AutoFe-Sel: A Meta-learning based methodology for Recommending Feature Subset Selection Algorithms

Irfan Khan[1], Xianchao Zhang[1], Ramesh Kumar Ayyasamy[2*] and Rahman Ali[3]
[1] School of Software, Dalian University of Technology, Dalian 116620, China
[e-mail: irfankhan2021@mail.dlut.edu.cn , xczhang@dlut.edu.cn]
[2] Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar 31900,
Malaysia [e-mail: rameshkumar@utar.edu.my]
[3] QACC, University of Peshawar, Peshawar 25120, Pakistan [e-mail: rehmanali@uop.edu.pk]
[*]Corresponding author: Ramesh Kumar Ayyasamy

Abstract

Automated machine learning, often referred to as "AutoML," is the process of automating the time-consuming and iterative procedures that are associated with the building of machine learning models. There have been significant contributions in this area across a number of different stages of accomplishing a data-mining task, including model selection, hyper-parameter optimization, and preprocessing method selection. Among them, preprocessing method selection is a relatively new and fast growing research area. The current work is focused on the recommendation of preprocessing methods, i.e., feature subset selection (FSS) algorithms. One limitation in the existing studies regarding FSS algorithm recommendation is the use of a single learner for meta-modeling, which restricts its capabilities in the meta-modeling. Moreover, the meta-modeling in the existing studies is typically based on a single group of data characterization measures (DCMs). Nonetheless, there are a number of complementary DCM groups, and their combination will allow them to leverage their diversity, resulting in improved meta-modeling. This study aims to address these limitations by proposing an architecture for preprocess method selection that uses ensemble learning for meta-modeling, namely AutoFE-Sel. To evaluate the proposed method, we performed an extensive experimental evaluation involving 8 FSS algorithms, 3 groups of DCMs, and 125 datasets. Results show that the proposed method achieves better performance compared to three baseline methods. The proposed architecture can also be easily extended to other preprocessing method selections, e.g., noise-filter selection and imbalance handling method selection.

1. Introduction

Data analysis is a multistep process that employs algorithms for each step, such as preprocessing like data labeling, cleaning, handling imbalanced classes, and feature selection before training a base model on the data. The given diversity of data analysis tasks and large number of available ML algorithms pose a significant limitation, i.e., how to select adequate algorithms for a given problem from the large set of available candidate algorithms [1]. The process of choosing an adequate algorithm for each step of the multistep process of data analysis is an iterative, and nontrivial task, formally known as the "algorithm selection problem" (ASP) in literature [2]. To tackle ASP, a significant amount of effort has been devoted to automating the algorithm selection procedure. Automated machine learning or "Auto-ML," is the practice of automating the time-consuming and iterative procedures that are associated with the building of machine learning models [3]. Its major objective is to decrease human effort in constructing accurate prediction models, promote early deployment of optimal options, and save time and resources without compromising on accuracy.

In Auto-ML, methods that are based on meta-learning have been extensively researched and have shown substantial success with regard to algorithm selection. Meta-learning is a broad field and has many important and diverse research directions across multiple domains. Generally, it can be defined as the process of learning from past experience gathered through the application of learning algorithms to a wide range of data sets, with the end goal of minimizing the amount of time required to learn new tasks [4].

In order to automate algorithm selection, the meta-learning approach for algorithm recommendation is based on learning from dataset properties known as data characterization measures (DCMs) and previous model evaluations. The DCMs determine what properties various learning tasks share that make certain algorithms more suitable for learning them.

Meta-learning based Algorithm recommender systems are mainly comprised of two major components. 1 Accumulation of Meta-data 2 Meta-modeling. The first component involves the evaluation of the performance of algorithms on a diverse range of datasets in a specific domain and the extraction of DCMs from the dataset. The first component is computationally the most expensive part of developing any algorithm recommendation system. A large part of the work that goes into these recommender systems is the methodical gathering of dataset properties, i.e., DCMs, and the assessment of the performance of various machine learning (ML) algorithms on those datasets. The second component involves building a model on the meta-data such that it maps the DCMs into the performance evaluation measures. For, any given dataset, first the DCMs are extracted and given as input to the meta-model, which in turn recommends appropriate algorithms according to the learned meta-model. This meta-learning approach for algorithm selection has shown substantial success for building algorithm recommender systems in various domains, for instance, classifier selection [5], clustering method selection [6], time-series selection [7], and preprocessing method selection [8]. Auto-Sklearn [9] and Auto-Weka [10] are two well-known examples of meta-learning-based algorithm selection.

The selection of preprocessing methods is a relatively new but rapidly expanding research area in Auto-ML. Since preprocessing has a major role and is regarded as one of the most costly steps, it could account for 50%–80% of the entire process of data analysis [11]. Its proper planning and execution are critical for ensuring high-quality input data. The current study is focused on preprocessing method selection, more specifically the FSS algorithm recommendation.

The notable existing works of meta-learning based preprocess method recommendation includes [11]–[16]. A major limitation of these studies is the use of a single learner for meta-modeling, which restricts its capabilities in the meta-modeling, as acknowledged in a recent work [17]. For instance, the authors in [11] [12] has used KNN for meta-modeling, similarly, the meta-model in [13]–[15] is built on decision trees. In addition, the meta-modeling in the existing studies is typically based on a single group of DCMs. Nonetheless, there are a number of complementary DCM groups, and their combination will allow them to leverage their diversity, resulting in improved meta-modeling.

This study aims to address these limitations by proposing an architecture for preprocess method selection that uses ensemble learning for meta-modeling, namely AutoFE-Sel. The following is an outline of the contributions made by this work:

The AutoFE-Sel is based on stacking an framework [18]. It has two key advantages. (1) It improves the meta-modeling by combining multiple ML-KNN weak learners; (2) It takes advantage of the diversity of the existing complementary groups of DCM by generating their various alternative combinations, subsequently leveraging the performance of the system. The proposed architecture is compared with three other baseline methods in a large experimental evaluation composed of 125 datasets and 12 FSS algorithms. The results demonstrate that our approach outperforms the other FSS algorithm recommendation approaches. In addition, our architecture is component extendable and can be readily extended for the recommendation of algorithms for various preprocessing methods, such as noise-filter selection and imbalance handling methods. Experimental details are provided in the GitHub repository[3].

The remaining paper is structured as follows: In Section 2, we critically analyzed and summarized the existing similar works. The architecture of the proposed AutoFE-Sel is briefly described in Section 3. In Section 4 of the paper, we present details regarding the experimental setup along with the presentation and discussion of results. Finally, we conclude the paper in Section 5.

2. Background and Related Works

In a typical supervised learning task, a dataset is composed of independent variables called features and a target variable class. The quality of the features has a direct effect on the accuracy of the learning task [19]. It is preferable for the learnt model to have low variance, meaning that it does not over fit the training data and retains its ability to generalize on new instances. Usually, data extracted from various sources contains inconsistencies and has redundant, noisy, and irrelevant features that increase its complexity and computational cost. It is vital to eliminate redundant, noisy, and irrelevant characteristics in order to utilize just the most informative ones, hence minimizing the model's variance and maximizing its generalizability. Consequently, preprocessing methods like feature selection play a pivotal role and are considered one of the most costly processes, accounting for 50-80% of the whole data mining process [13]. To guarantee high-quality input data, good planning and execution are essential. Therefore, automation of preprocessing method selection has recently been the focus of research in AutoML.

