI)*, 2017, pp. 25–29.
[8] M. Lindauer, H. H. Hoos, F. Hutter, and T. Schaub, "AutoFolio: An automatically configured algorithm selector," *J. Artif. Intell. Res.*, vol. 53, pp. 745–778, Jul. 2018.
[9] F. Serban, J. Vanschoren, J.-U. Kietz, and A. Bernstein, "A survey of intelligent assistants for data analysis," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–35, Jun. 2013.

[10] H. Degroote, "Online algorithm selection," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 5173–5174
[11] M. Mustafa and M. Sebag, "ALORS: An algorithm recommender system for Classifiers," *Artif. Intell.*, vol. 244, pp. 291–314, Jun. 2017.
[12] X. Zhu, X. Yang, C. Ying, and G. Wang, "A new classification algorithm recommendation method based on link prediction," *Knowl.-Based Syst.*, vol. 159, pp. 171–185, Nov. 2018.
[13] G. Wang, Q. Song, X. Zhang, and K. Zhang, "A generic multilabel learning-based classification algorithm recommendation method," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 1, Oct. 2014, Art. no. 7.
[14] B. A. Pimentel and A. C. De Carvalho, "A new data characterization for selecting clustering algorithms using meta-learning," *Inf. Sci.*, vol. 477, pp. 203–219, Mar. 2019.
[15] J.-S. Lee and S. Olafsson, "A meta-learning approach for determining the number of clusters with consideration of nearest neighbors," *Inf. Sci.*, vol. 232, pp. 208–224, May 2013.
[16] D. G. Ferrari and L. N. De Castro, "Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods," *Inf. Sci.*, vol. 301, pp. 181–194, Apr. 2015.
[17] E. Leyva, A. Gonzalez, and R. Perez, "A set of complexity measures designed for applying meta-learning to instance selection," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 354–367, Feb. 2015.
[18] E. Leyva, Y. Caises, A. González, and R. Pérez, "On the use of meta-learning for instance selection: An architecture and an experimental study," *Inf. Sci.*, vol. 266, pp. 16–30, May 2014.
[19] A. C. Lorena, A. I. Maciel, P. B. C. De Miranda, I. G. Costa, and R. B. C. Prudêncio, "Data complexity meta-features for regression problems," *Mach. Learn.*, vol. 107, no. 1, pp. 209–246, Jan. 2018.
[20] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Inf. Sci.*, vol. 317, pp. 224–245, Oct. 2015.
[21] J. Kanda, A. de Carvalho, E. Hruschka, C. Soares, and P. Brazdil, "MetMeta-learning to select the best meta-heuristic for the Traveling Salesman Problem: A comparison of meta-features," *Neurocomputing*, vol. 205, pp. 393–406, Sep. 2016.
[22] T. Horváth, R. G. Mantovani, and A. C. P. L. F. De Carvalho, "Effects of random sampling on SVM hyper-parameter tuning," in *Proc. Adv. Intell. Syst. Comput. Intell. Syst. Design Appl.*, 2017, pp. 268–278.
[23] J. Vanschoren, B. Bischl, A. L. D. Rossi, and A. C. P. Carvalho, "To tune or not to tune: Recommending when to adjust SVM HYP," in *Proc. Int. J. Conf. Neural Netw. (IJCNN)*, 2015, pp. 1–8.
[24] M. Feurer, J. T. Springenberg, and F. Hutter, "Initializing Bayesian hyper-parameter optimization via meta-learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1128–1135.
[25] C. Lemke and B. Gabrys, "Meta-learning for time series forecasting and forecast combination," *Neurocomputing*, vol. 73, pp. 2006–2016, Jun. 2010.
[26] R. B. Prudêncio and T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, pp. 121–137, Oct. 2004.
[27] A. L. D. Rossi, A. C. P. de Leon Ferreirade Carvalho, C. Soares, and B. F. De Souza, "MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data," *Neurocomputing*, vol. 127, pp. 52–64, Mar. 2014.
[28] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, and T. Ing Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognit.*, vol. 48, no. 5, pp. 1925–1935, May 2015.
[29] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion*, vol. 41, pp. 195–216, May 2018.
[30] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "META-DES. Oracle: Meta-learning and feature selection for dynamic ensemble selection," *Inf. Fusion*, vol. 38, pp. 84–103, Nov. 2017.
[31] M. A. Souza, G. D. Cavalcanti, R. M. Cruz, and R. Sabourin, "Online local pool generation for dynamic classifier selection," *Pattern Recognit.*, vol. 85, pp. 132–148, Jan. 2019.
