Neurocomputing 518 (2023) 344–359

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Feature selection for distance-based regression: An umbrella review and a one-shot wrapper

Joakim Linja [a,*], Joonas Hämäläinen [a], Paavo Nieminen [a], Tommi Kärkkäinen [a]

[a] Faculty of Information Technology, University of Jyväskylä, Finland

ARTICLE INFO

Article history:
Received 8 March 2022
Revised 23 September 2022
Accepted 4 November 2022
Available online 10 November 2022
Communicated by Zidong Wang

Keywords:
Distance-based method
Feature selection
Feature saliency
Wrapper algorithm
EMLM

ABSTRACT

Feature selection (FS) may improve the performance, cost-efficiency, and understandability of supervised machine learning models. In this paper, FS for the recently introduced distance-based supervised machine learning model is considered for regression problems. The study is contextualized by first providing an umbrella review (review of reviews) of recent development in the research field. We then propose a saliency-based one-shot wrapper algorithm for FS, which is called *MAS-FS*. The algorithm is compared with a set of other popular FS algorithms, using a versatile set of simulated and benchmark datasets. Finally, experimental results underline the usefulness of FS for regression, confirming the utility of certain filter algorithms and particularly the proposed wrapper algorithm.

1. Introduction

Dissimilarity has a central role in unsupervised learning, but the increased popularity of using distance-based models for supervised problems (e.g., [1–4]) illustrates how the gap between unsupervised and supervised learning tasks is diminishing. In fact the dissimilarities among a set of prototypical observations can be used as features with any predictive model [5,6]. Further, the history of distance-based supervised models can be traced back to the radial basis function networks [7,8] with a linear kernel [9,10]. The latter references, as noted in [Remark 1] [4], provide proof for the universal approximation capability of the linear distance-regression model. The scope of this article is to consider dimension reduction, specifically feature selection (FS) for this distance–based learning machine [4]. Compared to its sibling method, feature extraction (FE), FS keeps the features as they are while removing those considered unnecessary. FE, on the other hand, aims to reduce the number of features for example, through projections [11]. In this way, FE mixes information of the original features.

The need for FS stems from increasingly complex and demanding datasets where the number of features may become detrimental to the practical operation of a machine learning model [12–14]. FS refers to the identification and selection of a subset of relevant features for a data-based model. Therefore, it is basically a search problem, which can generally be addressed using many techniques (e.g., forward or backward search, exhaustive search, branch-and-bound, evolutionary approaches) and multiple feature assessment criteria (information, distance, dependency, consistency, and accuracy measures) [15]. The FS process has three main goals [16]: improve the model performance, provide faster and more cost-effective models, and improve the understanding of the data generation process. However, the last goal cannot be fully addressed when working with an already featurized, secondary dataset. The classical division of the main types of features is given in [17,18]: *irrelevant, weakly relevant,* and *strongly relevant*. The weakly relevant features were further categorized in [19] as redundant or non-redundant.

The main branches of FS techniques are filter and wrapper approaches [17], depending on whether the intended machine learning model itself is used in the FS process. Usually, this means that a filter approach is faster, and a wrapper approach is more accurate [18]. Filters usually contain two main steps [20]: 1) ranking of features according to importance scoring; and 2) selection of most important features based on step 1). Hybrid [21–24] or embedded methods [25,26] perform FS by using another model or an untrained model to assess feature relevancy. Embedded methods that rank and select features during the construction of the predictive model include decision trees [27] and ensemble learning methods, most prominently, random forests [28,29].

For wrappers, the search phase of the used features means multiple repetitions of training, i.e., the estimation of the model's parameters. Therefore, one aspect of categorizing different wrap-

* Corresponding author.

https://doi.org/10.1016/j.neucom.2022.11.023
0925-2312/

pers is given by the number of model trainings needed during the search of the final feature subset. This number can be very large, for instance, when optimization-based methods and metaheuristics are used to search the features (see Section 2.4). Here, we propose a *one-shot wrapper*: rank the features using scores computed from a trained distance-based model with the full feature set and, through a threshold, select the most important ones for the reduced model. The entire endeavor to determine the final model then requires two training rounds — initially with the original feature set and finally with the selected feature set. In between, the full feature set model is used to compute feature importances for the thresholding. It should be noted that if the feature ranking based on the full model is computationally not more expensive than training the model, then this one-shot approach does not add any computational complexity to the (unavoidable) model training.

Feature scoring and ranking is actually a specific technique to quantify and improve the understandability of a model, to explain its behavior [30]. Indeed, interpretable machine learning refers to the ability to understand the work logic of machine learning models and algorithms [31]. A branch of techniques for this purpose uses the saliency of features to rank their explanatory power as a post hoc explainability approach [32,33]. For neural networks (NN), one measure of saliency is the input sensitivity, i.e., the partial derivative of the network's output with respect to its input. For shallow networks, feature assessment originating from this idea was proposed in [34] and, since then, many similar FS techniques have been considered [35]. Use of a partial derivative method was rediscovered within the context of deep neural networks in [36], where the first-order Taylor's expansion was used to generate an image-specific saliency map for visual interpretation of a convolutional neural network classifier.

In this work, we provide a wide-ranging overview of earlier FS research, noting that explorations of FS for distance-based regression methods have been scarce. Thus, we aim to fill this gap by proposing and evaluating a new FS algorithm for this learning task. Our main contributions are as follows:

- an umbrella review of recent reviews on FS
- derivation of a one-shot FS approach for distance-based regression
- extensive experimental comparison of filter and wrapper FS algorithms, ensuring the viability of the proposed algorithm and its building blocks.

The rest of the paper is organized as follows. Section 2 presents the umbrella review. Section 3 introduces the proposed FS algorithm and the necessary mathematical foundations. Section 4 describes the used synthetic datasets as well as presents the used open access datasets. In addition, it presents the evaluation criteria used in measuring the gained results. Section 5 details the experiments and results while also presenting the related discussion. The data tables with the results discussed in Section 5 are in the appendix. Finally, Section 6 discusses and concludes the work.

2. Umbrella review on FS techniques

FS is a particular instance of model selection for which a considerable number of techniques have been depicted and experimented with over the years [37]. Therefore, the full coverage of the development and current status of the FS field of research from primary studies, after the influential classical works like [38,39,15,16], is out of the scope of this article. However, in order to position our work in the research field, the recent developments of FS research will be summarized through an umbrella review, the purpose of which is to locate and consider different views and perspectives on a broad area of interest [40]. Here, instead of addressing the primary studies in the field, the already undertaken surveys and reviews and their summarization are considered [41]. Further, the umbrella review is a common practice in the medical research domain where summarizing the vast amount of knowledge from primary articles is a tedious task. However, while this type of review is rarely used in the machine learning field, there was one such recent study related to deep learning [42]. Similar to the deep learning field, there exist many recent reviews for FS, which indicates the need for conducting an umbrella review for FS as well.