The most recent and relevant extensive literature surveys in AutoML [5] highlight two key research directions in meta-learning-based algorithm recommendation. Which are: (1) DCM (2) Meta-modeling. Research focusing on Data Characterization Measures (DCM) is concerned with the investigation of techniques for extracting measures of data characterization that are consistent over a given domain and are able to correlate the data distribution of datasets with the inherent fixed-bias of algorithms. The DCM's that are utilized in a meta-learning

systems depend on the problem domain [20]. This is because the DCM's have to capture properties that have the potential to predict the performance of the machine learning algorithms [3]https://github.com/iyousafzai1/AutoFe-Sel under consideration. DCM's being an integral and challenging component of algorithm recommender systems had remained the focus of research for a long time and significant success has been achieved in this regard [21]. Various groups of DCMS's have been proposed in various application domains e.g. classification [22] [23], clustering [24], regression [25], and time-series [26]. With regard to preprocessing method selection for classification, the DCM's measures that are been proposed and empirically evaluated in the literature include complexity based feature overlapping measures, statistical and information theoretic based, model based. Authors in [27] and [28], had provided an excellent presentation and evaluation of DCMs for classification tasks.

The process of mapping the DCM measures to the performance evaluation information of the candidate algorithms at meta-level is known as meta-modeling. Usually, a machine learning model is employed for this purpose, which is referred to as meta-learner in literature. In previous works, a wide range of Ml algorithms are being investigated for meta-modeling for instance, KNN, MLK-NN, rule based. Each of these meta-modeling approaches have its own merits and demerits and depends on the suitability to the meta-data and application domain under consideration. Latest comparative study shows that employing MLK-NN has many advantages compared to the others [5].

With regard to meta-learning for FSS algorithm recommendation, the first major study was conducted by the authors in [11]. They employed KNN as meta-learner and the meta-data was generated from 22 candidate FSS algorithms and 115 datasets in order to provide a ranking of the best performing algorithms on a given problem. The ranking was provided through the direct mapping of the top nearest neighbors to the candidate algorithms based on a multi-criteria performance metric comprising of accuracy and run time of algorithms. Another study [12]; used a similar method for choosing an FSS algorithm, but in addition to statistical and information measures, they also used model-based DCMs. Calculating model-based DCMs involves expressing the dataset in a unique structure, such as a decision tree, in order to gain insight into the learning complexity. The primary drawback of employing KNN as a meta-learner is determining the optimal value for the parameter K, which is fixed at the meta-level but varies for various datasets in practice, thereby impacting the system's performance. [5].

Employing rule-based models at meta-level is another meta-learning strategy for algorithm selection reported in [13]. A decision tree based learner J48 was trained on meta-data obtained from 150 datasets described by statistical information theoretic and complexity based characterization measures and mapped into a group of four FSS algorithms. In another study [14], the authors built a meta-learning framework for FSS algorithms by employing the rule based J48 as meta-learner. It also confirmed the usefulness of employing rule based learners at meta-level. In addition, authors in [15], investigated another rule based C4.5 for preprocessing method selection. The interpretability of rule-based learners at the meta-level, which translates to the ability to analyze the rules that led to the selection of an algorithm, is an evident advantage. When used as a meta-learner in a meta-learning setup, extensive empirical evaluation of algorithm selection studies indicates that it cannot compete with other methods in terms of accuracy.

Other works for preprocessing methods recommendation based on meta-learning includes, noise filter selection [8], imbalance handling methods [29], [30]. In addition to these works, in other closely related works, various research groups contributed to the literature by studying the intrinsic relationship among DCMS and how it affects performance measures like accuracy

and time complexity in FSS. For instance, authors in [31] evaluated standard, statistical, and information-theoretic-based DCMS on five filter-based FSS techniques induced on three base classifiers. Likewise authors in [32], investigated complexity based DCMS for estimating feature importance. The summary of related works is provided in **Table 1** which elaborate the main points of reviewed literature.

Table 1. Summary of reviewed literature

Ref	Application Domain	#Datasets	#Algo	Dataset Characterization measures	Meta-Modeling
[12]	Feature selection	84	16	Statistical and information theoretic, Model based	Instance Based (IBK)
[11]	Feature selection	115	22	Statistical and information theoretic	Instance Based (IBK)
[13]	Feature selection	150	4	Statistical and information theoretic, complexity based	J48
[16]	Feature selection	100	4	Statistical and information theoretic	Fuzzy similarity
[15]	Noise Filter selection	34	8	Complexity based measures	C4.5 Decision Tree
[8]	Noise Filter selection	53	6	Statistical and information theoretic, Complexity based, Model based, Land-marking	k-NN,RF and SVM Regression
[29]	Imbalance handling methods selection	80	11	Statistical and information theoretic Complexity based, Model based, Land-marking	Instance Based

   The literature review reveals that prior meta-modeling research have employed a single learner, which, as stated in a recent work [17], limits its potential. In addition, a number of DCM groups are complimentary, and their combination will allow them to capitalize on their variety, resulting in enhanced meta-modeling. This paper proposes AutoFE-Sel, an architecture for preprocess method selection that employs ensemble learning for meta-modeling to address these limitations.

   According to a recent study [17], the use of a singular learner in prior meta-modeling research restricts its potential. In addition, a number of DCM groups are complementary, and the combination of these groups will enable them to capitalize on their diversity, resulting in improved meta-modeling. AutoFE-Sel, an architecture for preprocess method selection that employs ensemble learning for meta-modeling, is proposed to resolve these limitations.

3.   Architecture of AutoFE-Sel

The architecture of the ensemble-based learning for FSS algorithm recommendation is shown in **Fig. 1**. Details of each subcomponent in the architecture can be found in the next subsections. There are two fundamental prerequisites for applying ensemble learning, namely: (1) The base models should be accurate. 2. The models should be independent and diverse. Current recommendation models in the literature are constructed using several techniques, such as KNN [11], ML-KNN [33], and rule-based learners, i.e., J48 and C4.5 [34]. Even though there are variations in the recommendation performance of these models, they are still able to narrow down the search space of candidate algorithms for any given problem and have reasonable recommendation accuracies. With regard to assessing the second prerequisite, the suitability