[32] O. Wu, "Classifier ensemble by exploring supplementary ordering information," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 11, pp. 2065–2077, Nov. 2018.
[33] K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Comput. Surv.*, vol. 41, no. 1, pp. 1–25, Dec. 2008.
[34] J. R. Rice, "The algorithm selection problem," *Adv. Comput.*, vol. 15, pp. 65–118, Jan. 1976.
[35] C. Cui, M. Hu, J. D. Weir, and T. Wu, "A recommendation system for meta-modeling: A meta-learning based approach," *Expert Syst. Appl.*, vol. 46, pp. 33–44, Mar. 2016.
[36] L. Kotthoff, "Algorithm selection for combinatorial search problems: A survey," in *Data Mining and Constraint Programming* (Lecture Notes in Computer Science), vol 10101, C. Bessiere, L. De Raedt, L. Kotthoff, S. Nijssen, B. O'Sullivan, and D. Pedreschi, Eds. Cham, Switzerland: Springer, 2016
[37] J. Vanschoren, "Meta learning," in *Automated Machine Learning*. Springer, 2019, pp. 35–61,
[38] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: A survey of trends and technologies," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 117–130, Jun. 2015.
[39] L. P. F. Garcia, A. C. Lorena, S. Matwin, and A. C. P. L. F. de Carvalho, "Ensembles of label noise filters: A ranking approach," *Data Mining Knowl. Discovery*, vol. 30, no. 5, pp. 1192–1216, Sep. 2016.
[40] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proc. ICLR*, 2019, pp. 1–14.
[41] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 3, 2017, pp. 1856–1868.
[42] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 3637–3645, 2016.
[43] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 278–286.
[44] L. Zhu and Y. Yang, "Compound memory networks for few-shot video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 782–797.
[45] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artif. Intell. Rev.*, vol. 18, no. 2, pp. 77–95, 2002.
[46] G. Wang, Q. Song, H. Sun, X. Zhang, B. Xu, and Y. Zhou, "A Feature Subset Selection Algorithm Automatic Recommendation Method," *J. Artif. Intell. Res.*, vol. 47, pp. 1–34, Jul. 2018.
[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[48] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.
[49] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, and T. K. Ho, "How complex is your classification problem? A survey on measuring classification complexity," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–34, Sep. 2019.
[50] A. Rivolli, L. P. F. Garcia, C. Soares, J. Vanschoren, and A. C. P. L. F. de Carvalho, "Characterizing classification datasets: A study of meta-features for meta-learning," Aug. 2018, *arXiv:1808.10406*. [Online]. Available: https://arxiv.org/abs/1808.10406
[51] S. M. Abdulrahman, P. Brazdil, J. N. Van Rijn, and J. Vanschoren, "Speeding up algorithm selection using average ranking and active testing by introducing runtime," *Mach. Learn.*, vol. 107, no. 1, pp. 79–108, Jan. 2018.
[52] J. N. Van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The Online performance estimation framework: Heterogeneous ensemble learning for data streams," *Mach. Learn.*, vol. 107, no. 1, pp. 149–176, Jan. 2018.
[53] P. Brazdil, J. Gama, and B. Henery, "Characterizing the applicability of classification algorithms using meta-level learning," in *Proc. Eur. Conf. Mach. Learn.*, 1994, pp. 83–102.
[54] J. W. Lee and C. Giraud-Carrier, "Predicting algorithm accuracy with a small set of effective meta-features," in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 808–812.
[55] S. Young Sohn, "Meta analysis of classification algorithms for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1137–1144, Nov. 1999.
[56] A. R. S. Parmezan, H. D. Lee, and F. C. Wu, "Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework," *Expert Syst. Appl.*, vol. 75, pp. 1–24, Jun. 2017.
[57] L. P. Garcia, A. C. D. Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level," *Neurocomputing*, vol. 176, pp. 14–25, Feb. 2016.
[58] J. Luengo and F. Herrera, "An automatic extraction method of the domains of competence for learning classifiers using data complexity measures," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 147–180, Jan. 2015.
[59] T. Kam Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002.