For this purpose, we used Google Scholar on December 9–10, 2021, with the search term "feature selection review" and checked the first 500 returned links. Of these links, 32 high-quality reviews and summaries on FS for supervised learning, published since 2013 in journals by leading publishers (IEEE, ACM, Elsevier, Springer, Wiley), were identified to be summarized next. The final paper of the search that was included in the summary, [43], had an entry number of 475. This was the only hit for the last 50 checked papers. The annual number of reviews and summaries identified was as follows: one paper from 2013, three from 2014, two from 2015, three from 2016, three from 2017, seven from 2018, four from 2019, seven from 2020, and two from 2021. The numbers indicate an increasing trend in summarizing the overall research achievements of the research field.

The narrative review of the papers is primarily organized in a chronological order. From the first paper [44] onward, we have not repeated the shared contents but tried to provide only new, relevant information from the chronologically subsequent papers. However, reviews addressing a common topic are presented together, and this defines the subtitle structure used below. The subtitles are ordered according to the publication year of the first review article of the topic, although a review of general FS reviews is presented first. Due to the abundance of abbreviations in the umbrella review, we have included Table 1, in which they are listed.

2.1. General reviews

A general review on FS in classification was provided in [43]. Correlation criterion and MI-based criteria were depicted for filter methods. Wrapper methods were classified into Sequential Selection Algorithms (backward search) and Heuristic Search Algorithms (use of a GA) to identify a subset of features. For embedded methods, MI and weights of a classifier were depicted as feature assessment criteria. As classifiers, SVM and RBFN were introduced. Experiments with six datasets (Breast cancer, Diabetes,

Table 1
List of abbreviations used in the umbrella review.

ANN	Artificial Neural Network	Mb	Markov blanked
BNS	Bi-Normal Separation	MB	Markov boundary
CBM	Correlation-Based Methods	MCO	MultiCriteria Optimization
DBM	Deep Boltzmann Machine	MD	Maximum Discrimination
DF	Document Frequency	MH	MetaHeuristic
DT	Decision Tree	MI	Mutual Information
ELM	Extreme Learning Machine	MLM	Minimal Learning Machine
EMLM	Extreme MLM	MLP	MultiLayered Perceptron
FS	Feature Selection	NB	Naive Bayes
GA	Genetic Algorithm	RBFN	Radial Basis Function Network
IG	Information Gain	RF	Random Forest
kNN	k-Nearest Neighbors	SVM	Support Vector Machine
LR	Linear Regression techniques	SVR	Support Vector Regression
LRR	Logistic Ridge Regression	TF-IDF	Term Frequency-Inverse DF

Ionosphere, Liver disorder, Medical, Fault mode) demonstrated the usefulness of FS but did not provide any methodological rankings.

The second general review encountered was [45]. However, after introducing eight different algorithms for FS including maximum variance, Laplacian score, spectral regression, sparsity favoring, etc. approaches, the experiments were only performed for unsupervised FS. Therefore, we did not consider this work further.

In [46], a large and very comprehensive FS review organized from a data perspective, mainly for ranking the features or identifying feature subsets through sparsity-favoring linear learning, was given. The different cases for FS were defined as follows: traditional FS and FS with structured features for conventional data; FS with linked data, multi-source FS, and multi-view FS for heterogenous data; and FS with streaming data or streaming features for streaming data. For conventional data, five similarity-based score criteria (Laplacian Score, SPEC, Fisher Score, Trace Ratio, and ReliefF) were introduced. Then, nine information theory -based methods (IG, MI, Minimum Redundancy Maximum Relevance, Conditional Infomax, Joint MI, Conditional MI, Interaction Capping, Double Input Symmetrical Relevance, and a Correlation-Based Filter) were depicted. Third category of methods for conventional data favored sparsity. Their construction was based on a suitable loss function with nonconvex regularization using the $\|\cdot\|_p$-norm for $0 \leqslant p \leqslant 1$ or $\|\cdot\|_{p,q}$-norm for $p > 1$ and $0 \leqslant q \leqslant 1$ with both vector- or matrix-valued unknowns. The Actual formulations for different methods are given in [Section 2.3][46]. In statistical methods, the features with low variance or t-score or Chi-square score or with a high Gini index are eliminated. In Correlation-based FS (CFS), one searches a feature subset which has a strong correlation with class labels but weak inter correlations. FS with structured features included Group Lasso, and its sparse and overlapping variants. For tree structured features, a Tree-guided Group Lasso was introduced, and for graph structure, a Graph-Lasso with two additional variants (GFLasso and GOSCAR) was defined. For linked data, FS using graph regularized least-squares, user-post relationship regularizer, and unsupervised techniques encoding latent and low-rank representations were depicted. Multi-source FS could be addressed using Geometry-Dependent Covariance Analysis and multi-view scenarios, which refer to FS from different feature spaces simultaneously by using linear least-squares with specially constructed desired outputs, constraints, and sparsity favoring regularizers [Section 4.3] [46]. For streaming data with feature streams, assuming a constant number of instances for e.g., Grafting Algorithm (Lasso-like method) and Alpha-Investing Algorithm (a statistical threshold technique) were described. For actual data streams, a particular online FS algorithm and an unsupervised least-squares approach were introduced. Entire sections in the review were dedicated to the evaluation of different FS approaches and open problems in the field. This paper was clearly one of the most comprehensive reviews that was found, even though it should be noted that the linkage between a particular FS technique and the form of data is not a function but a relation: many FS techniques are suitable or modifiable for use with many forms of data.

A general FS review was provided in [47], where the techniques were not directly introduced but discussed through a Problem–Solution-Discussion contents presentation model. Many of its themes coincided with those in [46] as summarized in the previous paragraph, so we only depict the additional topics covered: distributed algorithms for FS (utilizing, e.g., MPI, MapReduce, and peer-to-peer networks), multi-label FS (of which a dedicated review was provided in [48] and summarized in Section 2.6), privacy-preserving FS (where the overall privacy degree of the chosen features is controlled), and adversarial techniques for FS (based on currently popular adversarial network architectures and attacks on the classification model).