of ensemble learning in a meta-learning setup, existing research has demonstrated that the correlation among different groups of DCMs is low and that models built on different types of DCMs are independent and diverse [17]. Moreover, a recent study [18], focusing on classification algorithm recommendation, has also confirmed that the use of ensemble learning leverages the diversity of DCMs and improves the recommendation accuracy. These observations motivated the proposal of the AutoFE-Sel architecture, an ensemble-based meta-learning method for FSS algorithm selection.
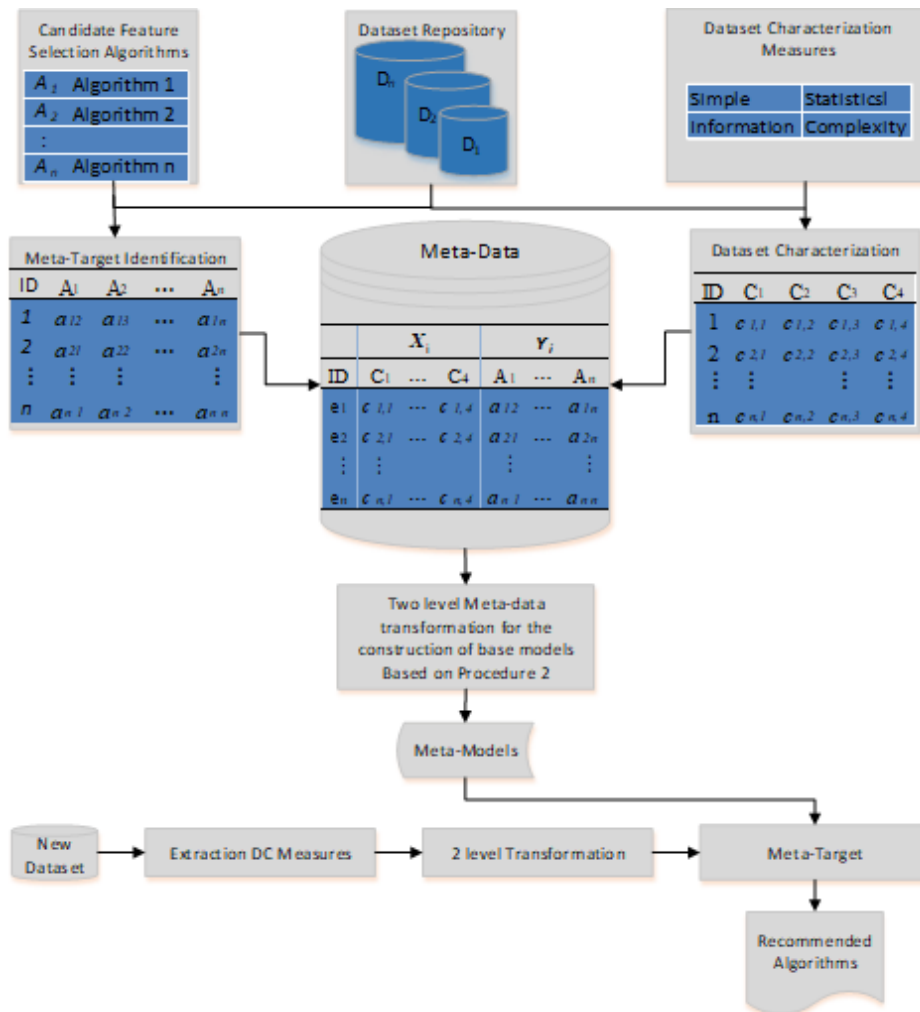

Fig. 1. Architecture of AutoFE-Sel

The graphical presentation of the AutoFE-Sel is shown in **Fig. 1**. It consists mainly of three steps. 1 collection of metadata Meta-modeling 3. Algorithm recommendation. The collection of meta-data involves the extraction of DCMs and the identification of meta-targets. Then, in the next phase of the construction of the meta-models, a two-level data transformation method is adopted to form meta-models based on meta-data. Finally, after the construction of meta-models, algorithm recommendation for any given problem involves first extracting its DCMS

and then applying the two-level data transformations, which are then fed into the meta-models. That subsequently recommends an adequate set of algorithms.

Finally, at the last phase, for any new dataset instance, first its DCM is extracted, and then a recommendation on an adequate subset of candidate algorithms is provided, guided by the learned meta-models. Each of these steps is described in detail in the following subsections.

Table 2. Description of Notations used in this study

Notations	Description
P:	the n diverse range of problems(datasets) $\{P_i \mid i = 1,2,...,n\}$
A:	the set of q candidate classification algorithms $\{a_i \mid j = 1,2,...,q\}$
P:	$\{p_1, p_2, ... p_n\}$ denotes diverse range of representative classification problems P_i
X_i:	$\{x_1, x_2, ... x_n\}$ represents DCMs of P_i
Y_i:	$\{y_1, y_2, ... y_n\}$ represents meta-target of P_i
Z_i:	$\{a_1, a_2, a_3\}$ denotes the algorithms recommended by AutoFE-Sel for a problem P_i
E:	$\{X_i, Y_i\}$ where $\{E_i \mid i = 1,2,...,n\}$ represents meta-data
D^I:	Level-1 Meta-datasets
D_k^I:	Level-1 Meta-dataset produced by the k_{th} combination of DCM types
D^{II}:	Level-2 Meta-datasets
D_j^{II}:	Level 2 Meta-dataset for the j_{th} candidate FSS algorithm

For the purpose of clarity and understanding, we provide in **Table 2**, the notations used throughout the rest of the study.

3.1.1 Meta-data collection

The collection of meta-data involves estimation of meta-target and extraction of DCMs. Both these steps are described in detail in the following subsections.

3.1.2 Meta-Target

Meta-target concern with that for each $p_i \in P$, identify the best subset of appropriate algorithms among the pool of available candidate algorithms A. It requires performance evaluation of all the candidate algorithms on each dataset at meta level. In this work, we have adopted multi-criteria metric EARR (Adjusted ratio of ratios) from [11], for performance evaluation. It takes into account other necessary factors besides accuracy when selecting an algorithm, i.e., run time and number of selected features. The multi-criteria performance EARR metric for performance evaluation of a_i with reference to a_j on dataset p_k is given by

$$EARR_{a_i,a_j}^{p_k} = \frac{acc_i^k / acc_j^k}{1 + \alpha \cdot \log(t_i^k / t_j^k) + \beta \cdot \log(nf_i^k / nf_j^k)} \quad (1 \le i \ne j \le E, 1 \le k \le n) \qquad (1)$$

Since accuracy of FSS algorithm cannot be directly calculated therefore a base algorithm usually classifer is used for this purpose. It takes into account the accuracy of a FSS algorithm induced by a classifier, the number of selected features, and the run time. Here acc_i^j represents accuracy of FSS A_i induced by a base classifier on a dataset $p_j (1 \le i \le M, 1 \le j \le N)$. Where t_i^k and nf_i^k denote the runtime and number of features selected by a FSS algorithm.

Moreover α and β represents user provided parameters in order to tradeoff for accuracy and number of selected features. The heigher value of $EARR_{a_i, a_j}^{P_k}$ with reference to $EARR_{a_j, a_i}^{P_k}$ indicate that a_i performs better than a_j on dataset P_k. In order to compare an algorithm a_i with rest of all available candidate algoithms i.e., $\{A\} - \{a_i\}$, then the following equation is used.

$$EARR_{a_i}^{P} = \frac{1}{E-1} \sum_{j=1 \wedge j \neq i}^{M} EARR_{a_i, a_j}^{P} \tag{2}$$

After the calculation of $EARR$ metrics values, the next step is the estimation of meta-target. Usually, it involves a statistical test procedure; although there are various statistical tests used in prior studies, it is generally agreed that the non-parametric multiple comparison procedure (MCP) Friedman followed by Holm procedure test is the most suitable [35]. For example, for any given dataset $P_i \in \mathrm{P}$, the algorithm among the candidates that performs best on the given problem in terms of the EARR metric is chosen as a reference, and the Holm procedure test is then used to identify algorithms whose difference in performance is not statistically significant from the referenced algorithm. The referenced algorithm, along with the statistically equivalent ones in performance, are considered appropriate algorithms, i.e., meta-target. These algorithms constitute the multi-label meta-target $Y_i = \{y_{i,j} | 1 \leq j \leq q\}$ of p_i and label $y_{i,j} = 1$ or 0 shows if the algorithm a_j is suitable for p_i or not. This step is performed for all the datasets, and each meta-example is labeled with 0 or 1.

3.1.3 Data Characterization Measures

The DCMs used in this study are given in **Table 3**. The DCMs are extracted in a standard and unified way through the standard R library built upon ECoL1 framework for complexity measures [27], and for the remaining group of DCMs we have used the standard R library built upon MFE[2] framework [37]. The complexity-based features determine the relevance of the features in identifying and separating the classes. Feature overlap measures describe how informative the available features are to separate the classes, i.e., they assess the discriminative power of the attributes and features.