[60] J. A. Sáez, J. Luengo, and F. Herrera, ''Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification,'' *Pattern Recognit.*, vol. 46, no. 1, pp. 355–364, Jan. 2013.

[61] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, ''Can classification performance be predicted by complexity measures? A study using microarray data,'' *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 1067–1090, Jun. 2017.

[62] H. Bensusan, ''God doesn't always shave with Occam's razor—Learning when and how to prune,'' in *Proc. 10th Eur. Conf. Mach. Learn.* Springer, 1998, pp. 119–124.

[63] H. Bensusan and C. Giraud-Carrier, ''Discovering task neighbourhoods through landmark learning performances,'' in *Principles of Data Mining and Knowledge Discovery* (Lecture Notes in Computer Science), vol. 1910, D. A. Zighed, J. Komorowski, and J. Żytkow, Eds. Berlin, Germany: Springer, 2000.

[64] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier, ''Meta-learning by landmarking various learning algorithms,'' in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 743–750.

[65] N. Tatti, ''Distances between data sets based on summary statistics,'' *J. Mach. Learn. Res.*, vol. 8, no. 1, pp. 131–154, 2007.

[66] Q. Song, G. Wang, and C. Wang, ''Automatic recommendation of classification algorithms based on data set characteristics,'' *Pattern Recognit.*, vol. 45, no. 7, pp. 2672–2689, Jul. 2012.

[67] X. Zhang, R. Li, B. Zhang, Y. Yang, J. Guo, and X. Ji, ''An instance-based learning recommendation algorithm of imbalance handling methods,'' *Appl. Math. Comput.*, vol. 351, pp. 204–218, Jun. 2019.

[68] S. Ali and K. A. Smith, ''On learning algorithm selection for classification,'' *Appl. Soft Comput.*, vol. 6, no. 2, pp. 119–138, Jan. 2006.

[69] C. Giraud-Carrier, ''The data mining advisor: Meta-learning at the Service of Practitioners,'' in *Proc. 4th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Mar. 2006, pp. 113–119.

[70] A. F. Sousa, R. B. Prudêncio, T. B. Ludermir, and C. Soares, ''Active learning and data manipulation techniques for generating training examples in meta-learning,'' *Neurocomputing*, vol. 194, pp. 45–55, Jun. 2016.

[71] P. B. Brazdil, C. Soares, and J. P. da Costa, ''Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results,'' *Mach. Learn.*, vol. 50, no. 3, pp. 251–277, Mar. 2003.

[72] A. Kalousis and T. Theoharis, ''NOEMON : An intelligent assistant for classifier selection,'' *Intell. Data. Anal.*, vol. 3, pp. 319–337, Aug. 1999.

[73] Y. Peng, P. Flach, P. Brazdil, and C. Soares, ''Decision tree-based data characterization for meta-learning,'' in *Proc. 2nd Int. Workshop Integr. Collaboration Aspects Data Mining, Decis. Support Meta-Learn. (IDDM)*, vol. 3, 2002, pp. 111–122.

[74] M. Reif, F. Shafait, M. Goldstein, T. Breuel, and A. Dengel, ''Automatic classifier selection for non-experts,'' *Pattern Anal. Appl.*, vol. 17, no. 1, pp. 83–96, Feb. 2014.

[75] H. Bensusan and A. Kalousis, ''Estimating the predictive accuracy of a classifier,'' in *Proc. Eur. Conf. Mach. Learn. (ECML)*, 2007, pp. 25–36.

[76] K. A. Smith, F. Woo, V. Ciesielski, and R. Ibrahim, ''Matching data mining algorithm suitability to data characteristics using a self-organizing map,'' *Hybrid Information Systems* (Advances in Soft Computing), vol. 14, A. Abraham and M. Köppen, Eds. Berlin, Germany: Physica, 2002.