In [49], FS in machine learning was addressed on a general level. However, this high quality review was superseded by the even more extensive exposure in [46] summarized above. Nevertheless, in relation to wrappers, clustering-based unsupervised techniques, GAs, and particle swarm optimization methods were presented, for which a thematic overview is given below in Section 2.4. Semi-supervised FS, which was more thoroughly summarized in [50] as depicted in Section 2.5, was also examined. Future challenges were linked to the properties of data (small, large, imbalanced) and ensemble or online FS techniques.

The FS for ensemble-based machine learning models was reviewed in [51]. The basic relation between ensembles and FS is composed on the triplet of {base learners}× {feature subsets}× {observation subsets}. Clearly, we can link filters, wrappers, or embedded FS methods to all learners or perform FS individually in the base learners. Interestingly, RF, which is truly a prototypical feature-subset based FS method, was not addressed explicitly in this review. However, a comprehensive depiction of the existing FS tools and techniques that are available on the commonly used software platforms was presented.

In [52], causality-based FS was reviewed, and a new open-source library was proposed and tested. Instead of co-occurrences and correlations within a set of features and targets (usually labels), causality refers to the identification of cause-and-effect relationships typically using graph-based graphical models such as the Bayes/Bayesian Networks with Markov blanket (Mb) or Markov boundary (MB). For MB, which is the minimal set of Mb referring to a subset of variables containing all necessary information to infer a random variable, this review first classified (into five categories of learning) and described 30 different constraint-based FS algorithms. Similarly, three categories with eight algorithms for score-based (scoring or the actual cost function and how it is searched) identification of MB were described. Furthermore, four categories of 18 algorithms to separate features of direct causes (parents) from those with only direct effects (children) were depicted. The new toolbox, CausalFS, was then presented with a list of future challenges regarding form and quality of data (see Section 2.3), causal effect estimation, and causal FS for NNs.

The most recent review in our umbrella review was [53], which provided a clear introduction to search methods and mechanisms in FS. The measures which originated from four categories (statistics, probability, similarity, and sparsity) were a proper subset of those provided in [46]: evolutionary FS algorithms are described in the individual reviews in Section 2.4, and the additional SVM-RFE method mentioned is already part of the first review in here [44]. However, compared to Section 2.5, the body of domains where FS is needed was enlarged to cover natural language processing, emotion recognition, speech processing, sentiment analysis, and biometrics. These and other domains were carefully linked to primary publications, datasets, evaluation measures, and future challenges.

2.2. Filter methods

One of the most popular class of methods for FS filters are the information-theoretic methods that utilize MI. These techniques were reviewed in [54]. The basics of information theory and key concepts of filters (relevance, redundancy, and complementarity) were presented. The main results of the work were a unifying framework and a list of open problems without any empirical experiments.

The similarity of the FS techniques, especially the rankings they provided were compared using the Kuncheva index averaged over

all pairwise comparisons in [55]. Five univariate (χ^2, IG, Symmetrical Uncertainty, Gain Ratio, and OneR) and three multivariate (ReliefF, SVM-ONE, and SVM-RFE, the latter two both with linear kernel) FS methods were considered for 16 classification datasets with the number of features ranging from 1559 to 10 458. Similar behavior was observed for the three first univariate methods, whereas different behavior of Gain Ratio was witnessed. Difference of multivariate methods compared to the univariate ones was also observed, together with the sensitivity of SVM-related methods on their internal parameters concerning the number/portion of features discarded at each iteration. In this respect, ReliefF had the most stable and consistent behavior.

In [56], structured, sparsity-inducing methods are presented by separating vector-based and matrix-based FS. However, even though the work provides a comprehensive survey, it is superseded by the even more extensive exposure [46] already summarized above.

One of the most commonly used filters, as also demonstrated by this umbrella review, are Relief-based algorithms (RBAs), which were comprehensively depicted and reviewed in [57]. Altogether, 21 variants of RBAs from four general branches were presented, and among them, from our perspective, the most important being the regression-oriented ReliefF with $\mathcal{O}\left(N^2 n\right)$ computational costs for the number of observations N and the number of features n. Because feature scoring in Relief is based on the feature value differences between a target and its neighboring observations, this filter is particularly relevant to our experiments presented in Sections 4 and 5.

A large comparison of filters for classification, utilizing 16 high-dimensional datasets and a specific R-package *mlr*, was presented in [58]. The 22 filters, also visible in other papers of this umbrella review, originated from statistical tests, feature variance, univariate predictive performance, feature importance with RF, and MI. The classifiers were kNN, LRR, and SVM. In the analysis, similarity of feature scores for ranking was first assessed, by identifying three groups of similar filter methods. The actual comparison for (data, model,filter)-triplets was generally concluded as follows "no filter method is better than all the other methods on all data sets," and "there is no subset of filter methods that outperforms all other filter methods." Because of these conclusions, it was recommended to test all filters in a particular context if computational resources suffice.

2.3. FS for particular forms of data

The use of synthetic data allows for rigorous comparison between the selected features and accuracies of the reduced feature models. In [44], FS for synthetic data classification was reviewed. Altogether, seven filters (correlation-based method, consistency-based filter, the Interact algorithm, IG, ReliefF-algorithm, the mRMR method, and the $\mathscr{M}_d$ filter), two embedded methods (SVM-RFE for SVM and FS-P for Perceptron), and two wrappers (Wrapper-C4.5 and Wrapper-SVM using the WrapperSubsetEval algorithm) were applied over eleven synthetic datasets (CorrAL, CorrAL-100, XOR-100, Parity3 + 3, LED-25, LED-100, Monk3, SDI1–3, and Madelon), which included irrelevant and redundant features, noise, and various interaction patterns. Moreover, four classifiers were used (NB and IB1 in addition to C4.5 and SVM). In this work, the challenges were pointed out in the threshold selection for methods that produce feature importance values and, therefore, allow ranking of individual features. They concluded that ReliefF and SVM-RFE with nonlinear kernel were the best methods, and recommended the former because of its independence on the classification model and computational efficiency. Additionally, the difficulties in comparing and ranking wrapper methods were also noted.