Table 3. DCMs used in AutoFE-Sel

Category	Measure description
Complexity Measures	Maximum Fisher's Discriminant Ratio (F1) Volume of Overlapping Region (F2). Maximum Individual Feature Efficiency (F3). Collective Feature Efficiency (F4)
Statistical based properties	Mean absolute linear correlation coefficient of all possible pairs $= \bar{\rho}(X, Y)$
	Mean skewness $= \overline{Skew}(X)$
	Mean kurtosis $= \overline{Kurt}(X)$
Information-theoretic properties	Normalized class entropy $= H(C)_{norm}$
	Mean normalized feature entropy $= \overline{H}(X)_{norm}$
	Mean mutual information of class and attribute $= \overline{MI}(C, X)$
	Maximum mutual information of class and attribute $= MI(C, X)_{max}$
	Equivalent number of features, $EN_{attr} = H(C)/\overline{MI}(C, X)$
	Noise-signal ratio, $NS_{ratio} = (\overline{H}(X) - \overline{MI}(C, X))/\overline{MI}(C, X)$

The F1 measure calculates the inter-class to intra-class dispersion ratio of each feature. Lower values of this measure indicate the presence of at least one feature whose values demonstrate minimal interclass correlation. This measure is very informative, especially when the probability distributions of the classes are close to normal. It is calculated as follows:

$$F1 = \frac{1}{1 + \max_{i=1}^{m} r_{fi}} \qquad (2)$$

Where r_{fi} represents the discriminant ratio of every feature f_i and can be computed as follows

$$r_{fi} = \frac{\sum_{j=1}^{n_c} n_{cj} (\mu_{cj}^{fi} - \mu^{fi})^2}{\sum_{j=1}^{n_c} \sum_{l=1}^{n_{cj}} (x_{li}^{j} - \mu_{cj}^{fi})^2} \qquad (3)$$

Where n_{cj} represents the number of examples for class c_j, μ_{fi} denotes the mean of f_i values across all the classes and the value of feature f_i for any instance in class c_j is represented by x_{li}^{j}.

The F2 assesses the degree to which the distributions of feature values overlap between classes. It is calculated by identifying the maximum and minimum class values for each feature f_i. Following this, the overlapping interval is approximated and normalized by the range of values in each class, as shown below.

$$F2 = \prod_{i}^{m} \frac{\text{overlap } (f_i)}{\text{range}(f_i)} = \prod_{i}^{m} \frac{\max\{0, \text{minmax}(f_i) - \text{maxmin}(f_i)\}}{\text{maxmax}(f_i) - \text{minmin}(f_i)} \qquad (4)$$

Where,

$$\text{minmax}(f_i) = \min\left(\max(f_i^{c_1}), \max(f_i^{c_2})\right)$$
$$\text{maxmin}(f_i) = \max\left(\min(f_i^{c_1}), \min(f_i^{c_2})\right)$$
$$\text{maxmax}(f_i) = \max\left(\max(f_i^{c_1}), \max(f_i^{c_2})\right)$$
$$\text{minmin}(f_i) = \min\left(\min(f_i^{c_1}), \min(f_i^{c_2})\right)$$

The $\min(f_i^{c_j})$ and $\max(f_i^{c_j})$ represent the highest and lowest values of each feature in the class c j (1, 2).

The F3 metric assesses the individual effectiveness of each feature in identifying the classes. In doing so, it examines if there is overlap of values among instances of various classes for each feature.

$$F3 = \min_{i}^{n} \frac{n_o(f_i)}{n} \qquad (5)$$

where $n_o(f_i)$ is the number of instance in the overlapping region for feature f_i and is calculated according to equation 6.

$$n_o(f_i) = \sum_{j=1}^{n} I(x_{ji} > maxmin(f_i) \wedge x_{ji} < minmax(f_i)) \qquad (6)$$

The F4 measure presents an overview of how the features interact and work together. It employs a process similar to that used for F3 in stages. Initially, the highest discriminative feature based on F3 is chosen i.e., the feature with the least overlap across various classes. All instances that can be separated by this feature are excluded from the dataset and the previous step is performed again after which the subsequent most discriminative feature is chosen. This approach is carried out until all features have been analyzed and can be terminated when there is no remaining instance. F4 is calculated after l iterations across the dataset, where l is a positive integer in the range [1,m].. If any one of the features is sufficient to differentiate among all the instances in task T, l is 1. however it can be as high as m if all of the features need to be taken into consideration. F4 is calculated as follows:

$$F4 = \frac{n_o(f_{min}(T_l))}{n} \qquad (7)$$

Where $n_o(f_{min}(T_l))$ estimate the overlapping region of feature f_{min} for the dataset from the lth iteration of (T_l).

The statistical measures provide details concerning the distribution of dataset e.g., central tendency and dispersion whereas the entropy based information-theoretic measures show variability and redundancy of the attributes. a detailed overview of DCM's and how they are calculated can be found in [28]. Due to space limitation we here present only the measures in Table 2 and for more information on the theoretical and practical calculation of each these measures, we refer readers to [27] and [36].

The outcome of the first component is the accumulation of meta-data in i.e., E = $\{e_1, e_2, e_3, \dots e_n\}$ in which every instance of the meta-data corresponds to the DCM and meta-target of the respective dataset i.e., $m_i = (X_i, Y_i)$. Where X_i is the three groups of DCM i.e. $Dc_1, Dc_2. Dc_3$, each group comprising of sub-measures and $Y_i = \{y_1\, y_{2\dots} y_n\}$ is the meta-target, labeled with 0 and 1.

3.2   Meta-Models Construction

In order to transform the collected meta-data to a meta-level learning form such that various base models can be formed on them, we have employed a two level data transformation procedure as shown in procedure 1.

3.2.1 Level 1 Transformation:

In order to appropriately leverage the diversity of the different groups of DCMs we generated their various combinations. A function Choose (X_i) is used for this purpose. The combination function (X_i) generate t = $2^q - 1$ different combinations on various q groups DCMs. For instance, if the $X_i = \{1, 2, 3\}$ contains three groups of DCMs then the combination function will produce t = $2^3 - 1 = 7$ {{1},{2},{3},{1,2},{1,3},{2,3},{1,2,3}}. First for

each meta-example M_i , $=$ (X_i, Y_i) the choose function is applied choose(X_i) to generate various combinations of DCMs.

Procedure 1: Level-2 Transformation

Input : $Metadata\ E = \{e_1, e_2, e_3, \dots e_n\}, where\ e_i = (X_i, Y_i)$
$D^I = \{D_j^I | 1 \leq j \geq 7\}\}$
Output: Level-2 Meta-data

1	Begin
2	**foreach** $E_i \in E$ **do**
	foreach label $y_{i,j} \in Y_i$ **do**
	foreach $D_k^i \in D^I do$
3	$Prob_{i,j,k}$ = ML-KNN $(X_i, y_{i,j}, D_i^k)$;
	endfor
	endfor
4	**endfor**
5	$D^{II} = NULL\ \emptyset$
6	***for*** *j = 1 to q* ***do***
7	$D_j^{II} = Null$
	for *i = 1 to n* **do**
8	$inst_i = (< Prob_{i,j,k}, Prob_{i,j,k}, Prob_{i,j,k} > y_{i,j})$
9	$D_j^{II} = D_j^{II}\ U\ inst_i$
10	$D^{II} = D^{II} = D_j^{II}$
11	**end for**
12	**Return** D^{II}

3.2.2 Level 2 Transformation:

A second-level transformation is adopted in order to transform the meta-data into a form appropriate for ensemble-based meta-learning. On each Tier-1 dataset, ML-KNN learning calculates the probability $prob_{i,j}$ of each label $y_{i,j} \in Y_i$ for every instance $M_i \in M$. Hence for $y_{i,j}$ 7 probability values were generated using 7 Level-1 datasets. For each label i.e., for each candidate algorithm, we created a new Level-2 dataset D^{II}. These seven probabilities are used as independent variables in the construction of each instance in D^{II}, with the value of $y_{i,j} \in Y_i$ serving as the class label. This permits the generation of q Level-2 datasets for q labels, with n instances per dataset.