[77] J. W. Lee and C. Giraud-Carrier, ''Automatic selection of classification learning algorithms for data mining practitioners,'' *Intell. Data Anal.*, vol. 17, no. 4, pp. 665–678, Jun. 2013.

[78] J. V. F. Hutter and L. Kotthoff, *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019, pp. 1–219.

[79] M. A. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles, *Instance Spaces for Machine Learning Classification*, vol. 107. New York, NY, USA: Springer, 2018.

[80] F. Pinto, C. Soares, and J. Mendes-Moreira, ''Towards automatic generation of metafeatures,'' in *Proc. Adv. Knowl. Discovery Data Mining*, 2016, pp. 215–226.

[81] R. C. Teamn, *R: A Language and Environment for Statistical Computing, R Version 3.5.1*, R Found. Stat. Comput., Vienna, Austria, 2018.

[82] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ''The WEKA data mining software: An update,'' *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[83] K. Hornik, C. Buchta, T. Hothorn, A. Karatzoglou, D. Meyer, and A. Zeileis, *R/Weka Interface, Version 0.4-41*, document 41, 2019, pp. 1–34.

[84] J. Pizarro, E. Guerrero, and P. L. Galindo, ''Multiple comparison procedures applied to model selection,'' *Neurocomputing*, vol. 48, nos. 1–4, pp. 155–173, Oct. 2002.

[85] T. M. Mitchell, *Machine Learning* (Computer Science), 1st ed. New York, NY, USA: McGraw-Hill, 1997.

[86] W. Liu and L. Lü, ''Link prediction based on local random walk,'' *Eur. Phys. Lett.*, vol. 89, no. 5, Mar. 2010, Art. no. 58007.

[87] A. P. Dempster, N. M. Laird, and D. B. Rubin, ''Maximum likelihood from incomplete data via the EM algorithm,'' *J. Roy. Stat. Soc. B Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

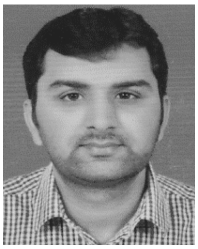
[88] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, ''Efficient and robust automated machine learning,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 2962–2970.

[89] J. Vanschoren, H. Blockeel, B. Pfahringer, and G. Holmes, ''Experiment databases: A new way to share, organize and learn from experiments,'' *Mach. Learn.*, vol. 87, no. 2, pp. 127–158, May 2012.

[90] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, ''Feature selection for multimedia analysis by sharing information among multiple tasks,'' *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

IRFAN KHAN received the master's degree in computer science from COMSATS University, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Software, Dalian University of Technology, Dalian, China. He has served as a data analyst at two international organizations. He also served as a Visiting Lecturer with the University of Swat. His research interests include machine learning, data mining, meta-learning, algorithm selection, classification, and Recommender systems.

XIANCHAO ZHANG received the bachelor's and master's degrees in mathematics from the National University of Defense Technology, China, in 1994 and 1998, respectively, and the Ph.D. degree in computer science from the University of Science and Technology of China, in 2000. From 2000 to 2003, he was a Research and Development Manager in some international companies. He joined the Dalian University of Technology, in 2003, where he is currently a Full Professor. His research interests include design and analysis of algorithms, machine learning, data mining, and information retrieval.



MOBASHAR REHMAN received the Ph.D. degree in information technology from Universiti Teknologi Petronas (Knowledge Management). He is currently working as Assistant Professor with the Department of Information Systems, Universiti Tunku Abdul Rahman, Kampar, Malaysia. His research interests include knowledge management, knowledge sharing, human factors in software engineering, machine learning, and cyberpsychology.

RAHMAN ALI received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2016. He was with the Laboratoired'Informatique de l'Universite du Maine, France, as a Research Assistant. He is currently an Assistant Professor with the University of Peshawar, Pakistan. He has authored over 40 publications and conference contributions. His main scientific research interests include machine learning, data mining, recommendation systems, reasoning and inference, and natural language processing.