Online settings provide a special context on the availability of data for feature construction. In [59], FS for streaming data classification was considered, with the full or only a partial subset of features being accessible for each arriving new instance (decided by the learner). Next, three novel algorithms, a truncated perceptron, a sparse projection approach, and learning with partial inputs, were presented. The experimental comparison was performed using nine smaller datasets (magic04, svmguide3, german, splice, spambase, a8a, RCV1, and two topic pairs from 20Newsgroup) and five larger datasets (KDDCUP08, ijcnn1, codrna, covtype, and KDDCUP99), focusing on the average number of mistakes made by the algorithms. Further, real word applications of image classification in computer vision and microarray gene expression analysis in bioinformatics were also demonstrated. The overall conclusion was that the proposed online algorithms turned out to be scalable and more efficient than some state-of-the-art batch FS techniques. However, upon taking a closer look, this paper turned out to be a primary study.

A particular focus on online FS with streaming features — a subtopic also covered in the extensive review [46] summarized in Section 2.1 — was undertaken in [60]. The additional FS techniques compared to [46] contained MI-based SAOLA and Group-SAOLA (Scalable and Accurate OnLine Approach), uncertainty-minimizing GFSSF (Group FS with Streaming Features), and Lasso-oriented OGFS (Online Group FS). Experiments with several (over 10) benchmark data sets did not provide methodological conclusions or rankings but, instead, generated a list of challenges related to multi-label cases (reviewed separately, e.g., in [48] as summarized in Section 2.6), quality of data in real-world applications, and the need to distribute computational efforts.

2.4. Optimization-based FS techniques and metaheuristics

A general approach for FS is to cast the problem of identifying a subset of features as an optimization problem. Such an approach needs the definition of a cost function that measures the goodness of a feature subset, typically through the accuracy of a classifier. However, the strict convexity and differentiability of such cost functions might be difficult to establish, so derivative free optimization methods provide a natural family of optimizers in these settings. Clearly, both the necessity of the metalevel fitness and the search that finds arguments of its extremums causes a significant increase in the computing time.

In [61], nature-inspired metaheuristics including GAs and ant colony optimization were reviewed for FS. Further, a taxonomy of such approaches consisting of stochastic algorithms, physical methods, evolutionary approaches, immune systems, and swarm intelligence were also depicted. Then, the elements and basic constituents of population-based approaches, memetic algorithms incorporating local searchers, clonal selection, harmony search, simulated annealing, tabu search, and swarm algorithms (artificial bee colony, ant colony, firefly algorithm, and particle swarm) were given. Experiments with 12 UCI datasets and C4.5 and NB classifiers concluded the capabilities of all tested algorithms in finding good solutions. Similar to [44] as shown above, in some cases, filter-based evaluators had better results as compared to the more complex FS approaches.

Years 2020–2021 were characterized by multiple FS reviews addressing optimization-based techniques. The latest is [62], which focused on nature-inspired MH techniques from both mapping (how much and what kind of publications) and review (what techniques and results) perspectives. Among others, the large number of nature-inspired MH techniques was summarized: 29 were inspired by insects and reptiles, 15 by birds, 13 by animals,

seven by sea creatures, five by plants, six by humans, and 25 other techniques. Altogether, 21 actual FS-MH algorithms were then listed, which were further divided into chaotic (utilizing various forms of randomness) and binary (strict inclusion/exclusion of features) variants. Moreover, the review [62] superseded a few other recent, more specific FS reviews: the grey wolf optimizer treated in [63,64] which considered the Dragonfly algorithm. Moreover, swarm optimization, which was examined in [65], introduced six different algorithms, which were all contained in five categories and 12 instances of "swarm" presented in [62]. Differently from [62], the chaotic category there was replaced with continuous representation of features. The review in [65] was summarized with the observation that MHs are typically applied to identify both the features and predictive model's parameters, and that FS problems with binary presentation need further studies.

Another perspective on optimization-based FS was detailed in [66], where a systematic review on the use of MCO was presented. In all the introduced cases, the multiple criteria were reduced to two basic objectives: minimal number of features with maximal classification performance. Contents of 38 papers were summarized, where the twofold nature noted above meant that most of the papers depicted wrappers and only five filters. Because of the computational costs of MCO algorithms, the mentional classifiers included rather simple techniques like kNN, NB, DT, and linear SVM, but also SVM, RF, ELM, MLP (i.e., shallow feedforward network), DBM, and Deep NN. Notably, this review also summarized 38 different datasets that were used to evaluate the methods.

2.5. FS in particular application domains

Gene expression data, which was focused on in [50], is characterized by a high ratio of the number of features to that of samples: there can be up to hundreds of thousands of features but only a small sample size. In this paper, a thorough introduction to feature evaluation and selection methods was given along with comprehensive summaries of the prediction accuracy vs. number of selected features for dozens of studies with five popular datasets. Expectedly for such problems, the usefulness of filters and the potential of semi-supervised FS methods integrating unsupervised FS from larger unlabelled data with supervised construction of classifiers was concluded. Similarly, the potential of hybrid FS methods combining multiple filter and/or wrapper approaches was emphasized.

FS in the multimedia context, covering, for example, texts, images, videos, audios, animations, etc. as formats and forms of data, was reviewed in [67]. The basics of FS methods and search strategies were depicted with summaries of their use in supervised, semi-supervised, and unsupervised FS techniques for multimedia data based on 70 original papers in 2001–2017. Interestingly, compared to our paper, years 2013 and 2014 were identified as the most active times of publications especially through the emergence of various heuristics. A special emphasis in the current review was given to interactive, active learning - based approaches. However, with both these techniques as well as in the whole research field, several open issues and challenges were identified. Additionally various metrics to evaluate the performance of FS methods with multimedia data were presented.

FS in the application domain of renewable energy was considered in [68]. This was chronologically the first study where regression problems had an explicit role. This was illustrated in the more detailed FS reviews on the following: *i*) *Wind Energy Prediction* using NN, Gaussian Process, kNN, ELM, SVR, RF, Boosting machine, and Nonlinear Auto-Regressive models, where FS was performed via optimization-based methods (see Section 2.4), and Empirical Mode Decomposition, *ii*) *Solar Energy Prediction* using correlations, Lasso, and optimization-based but mainly intrinsic (i.e., domain and data-specific) FS methods for NNs, Deep NNs, SVMs, and ELMs; *iii*) *Marine Energy Prediction* using rule- and optimization-based FS methods mainly for ELMs, and *iv*) *Energy-Related Problems* in general using, again, optimization-based FS methods with ELMs and SVMs, RReliefF with NNs, and entropy-based filters with, for e.g., RF and NNs. Over half (18) of the 32 reviewed papers used wrappers for FS.