The pseudocode for D^{II} dataset creation is given in Procedure 1. It takes as input, the meta-dataset denoted by E, along with and level-1 datasets denoted by D^I. The first step is to apply ML-KNN to all level-1 datasets, which is done in lines 1 through 7. This will yield the probabilities for every label $Y_{i,j}$ in the target Y_i. The level-2 dataset is then constructed one at a time for each label, beginning at line 9 and continuing until line 16. The i_{th} instance $inst_i$ in D^{II} as indicated in line 12, takes probability values learnt by ML-KNN for label $y_{i,j} \in Y_i$ on the 7 level-1 datasets. Accordingly, line 17 delivers the Tier-2 dataset D^{II}.

In order to assist in further understanding of the generation of D^{II} dataset, we present a simple illustration overview in the following figure. In **Fig. 2**, E represents meta-data, containing the three types of dataset characterization measures i.e., DCM_1, DCM_2, DCM_3 and

the corresponding meta-targets Y, extracted from the problem set P. In order to generate the level-1 dataset D^I, the three types of data characterization measures are combined in various combinations along with the meta-target labels Y.
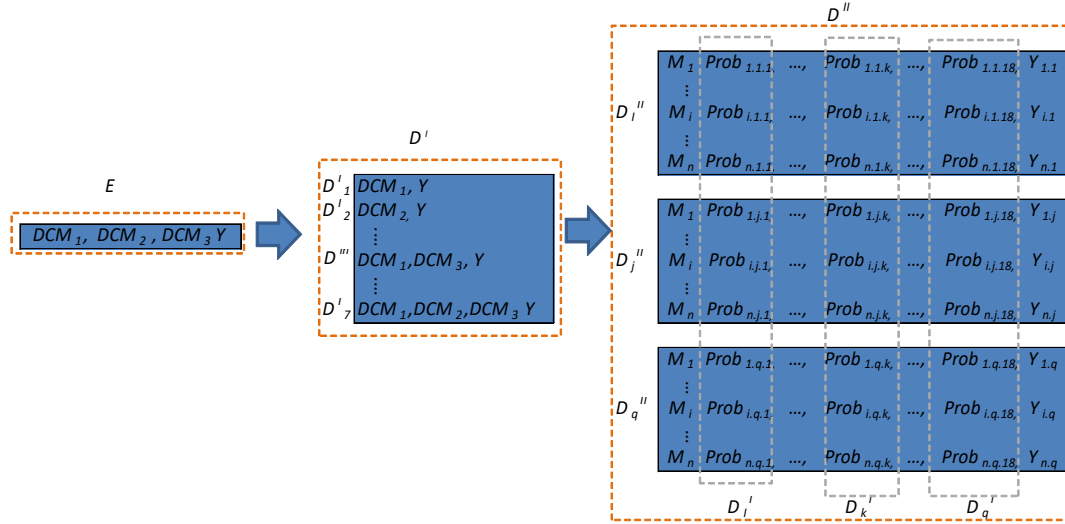

Fig. 2. Level 2 Transformation

Regarding the level-2 dataset transformation, there are q single label datasets in D^{II}. Each instance in dataset $D^{II}j$ has the probability values $Prob_{i,j,k}$ as a single label. ML-KNN is trained on D_k^i to estimate the probability that the j$_{th}$ label corresponds to the i$_{th}$ occurrence in E.

As stated earlier, the single-label Tier-2 datasets are created through the two-layer transformation. Using Tier-2 datasets, it is now possible to construct binary classification models. We used AdaBoost to create q binary classification models for the q labels, employing C4.5 as the basis classifier. We use CFS with BestFirst search technique for Tier-2 datasets to increase the generalization ability of binary models.

3.3   Recommendation

The process of recommending a subset of appropriate FSS algorithms for any new dataset P_{new} involves performing the steps listed here. 1. The first step is to extract the three types of DCMs from the dataset. 2. Using the Choose function, produce the seven sets of possible combinations from the four types of DCMs. 3. Using ML-KNN learning, calculate the 7 probability values for every set of DCMs generated in previous step. 4. After being provided with the probability values, every of the q meta models will return a binary classification result, indicating whether the each of the candidate FSS is appropriate for P_{new} or not. 5. Finally, recommend the set of appropriate algorithms, predicted to be appropriate for P_{new} according to the prediction in the previous step.

4.   Experimental Setup and Results

In this section, we briefly present the experimental setup and performance evaluation comparision of the proposed method with baseline methods. The exact experimental setup is provided in the github repository[3].

4.1   Meta-Data Collection

4.1.1 DCMs and Datasets

Adhering to the current practices of algorithm recommendation and in accordance to the guidelines of [5] in meta-learning we have included 125 standard classification problems. Details regarding the datasets i.e., number of instances, features and classes are provided in the GitHub repository[3]. The DCMs are extracted in a standard and unified way thorugh the standard R library built upon ECoL[1] framework for complexity measures [27], and for the remaining group of DCMs we have used the standard R library built upon MFE[2] framework [37]. These DCMs has already demonstrated success in meta-learning with regard to the selection of various processing methods, for instance FSS algorithms [32] [13][11] [31], noise filters [8] and imbalance handling methods [29], [30]. For the sake of reproducability of our experiments we have provided all the relevant details on github library[3].

4.1.2 Candidate Algorithms

The candidate FSS algorithms are shown in **Table 4**. A Consistency-Based Filter, often known as CBF, is one that looks for a small feature subset that is highly consistent with the class. The objective of correlation-based feature selection, often known as CFS, is to identify a subset of features that have a high correlation to the class while having a low correlation amongst features. Concerning Rank Search, it utilizes a feature evaluator, such as the gain ratio and ranks all the features. Once a feature evaluator has been chosen, a rating list is constructed using a forward selection search. Info-Gain calculates an estimate of the information gain, based on entropy, for each individual feature. Within a sample of examples, Relief-F rewards features that accounts for differentiating instances from other classes. Sequential forward subset that yields the greatest value of the objective function. search (SFS) begins with an empty set and iteratively adds the feature to the current feature

Moreover, as discussed previously, the accuracy of FSS algorithm on a classification task cannot be directly calculated and classification algorithms are necessary to evaluate its performance. However, the inherent bias of a classifier may favor a certain FSS algorithm some datasets [11]. Therefore, for the sake of generalization and fair comparison, we have employed five standard classifiers which are KNN, PART, J48, Navie Bayes and Bayes Network. The results presented in this section are based on the average of the FSS algorithms on these five classifiers. In addition, we have used 10×10 fold-cross-validation for the performance estimation of all FSS algorithms induced by each classifier on all datasets at meta-level. For the implementation of all the candidate FSS algoithms and base classifiers, the open source java based WEKA was used with their default parameters [38].

[1]https://cran.r-project.org/web/packages/ECoL,[2]hhttps://cran.r-project.org/web/packages/mfe,
[3]https://github.com/iyousafzai1/AutoFe-Sel

Table 4. Candidate FSS Algorithms

Evaluation Strategy	CBF	CFS	Rank Search	Info Gain	Relief-F	SFS
Consistency	✓		✓			✓
Dependency		✓	✓			✓
Information				✓		
Distance					✓	

4.2 Evaluation Metrics

In order to evaluate the propose method and compare it with other baseline methods, we have adopted standard metrics from literature that are frequently used in earlier meta-learning studies for evaluation of algorithm selection systems e.g., in [5], [18], [39]. According to literature, accuracy, precision, recall, F-measure and Hit-Ratio can be defined as follows. Accuracy (A) : Accuracy is defined for each instance as the proportion of predicted accurate labels to total number of labels (predicted and actual) for that instance. The overall accuracy is the mean of all instances.

$$\text{Accuracy, } A = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{8}$$

Precision ($\boldsymbol{P}$) : Averaging over all cases, precision is the ratio of correctly predicted labels to all actual labels.

$$\text{Precision, } P = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Z_i|} \tag{9}$$

Recall (R): Recall is defined as the ratio of correctly predicted labels to the actual correct labels.

$$\text{Recall, } R = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i|} \tag{10}$$

$\boldsymbol{F}_1$-Measure (F): The following definition for F 1-measure follows logically from the definition for accuracy and recall (harmonic mean of precision and recall).

Hit rate represents the probability the for any instance, the set of predicted appropriate algorithms contains at least one correctly predicted algorithm i.e. if $Z_i \cap Y_i \neq \emptyset$ then its value is 1 otherwise it is 0. Like the rest of the metrics, its result is also the mean of all the instances.

$$HR(d_i) \& = \begin{cases} 1, & \text{if } Z_i \cap Y_i \neq \emptyset \\ 0, & \text{Else} \end{cases} \tag{11}$$

4.3   Results and Discussion

In this part, we provide the outcomes of the experiments conducted, which included an evaluation of the proposed approach with other baseline methods, sensitivity analysis of the parameters, and statistical analysis of the variations of these methods. Given the scope of previous AutoML works and the presence of numerous competing solutions, we compared the performance of our proposed method for FSS algorithm selection to three cutting-edge baseline AutoML methods. The baseline methods are the MCFA Framework [13], IBL Ranking [11], and Fuzzy-Sim [16].

First, we present results of the Accuracy, Hit-Ratio and F-measure. In computing these metrics, the Leave-one-out cross validation approach is employed,, i.e., the learning process is applied once per instance, with all other instances serving as a training set and the selected instance serving as a single-item test set. **Fig. 3** shows the accuracy of the proposed and baseline methods.
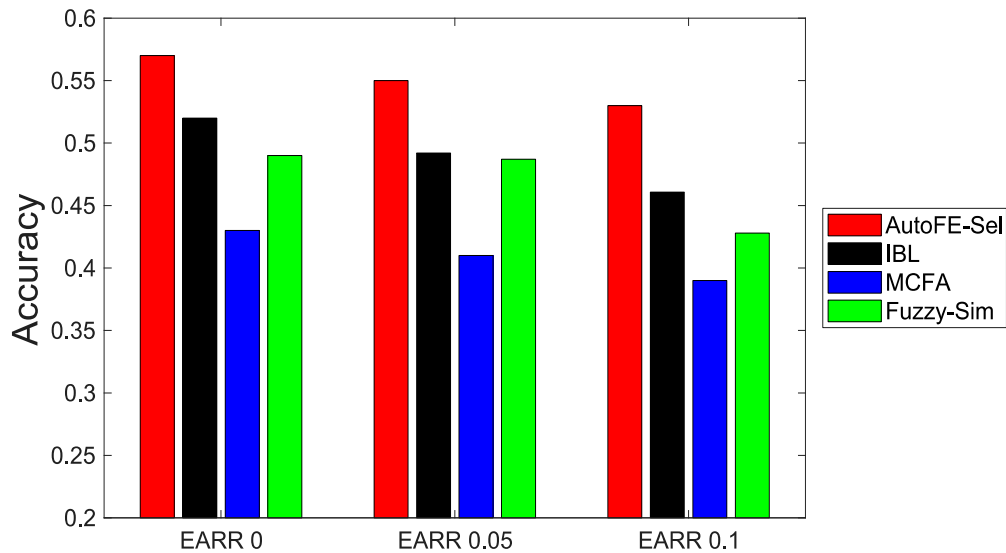

Fig. 3. Accuracy

The horizontal axis shows the variations in each of the methods under different values of α and β in the EARR metric. The three different outcomes on the horizontal axis correspond to the results obtained with the variations in the EARR metric, i.e., EARR = 0, EARR = 0.05, and EARR = 0.1.
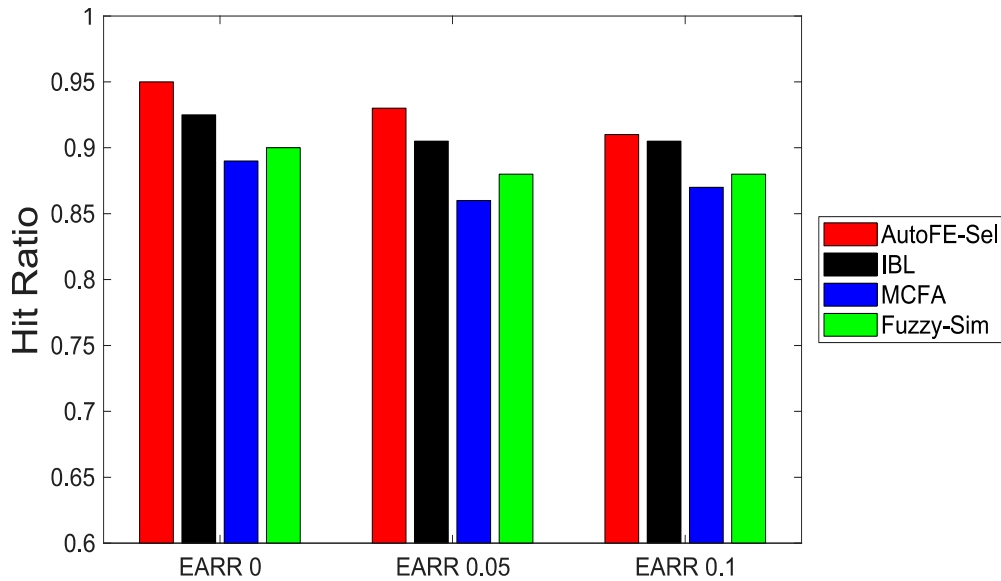
Fig. 4. Hit Ratio

Under various configurations of the EARR measure, the AutoFE-Sel outperforms the baseline approaches in terms of accuracy. Typically, when the EARR value increases, the accuracy decreases marginally. This is due to the fact that a high EARR value favors FSS algorithms with low computing cost that choose fewer features. Subsequently, fewer candidate algorithms become suitable for a given problem, which affects the overall accuracy. Among the three baseline methods, the accuracy of IBL-Ranking is higher than that of MCFA and Fuzzy-Sim. Even though, the definition of accuracy metric in equation 3 is a much stricker condition as it penalize the metrics by each false positive or false negative selection, still the proposed method is able to achieve high accuracy compared to other baseline methods.