Genomic big data, similar to [50] as shown above, was addressed in the systematic review in [69]. Most of the identified papers proposed new methods, architectures, and tools for processing genomic data, thus overlapping with other reviews mentioned in this paper. A terminological exception was the Integrative FS methods, which depicted multiple hybrid approaches with different datasets and/or FS methods as a preprocessing step before the actual training of a model (with or without FS).

A systematic review on FS for forecasting spatiotemporal traffic data (how much traffic, where) was presented in [70]. From the FS perspective, the categorization of the identified literature followed the normal model except for the division of filters into so-called *Exogenous and Endogenous feature filtering methods*. The latter encapsulated the typical filters like correlation and sparse linear regression-based methods. Whereas the former referred to the use of external data and knowledge to limit possibilities and useful features, such as, knowing that a car is moving in a specific direction at a certain speed. Additionally, optimization-based wrappers, and embedded methods using, for e.g., deep learning techniques were listed. In 211 papers from 1984–2018, a versatile pool of prediction methods were found including the following: Feedforward shallow and deep ANNs; time-delayed, recurrent, long short-term memory, convolutional, autoregressive exogenous ANNs; Deep belief and Bayesian networks; kNN; autoregressive models; Gaussian Process regression; RF and Regression tree; and Tensor decomposition models. It was concluded that urban traffic forecasting in particular needs further empirical FS studies.

Text classification and FS were the scope of the review in [71]. In this application domain, the starting point is the numerical encoding of texts and documents by using, for instance, the classical bag-of-words representation. This is a particularly interesting domain from the point of view of the distance-based methodology because of the key role of similarity of documents especially in unsupervised scenarios. The classifiers summarized in the review are the common ones: kNN method, NB, relevance-based Rocchio, multivariate regression models, DTs, SVMs, NNs, graph partitioning-based approach, and GA-based methods to train the models. From the FS perspective, the text domain is very similar to the genomic data due to the number of features being large when compared to the number of observations in both. Filters in the field result from preprocessing-like techniques such as DF and TF-IDF, as well as more tradional CBM, MI, IG, Term-Relatedness, χ^2, MD, LR techniques, BNS along with a few special filters. However, wrappers and embedded methods were only briefly addressed in this review. Interestingly and independently, the review was concluded with summarizing some recent FS categories using almost the same topical division as in our umbrella review.

FS in image analysis was considered in [72]. In this domain, one can distinguish low-level, mid-level, and high-level techniques, where the first refers to pixel-/voxel-level tasks like classification and segmentation, the second to the derivation of features and characteristics from images (typically for low-level tasks), and the last, for e.g., to image annotation, i.e., identification of objects and/or their labels. The actual methods and techniques summarized in the review are mostly the same as those already addressed

in many other papers in this umbrella review, with notable exceptions concerning multiple mentions of fuzzy-rough set FS. Out of c. 50 papers reviewed, more than half referred to the use of filters, 13 to embedded techniques, and 11 to wrappers. This review also depicted the main available datasets for FS and performed a small (four datasets times four methods) experiment with the following overall conclusions: results were dependent on all aspects, the classifier, the FS method, and the dataset, with the recommendation to use the subset FS methods with SVM or RF.

2.6. FS in multi-label classification problems

FS in multi-label classification (MLC) problems that are potentially characterized by many simultaneously active labels per instance was presented in [48] using a systematic literature review process. The authors first noted that the use of the straightforward Binary Relevance (BR) method allows for the usage of all single-label FS methods in MLC cases. In the paper, another feature construction method to build binary variables taking into account correlations between multiple labels was depicted. Experiments with 10 MLC datasets, a multi-label extension and the adaptation of kNN-classifier as well as the IG -based filter with BR were presented. The proposed method showed competitive performance with slightly increased computational costs. Finally, the literature review of 99 papers concluded that 70 applied a filter approach in FS.

Another review that focused solely on FS in MLC problems (MLC-FS) was [73]. The MLC-FS was considered from a taxonomy of four perspectives: label, search strategy, interaction with the learning algorithm, and data format. As a whole, this review basically linked the different problem transformation and algorithm adaptation methods of MLC problems with different existing classification models and FS techniques already covered above (for instance, the supervised, semi-supervised, and unsupervised treatment in [67]). In conclusion, the popularity of filters was observed.

Additionally, another review on MLC-FS was undertaken in [74]. Again, the characteristics of addressing MLC problems and a large catalog of existing FS methods were addressed through the analysis of primary publications. The developed taxonomy embeds the known triplet of filter, wrapper, and embedded FS methods into a MLC-specific hierarchy, consisting of direct and transformation-based approaches; the latter was further divided into single and internal/external BR categories.

2.7. Summary

Let us briefly summarize our findings. First, the years covered in different reviews varied substantially, naturally depending on when particular techniques (search- and optimization techniques, classifiers, etc.) actually emerged: for e.g., [70] covered years 1984–2018 and [62] 1983–2019, whereas [63] covered 2012–2020.

Next, we did not find any reviews even mentioning distance-based ML models or focusing solely on FS in regression problems, although [68] mainly considered regression tasks. Regarding classification tasks (see, e.g., [45]), as compared to regression problems, the existence of labels opens up possibilities for both filter methods (e.g., statistical tests to assess how strongly features separate the classes) and for embedded and hybrid methods (e.g., using one classification model for FS and another one as the actual classifier [75]).

Interestingly, many papers noted the existence of cases where filter methods performed either equally or even better than the more complex approaches (e.g., [44,61,50]). Further, for filters, the importance of threshold detection was emphasized in [76].

In general, FS using optimization means the generation of a higher-level search process, which inevitably increases the computational complexity. Other forms of filter and wrapper methods can be more direct: if they can provide a ranking on the importance of the given set of features, then the FS problem reduces to finding a rule that identifies the ranks that are large enough to be included and those that should be omitted from the final model. This is the exact method that is proposed next: Through construction and analysis of the feature importances of the predictive model (one feature sensitivity formula) and the direct generation of the inclusion/exclusion rule means no increase in the overall computational complexity and no addition, iterative search procedure.

To conclude this umbrella review — as readily stated in the first included article, [44], and confirmed (for filters) in one of the last reviewed papers [58] — there does not exist one, single "best method" for FS as different methods have their own strengths and weaknesses [61]. Therefore, identifying a good method for a specific problem setting drives the development of the research field, and in this article, our focus is on FS for regression problems. Our umbrella review shows a major research gap in recent years relating to FS for regression when compared to FS for classification. Therefore, our contributions in this paper seem timely and essential towards filling this gap.

3. Distance-based one-shot wrapper

In this section, we summarize the essence of the distance-based regression model and derive the one-shot wrapper.