The results of the Hit-Ratio metric are shown in **Fig. 4**. Hit-Ratio corresponds to the average probability that, for any dataset d_i, the set of selected algorithms Z_i contains at least one algorithm from meta-target Y_i. Heigher Hit-Ratio indicate better performance of the system. For the three variations of the EARR metric, the AutoFE-Sel method performs better than the baseline methods. The lowest Hit-Ratio metric value of AutoFE-Sel is greater than 92%, while the heiger value is around 95%. Among the baseline methods the performance of MCFA is the lowest, while IBL-Ranking has comparable efficacy on the Hit-Ratio to the AutoFE-Sel. The Hit-Ratio metric in algorithm selection literature is usually considered a loose metric and is used to indicate the feasibility and practicability of a method in real a environment. Generally, the majority of the baseline methods in various domains for algorithm selection performs better on Hit-Ratio compared to other metrics. The same pattern regarding the concerned metric is also observed in the current work of the evaluation in the proposed and baseline methods.
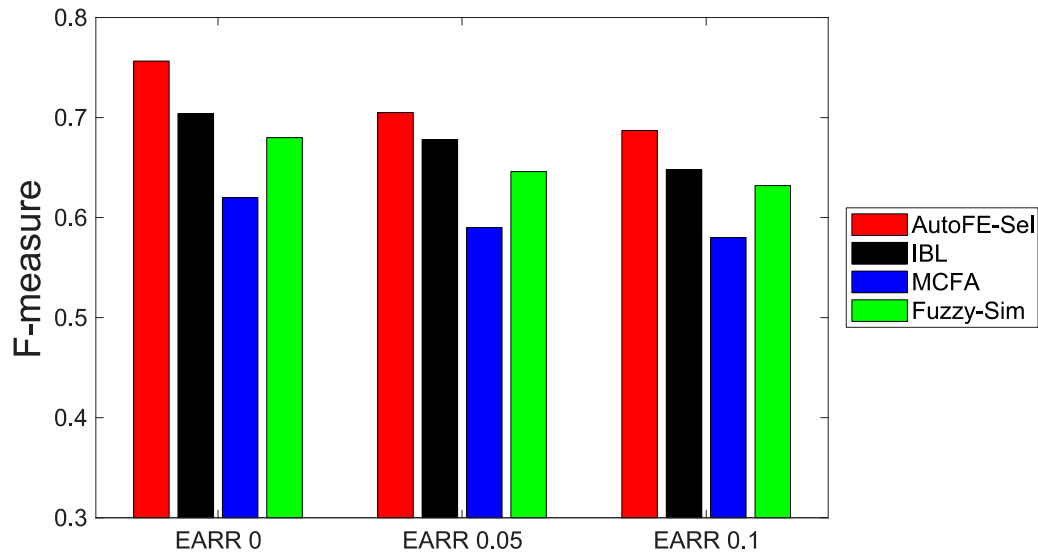
Fig. 5. F-measure

Regarding recall and accuracy, the goal is to enhance Recall without compromising Precision. However, Recall and Precision frequently contradict one another. This is due to the fact that a rise in the number $|Yi \cap Zi|$ (increasing Recall) results in an increase in the number $|Zi|$, hence decreasing precision. For the purpose of analyzing the performance of these two measures simultaneously, F-measure is utilized, which offers a balance between Precision and Recall. As shown in equation 11, F-measure is the harmonic mean of precision and recall; high true positive and high true negative values are required for good performance. The higher value of F-measure corresponds to better recommendaitons of algorithms for a given problem. Results on the F-measure are shown in **Fig. 5**, which shows that the AutoFE-Sel has a higher F-measure than the baseline methods for various values of EARR. It is important to notice that the difference in performance between AutoFE-Sel and the baseline approaches in terms of accuracy and F-measure is significantly larger than the performance gap in terms of Hit-Ratio.

To confirm the difference and advantage of AutoFe-Sel, we conduct statistical analysis utilising the Wilcoxon signed-rank test, which compares two or more methods on several problems. **Table 5** displays the findings of the statistical analysis at a significance level of 0.05. For the comparison of the AutoFE-Sel method with each of the basleine methods, the null hypothesis is that the AutoFE-Sel method is statistically equivalent to or worse than the baseline method while the alternative hypothesis is that the AutoFE-Sel method is statistically superior to the baseline methods. The alternative hypothesis is accepted if the p-value of the test result is less than 0.05. The null hypothesis is accepted otherwise. With regard to IBL-Ranking, it can be observed from **Table 5** that the p-values for the alternative hypothesis on accuracy and F-measures are lower than 0.05, hence the alternative hypothesis is accepted. It means that the difference is statistically significant and AutoFE-Sel performs better. However, the P values on Hit-Ratio metric are comparitevely heigher than the other metrics, as described earlier that Hit-Ratio is a loose matrix and generally all the method shows good performance on it. With regard to MCFA and Fuzzy-Sim based methods, results of the P values in the table shows that AutoFE-Sel performs better. Overall the results from all the metrics and statistical analysis show that AutoFE-Sel does improve the performance of algorithm selection on the given metrics.

Table 5. Statistical significance test

Alternative Hypothesis	Metrics	EAR 0	EARR 0.05	EARR 0.1
AutoFE-Sel > IBL-Ranking	Accuracy	0.03	0.02	0.02
	F-measure	0	0.01	0.02
	Hit Ratio	0.06	0.27	0.38
AutoFE-Sel > MCFA-Framework	Accuracy	0	0	0
	F-measure	0	0	0
	Hit Ratio	0.01	0	0.01
AutoFE-Sel > Fuzzy-Sim	Accuracy	0.02	0.01	0.01
	F-measure	0	0.01	0.01
	Hit Ratio	0.04	0.03	0.03

5.   Conclusion

In this paper, we present the AutoFE-Sel architecture for the selection of FSS algorithms. The AutoFE-Sel is based on stacking an framework [18]. It has two key advantages. (1) It improves the meta-modeling by combining multiple ML-KNN weak learners; (2) It takes advantage of the diversity of the existing complementary groups of DCM by generating their various alternative combinations, subsequently leveraging the performance of the system. The AutoFE-Sel framework is comprised of three phases: (i) collection of meta-data, including meta-target estimation and extraction of DCMs (ii) meta-model construction (iii) using the learned meta-model for selection of adequate subset of candidate algorithms for any given problem. In order to evaluate the performance of the proposed method, it is compared with three other baseline methods in a large experimental setup consisting of 125 datasets and 12 FSS induced by five classifiers. The experimental findings demonstrate that AutoFE-Sel outperforms the three baseline methods. Our research adds to the body of knowledge in the field of algorithm selection by empirically demonstrating that imitating the selection of FSS methods in a meta-learning setup based on ensemble learning, consequently enhancing the performance of algorithm selection systems. Moreover, the framework is easily extendable with regard to its components. This study lays the groundwork for developing more robust implementations of ensemble based methods for algorithm selection. In future we are planning to investigate the recommendation of other preprocessing methods like noise filter selection in the proposed ensemble based meta-learning setup.

References

[1] R. A. Da Silva, A. M. D. P. Canuto, C. A. D. S. Barreto, and J. C. Xavier-Junior, "Automatic Recommendation Method for Classifier Ensemble Structure Using Meta-Learning," *IEEE Access*, vol. 9, pp. 106254–106268, 2021. Article (CrossRef Link)
[2] B. Bischl et al., "ASlib: A benchmark library for algorithm selection," *Artificial Intelligence*, vol. 237, pp. 41–58, 2016. Article (CrossRef Link)
[3] M.-A. Zöller and M. F. Huber, "Benchmark and Survey of Automated Machine Learning Frameworks," 2019. Article (CrossRef Link)
[4] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: a survey of trends and technologies," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 117–130, Jun. 2015. Article (CrossRef Link)

[5]   I. Khan, X. Zhang, M. Rehman, and R. Ali, "A Literature Survey and Empirical Study of Meta-Learning for Classifier Selection," *IEEE Access*, vol. 8, pp. 10262–10281, 2020. Article (CrossRef Link)

[6]   B. A. Pimentel and A. C. P. L. F. de Carvalho, "A new data characterization for selecting clustering algorithms using meta-learning," *Information Sciences*, vol. 477, pp. 203–219, 2019. Article (CrossRef Link)

[7]   R. B. C. Prudencio, T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, no. 1–4, pp. 121–137, 2004. Article (CrossRef Link)

[8]   L. P. F. Garcia, A. C. P. L. F. de Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level," *Neurocomputing*, vol. 176, pp. 14–25, 2016. Article (CrossRef Link)

[9]   M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Auto-sklearn: Efficient and Robust Automated Machine Learning," *Automated Machine Learning*, pp. 113-134, 2019. Article (CrossRef Link)

[10] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, 2013. Article (CrossRef Link)

[11] G. Wang, Q. Song, H. Sun, X. Zhang, B. Xu, and Y. Zhou, "A Feature Subset Selection Algorithm Automatic Recommendation Method," *Journal of Artificial Intelligence Research*, vol. 47, no. 1, pp. 1–34, 2013. Article (CrossRef Link)

[12] F. Andrey and A. Pendryak, "Datasets meta-feature description for recommending feature selection algorithm," in *Proc. of IEEE the AINL-ISMW FRUCT Conference*, pp. 11–18, 2015. Article (CrossRef Link)

[13] 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 Systems with Applications*, vol. 75, pp. 1–24, 2017. Article (CrossRef Link)

[14] S. Shilbayeh and S. Vadera, "Feature selection in meta learning framework," in *Proc. of Science and Information Conference*, pp. 269–275, 2014. Article (CrossRef Link)

[15] F. H. Jose A. Saez, Julian Luengo, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognition*, vol. 46, pp. 355–364, 2013. Article (CrossRef Link)

[16] Z. Shen, X. Chen, and J. M. Garibaldi, "A novel meta learning framework for feature selection using data synthesis and fuzzy similarity," in *Proc. of IEEE International Conference on Fuzzy Systems*, pp. 19–24, 2020. Article (CrossRef Link)

[17] G. Wang, Q. Song, and X. Zhu, "Ensemble Learning Based Classification Algorithm Recommendation," 2021. Article (CrossRef Link)

[18] X. Zhu, C. Ying, J. Wang, J. Li, X. Lai, and G. Wang, "Ensemble of ML-KNN for classification algorithm recommendation," *Knowledge-Based Systems*, vol. 221, p. 106933, 2021. Article (CrossRef Link)

[19] H. L. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, "Feature Selection: A Data Perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2017. Article (CrossRef Link)

[20] J. P. Monteiro, D. Ramos, D. Carneiro, F. Duarte, J. M. Fernandes, and P. Novais, "Meta-learning and the new challenges of machine learning," *International Journal of Intelligent Systems*, vol. 36, no. 11, pp. 1–33, 2021. Article (CrossRef Link)

[21] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski, "Using meta-learning for automated algorithms selection and configuration: an experimental framework for industrial big data," *Journal of Big Data*, vol. 9, no. 1, 2022. Article (CrossRef Link)

[22] Q. Song, G. Wang, and C. Wang, "Automatic recommendation of classification algorithms based on data set characteristics," *Pattern Recognition*, vol. 45, no. 7, pp. 2672–2689, 2012. Article (CrossRef Link)

[23] G. Wang, Q. Song, and X. Zhu, "An improved data characterization method and its application in classification algorithm recommendation," *Applied Intelligence*, vol. 43, no. 4, pp. 892–912, 2015. Article (CrossRef Link)

[24] 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," *Information Sciences*, vol. 301, pp. 181–194, 2015. Article (CrossRef Link)

[25] 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," *Machine Learning*, vol. 107, no. 1, pp. 209–246, 2018. Article (CrossRef Link)

[26] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," *Information Sciences*, vol. 484, pp. 302–337, 2019. Article (CrossRef Link)

[27] 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 Computing Surveys*, vol. 52, no. 5, pp. 1–34, 2019. Article (CrossRef Link)

[28] A. Rivolli, L. P. F. Garcia, C. Soares, J. Vanschoren, and A. C. P. L. F. De Carvalho, "Meta-features for meta-learning," *Knowledge-Based Systems*, vol. 240, p. 108101, 2022. Article (CrossRef Link)

[29] X. Zhang, R. Li, B. Zhang, Y. Yang, J. Guo, and X. Ji, "An instance-based learning recommendation algorithm of imbalance handling methods," *Applied Mathematics and Computation*, vol. 351, pp. 204–218, 2019. Article (CrossRef Link)

[30] N. M. and Vitor Cerqueiraa, "Automated imbalanced classification via meta-learning," *Expert Systems with Applications*, vol. 178, p. 115011, 2021. Article (CrossRef Link)

[31] D. Oreski, S. Oreski, and B. Klicek, "Effects of dataset characteristics on the performance of feature selection techniques," *Applied Soft Computing Journal*, vol. 52, pp. 109–119, 2017. Article (CrossRef Link)

[32] L. C. Okimoto and A. C. Lorena, "Data complexity measures in feature selection," in *Proc. of IEEE International Joint Conference on Neural Networks*, pp. 1–8, 2019. Article (CrossRef Link)

[33] G. Wang, Q. Song, X. Zhang, and K. Zhang, "A Generic Multilabel Learning-Based Classification Algorithm Recommendation Method," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 1, pp. 1–30, 2014. Article (CrossRef Link)

[34] A. Rafael, S. Parmezan, H. Diana, N. Spolaôr, and F. Chung, "Automatic recommendation of feature selection algorithms based on dataset characteristics," *Expert Systems With Applications*, vol. 185, p. 115589, 2021. Article (CrossRef Link)

[35] J. Demˇsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. Article (CrossRef Link)

[36] 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," 2018. Article (CrossRef Link)

[37] E. Alcobaça, F. Siqueira, A. Rivolli, L. P. F. Garcia, J. T. Oliva, and A. C. P. L. F. de Carvalho, "MFE: Towards reproducible meta-feature extraction," *Journal of Machine Learning Research*, vol. 21, pp. 1–5, 2020. Article (CrossRef Link)

[38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009. Article (CrossRef Link)

[39] X. Zhu, X. Yang, C. Ying, and G. Wang, "A new classification algorithm recommendation method based on link prediction," *Knowledge-Based Systems*, vol. 159, pp. 171–185, 2018. Article (CrossRef Link)

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 Lecturer at 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.

RAMESH KUMAR AYYASAMY (Senior Member, IEEE) received the Ph.D. degree in information technology from Monash University, Australia, in 2013. Starting from 2003 to 2008 and from 2013 to 2014, he worked as a Lecturer in Tamil Nadu, India, and Monash University, Malaysia, consecutively. Since 2015, he has been working as an Assistant Professor with the Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Malaysia. His research interests include artificial intelligence, big data analytics, cyberbullying, deep learning, machine learning, and text mining. Dr. Ramesh serving as a reviewer for different journals and conferences and a member of editorial boards

RAHMAN ALI received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2016. He was with the Labo-ratoired'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 60 publications and conference contributions. His main scientific research interests include machine learning, data mining, recommendation systems, reasoning and inference, and natural language processing