3.1. EMLM

EMLM is a supervised distance-based machine learning method. It combines the regularized ridge regression-type learning characteristics of the ELM [77,78] with the distance-based feature map used in the MLM [1,79]. It was proposed by Kärkkäinen [4] and due to its origins, this technique is referred to as EMLM. This model has a structural resemblance to RBFNs with a linear kernel [9,10]. However, the algorithms that select most or even all observations as reference points for the distance-based kernel [4,79] differentiate the overall technique from the RBFNs: reference points for MLM and EMLM are always selected from among observations; not, e.g., as cluster centers. Therefore, the EMLM incorporates only one metaparameter — the number of reference points — and when used with the *RS–maximin* [79] reference point selection algorithm, it provides a deterministic and simple-to-use supervised learning method [4].

The *RS-maximin* method has its origin in the K-means seeding approach [80] known as maximin or the furthest point selection. This seeding approach, in turn, originated from the traveling salesman problem, where it is known as the greedy permutation [81]. The *RS-maximin* approach selects the first reference point as the closest point to the input data mean and then adds the rest of the reference points deterministically with the farthest-first-traversal algorithm. For regression problems, maximizing the input space reference points' pairwise distances is known to improve the MLM's generalization performance [79]. MLM was also found to have the tendency not to overlearn [4,79,82,83].

The training phase of the EMLM is depicted in [Algorithm 3] [4]. Construction of the distance-based regression model starts by computing the distance matrix $\mathbf{H} \in \mathbb{R}^{m \times N}$ as

$$(\mathbf{H})_{ij} = ||\mathbf{r}_i - \mathbf{x}_j||_2, \quad i = 1, \ldots, m, j = 1, \ldots, N, \tag{1}$$

where $\mathbf{r}_i \in \mathbb{R}^n$ is the i:th selected reference point and $\mathbf{x}_j \in \mathbb{R}^n$ denotes the j:th observation. Here, n is the number of features, m denotes the number of selected reference points, and N specifies

the number of observations in the training set. Distance regression weights $\mathbf{W} \in \mathbb{R}^{p\times m}$ are then solved from the linear problem

$$\mathbf{W}\left(\mathbf{H}\mathbf{H}^T + \frac{\alpha N}{m}\mathbf{I}\right) = \mathbf{Y}\mathbf{H}^T, \tag{2}$$

where $\mathbf{Y} \in \mathbb{R}^{p\times N}$ (in the datasets used in this paper, $p = 1$) contains the desired output vectors in its columns, and $\mathbf{I} \in \mathbb{R}^{m\times m}$ is the identify matrix whose multiplier includes the fixed regularization parameter $\alpha = \sqrt{\varepsilon}$ corresponding to the square root of machine epsilon ε. The predicted output $\mathbf{y}^*$ of a trained EMLM for a given input $\mathbf{x}^*$ is $\mathbf{y}^* = \mathbf{W}\mathbf{H}^*$, where $(\mathbf{H}^*)_{i1} = \|\mathbf{r}_i - \mathbf{x}^*\|_2, \quad i = 1,\ldots,m$. The usage of a trained EMLM model consists of computing the distances between new inputs and fixed reference points as shown in (1) and multiplying by the weight matrix $\mathbf{W}$ to calculate the predicted output. The dimensions of $\mathbf{W}$ depend on the number of targets in a dataset. In this paper, we use datasets with single targets, so $\mathbf{W}$ is a row vector of length m.

3.2. Feature scoring using mean absolute sensitivity

Next, we delineate the wrapper approach for the distance-based model. It should be noted that a sampling-based technique for feature scoring and selection with EMLM, similar to [28,29], was proposed and tested in [84].

One form of the classical Taylor's formula as given in [Lemma 4.1.5] [85] reads as follows: in the neighborhood of a point $\mathbf{x}_0 \in \mathbb{R}^n$, there exists $\mathbf{z} \in l(\mathbf{x}, \mathbf{x}_0)$ (a line segment connecting the two points) such that for $\mathbf{y} = \mathbf{x} - \mathbf{x}_0$,

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T\mathbf{y} + \frac{1}{2}\mathbf{y}^T\nabla^2 f(\mathbf{z})\mathbf{y}, \tag{3}$$

where $\nabla f(\mathbf{x}_0)$ denotes the gradient vector at $\mathbf{x}_0$ and $\nabla^2 f(\mathbf{z})$ the Hessian matrix at $\mathbf{z}$, respectively. Usually, this formula is used to underlay a linear approximation or second-order optimization algorithm. Analogous to the latter case, we note that a small value of an individual gradient component $\nabla_i f(\mathbf{x}_0)$ is linked to the weak relevance of the ith feature in depicting the function's local behavior. This observation suggests the inclusion of a feature importance criterion $\mathscr{FI} \in \mathbb{R}^n$ in [35,86], which was based on *Mean Absolute Sensitivity* (MAS) of the training data:

$$\mathscr{FI} = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{\partial \mathscr{M}}{\partial \mathbf{x}_j}\right|, \tag{4}$$

where $\mathscr{M}$ denotes the output of the distance-based regression model. Note that the use of the Cityblock distance makes $\mathscr{FI}$ both robust and independent between the features (see [87]). The analytic derivative of the output with respect to the input vector $\mathbf{x}_j$ is straightforward to compute, yielding a penalized expression similar to that of the unsupervised case in [formula (3)] [88]:

$$\frac{\partial \mathscr{M}}{\partial \mathbf{x}_j} = \mathbf{W}\mathbf{D}^T,$$

where the i:th column $\mathbf{d}_i$ of $\mathbf{D}$ is defined as

$$\mathbf{d}_i = \frac{\mathbf{r_i} - \mathbf{x_j}}{\max(\varepsilon, |\mathbf{r_i} - \mathbf{x_j}|)}, i = 1,\ldots,m.$$

Remark 1. It should be noted that it is not completely clear, especially in the context of FS, whether a distance-based feature map would benefit from a separate bias term. The theoretical basis behind EMLM does not require this [Remark 1] [4], and its omission has also been recommended for the ELM [89]. However, a separate bias is known to enforce an unbiased regression estimate [90], and it could be included in the model simply by enlarging $\mathbf{H}$ into $\begin{pmatrix}\mathbb{1}\\ \mathbf{H}\end{pmatrix}$ where $\mathbb{1}$ denotes the unit matrix of size $\mathbb{R}^{1\times N}$. We conducted a brief experimental pursuit of the question—which has not been reported here—and concluded that a separate bias term added no value. Therefore, the use of the original formulation was confirmed.

Remark 2. MAS-formula (4) to quantify feature importance is independent of the model $\mathscr{M}$; one only needs the model's derivative with respect to features. Basically, this can be obtained using finite differences or automatic differentiation but here we confine ourselves to analytic formulae. A preliminary work with MAS and the analytic derivative of a feedforward neural network (FNN) model, with two transformation layers, was presented in [35]. In order to enable more extensive testing of the MAS-based wrapper approach, we include here the MAS formula for FNNs with any number of layers. The calculus is omitted because it can be performed similarly to [90]. For convenience and building from the previous remark, we assume that the training data has been scaled into $[-1, 1]$, the FNN does not contain bias nodes, and that tanh-functions $\tanh(x) = \frac{2}{1+\exp(-2x)} - 1$ are used as the activation functions throughout (note that the activation functions need to be differentiable which rules out the use of ReLU). Then, a layerwise formalism for the input–output mapping of any FNN with weights $\left\{\mathbf{W}^l\right\}_{l=1}^{L}$ (i.e., weights of layers stored in matrices from the first layer $\mathbf{W}^1$ up to the last layer $\mathbf{W}^L$) can be represented using a *diagonal function matrix* $\mathscr{F} = \mathscr{F}(\cdot) = \mathrm{Diag}\{f_i(\cdot)\}_{i=1}^{m}$, where $f_i \equiv \tanh$, as follows

$$\mathbf{o} = \mathbf{o}^L = \mathscr{M}(\mathbf{x}) = \mathbf{W}^L\mathbf{o}^{(L-1)}, \tag{5}$$

where $\mathbf{o}^0 = \mathbf{x}$ (a given input vector) and $\mathbf{o}^l = \mathscr{F}^l = \mathscr{F}\left(\mathbf{W}^l\mathbf{o}^{(l-1)}\right)$ for $l = 1,\ldots,L-1$. The analytic derivative of such mapping with respect to the input features reads as

$$\frac{\partial \mathscr{M}}{\partial \mathbf{x}} = \mathbf{W}^L\prod_{l=L-1}^{1}(\mathscr{F}\prime)^l\mathbf{W}^l. \tag{6}$$

Algorithm 1 Distance-based one-shot wrapper

Input: Input data $\{\mathbf{x}_j \in \mathbb{R}^n | j = 1,\ldots,N\}$, target data $\left\{y_j \in \mathbb{R} | j = 1,\ldots,N\right\}$
Output: Indices of most important features
1: Train EMLM model using (2) with the full set of features
2: Compute $\mathscr{FI}$ using (4)
3: Sort $\mathscr{FI}$
4: Using Kneedle, find kneepoint of sorted $\mathscr{FI}$ at feature index k
5: Keep features that satisfy $\{i \,|\mathscr{FI}_i \geqslant \mathscr{FI}_k,\ 1 \leqslant i \leqslant n\}$

3.3. Threshold selection

Once the features are ranked according to their score, there needs to be a way to decide how many of them are retained and on what basis. Because the scores, sorted according to their rank, define a 1D curve, the classical knee-point could be used to identify a change in the characteristic behavior [91]. A widely used technique for knee-point detection is to maximize the curvature, for which explicit formula is given in [92]. This is realized in a readily

implemented kneepoint detection algorithm, Kneedle [93], which identifies the cutoff point of a smoothed curvature $\frac{f''(x)}{\left(1+f'(x)^2\right)^{1.5}}$.

The proposed one-shot FS algorithm is detailed in Algorithm 1. Fig. 1 illustrates the use of a kneepoint and Kneedle for FS with the EMLM and the MAS formula (4). In the figure, the mean validation error and the standard deviation for it is shown using blue, while the MAS-values representing each number of features is shown in orange. The simulated dataset for the demonstration is defined in Section 4.1.

4. Experimental setup

In this section, we detail our experimental design related to selected datasets, compared methods, and evaluation metrics. We also compare our proposal [1] with popular FS methods using a representative set of synthetic and benchmark datasets. We utilize the area under the receiver operating characteristic curve as an evaluation metric with the synthetic datasets and the root-mean-square error with the benchmark datasets.

4.1. Datasets

Here, we present the datasets used in the experiments. The use of readily available benchmark datasets is augmented by the use of synthetic data, which is similar to [94].

Synthetic. We created a set of synthetic datasets to analyze the goodness of feature importance scoring. We can utilize ranking-based evaluation metrics when the ground truth features are available. Therefore, with the synthetic datasets, we can focus on the primary problem of FS independently from the thresholding of the feature importance score. We used two sets of synthetic datasets: one (denoted Y_{Rx}) that has already been used in other studies [94] and a set inspired by the first (denoted Y_{Ax}). The functions used to generate the synthetic datasets are presented in Table 2. The last row containing Y_{R4} consists of two equations forming a spiral equation. The datasets $Y_{A1}, Y_{A3}, Y_{A4}, Y_{A5}$, and Y_{A6} have progressive complexity. The Y_{A2} dataset is the most challenging, since it mostly represents incoherent noise. However, it can show if a feature ranking algorithm will find results that are not practically there. Thus, it functions as a sanity check. For the Y_{A1}-Y_{A6} datasets, half of the features are true ones, while for the Y_{R1}-Y_{R4} datasets, there are only one to four true features. For each synthetic dataset, we generated 1000 observations with 200 features ($N = 1000, n = 200$). For datasets Y_{A1}-Y_{A6} and Y_{R1}-Y_{R3}, the features were randomly generated as defined in

$$x_i = \mathscr{U}[0,1)\ i \in [0,\cdots,n-1]. \tag{7}$$

For dataset $Y_{R4}, Y = \mathscr{U}[0,20)$. After the target of a synthetic dataset has been calculated, the dataset target is augmented with Gaussian noise of zero mean and unit variance, which is augmented by normalizing it with the maximum difference to prevent egregious "measurement errors."

Benchmark datasets. A group of openly available datasets, also mentioned in the umbrella review in Section 2, were used to get comparable results. The benchmark datasets' characteristics are presented in Table 3, where the column headers #Obs., #Feat., #Trgt. refer to the numbers of observations, features and targets, respectively. Column #Un.Trgt. refers to the number of unique values found in the target vector, while header T. refers to the dataset type (regression R, classification C) and Src to the source of the dataset. We have used the first target when the dataset had more than one target.

[1] Source codes available at: https://gitlab.jyu.fi/hnpai-public/extreme-minimal-learning-machine/

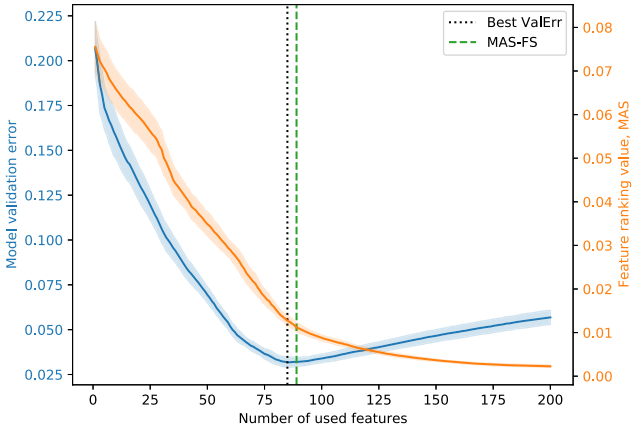

Fig. 1. Example of the cutoff point given by MAS kneepoint (in green) using the synthetic dataset Y_{A1} and the computed and sorted *MAS* values. The MAS-kneepoint provided the correct set of features that minimized the validation error.

Table 2
Synthetic datasets. Column F_T represents the number of true features, while column F_F represents the number of false features.

Function	F_T	F_F
$Y_{A1} = \sum_{i=0}^{99}(99-i)x_i + \mathcal{N}(0,1)$	100	100
$Y_{A2} = \sum_{i=0}^{99}\sin(2\pi(99-i)x_i) + \mathcal{N}(0,1)$	100	100
$Y_{A3} = \sum_{i=0}^{99}(99-i)x_i^2 + \mathcal{N}(0,1)$	100	100
$Y_{A4} = \sum_{i=0}^{99}(99-i)x_i^6 + \mathcal{N}(0,1)$	100	100
$Y_{A5} = \sum_{i=0}^{99}(99-i)e^{x_i} + \mathcal{N}(0,1)$	100	100
$Y_{A6} = \sum_{i=0}^{99}(99-i)\log(1-x_i) + \mathcal{N}(0,1)$	100	100
$Y_{R1} = -2\sin(2x_0) + x_1^2 + x_2 + e^{x_3} + \mathcal{N}(0,1)$	4	196
$Y_{R2} = x_0 e^{2x_1} + x_2^2 + \mathcal{N}(0,1)$	3	197
$Y_{R3} = \sin(2\pi x_0) + \mathcal{N}(0,0.1)$	1	199
$\begin{cases} x_0 = Y_{R4}\sin(Y_{R4}) + \mathcal{N}(0,1) \\ x_1 = Y_{R4}\cos(Y_{R4}) + \mathcal{N}(0,1) \end{cases}$	2	198

It should be noted that due to a lack of benchmark regression datasets for FS, we also used a set of classification datasets (see Table 3) in a separate experiment. The aim was to observe how our proposed FS algorithm would function with datasets for which it was not designed.

4.2. Compared ranking and FS method

The quality of the *MAS* score was first compared to the most common ranking algorithms of filters and embedded methods.

Table 3
Benchmark datasets.

Dataset	#Obs.	#Feat.	#Trgt.	#Un.Trgt.	T.	Src
StudentTest	258	5	1	4	C	[95]
ATP1D	337	411	6	83	R	[96]
COIL	1800	21	1	100	C	[95]
Madelon	2000	500	1	2	C	[95]
Outdoor	2400	21	1	40	C	[95]
ThyroidAnn	3772	21	1	3	C	[95]
OptDigits	3823	64	1	10	C	[95]
SatImage	4435	36	1	6	C	[95]
COIL2000	5822	85	1	2	C	[95]
RF2	7679	576	8	515	R	[96]
ComputerActivity	8192	21	1	56	R	[97]
SCM1D	9803	280	16	1092	R	[96]
Census	22784	8	1	2045	R	[97]

Here, we used *RreliefF*, *SpearmanR*, *Mutual-Info*, *Fisher-score* and *Mean Absolute Difference* (MAD) as non-model-based comparisons. We also included model-based comparisons, namely *DT* and *RF*. Both model based ones were used only to gain a rank for each feature in the feature ranking experiment.

The quality of the distance-based regression model after feature ranking and one-shot selection was then compared to a selected group of well-known reference models, which have commonly been referenced in the literature. The selected group consisted of linear models, *Linear Regression*, *Ridge Regression*, and *Lasso* as well as tree-based models, *Decision Tree* and *Random Forest* and finally *SVR*. The methods are all readily available in the Python library Scikit-Learn [98].

4.3. Evaluation criteria

The quality of feature ranking and selection were assessed with multiple evaluation criteria, which are discussed next. The starting point for the evaluation is the existence of a separate validation dataset that can be used to assess FS performance and the resulting data-driven models. Moreover, since the specific true features of the synthetic datasets are known, we can directly count the number of true and false features. However, due to random number generation being involved in data generation, these counts may differ from the intended ground truth in rare occasions, which must be considered when looking at evaluation results. The used evaluation criteria are as follows:

Root Mean Square Error (RMSE) is a standard way to compute the validation error of a regression model [99]: $RMSE = \sqrt{\frac{\sum_{i=0}^{P}\left(y_i^* - y_i\right)^2}{P}}$, where P is the number of observations in the validation set, y_i^* is the predicted validation target, and y_i is the true validation target.

Kruskal–Wallis H test is a well-known statistical significance test between two data groups [100]. Because the test is non-parametric, data does not need to be from a normal distribution. We used significance level 0.05 for the H test. Any p-value lower than 0.05 indicates that the two tested groups have significant differences. In our results, we took the best achieved result and compared the other results to it in a pairwise manner.

According to Table 3 of the review by Solorio-Fernández et al. [14], there is no proper consensus on how to verify that a feature ranking algorithm works. Consequently, we have opted to use a couple of different measures for verification purposes. **Area under receiving operating characteristic** (AUROC) is another way to measure and quantify the quality of a feature ranking, when the true features are known for a dataset. Teisseyre [101] defined the receiving operating characteristic curve (ROC) as $(FPR(k), TPR(k))$, where $k = 1, \ldots, m$ refers to the top k selected features, *FPR* = false positive rate among the top k selected features and *TPR* = true positive rate among the top k selected features. When the ROC is paired with the area under curve, we get a single number describing the performance of a feature ranking algorithm. The measurement intuitively explains a performance. The more correctly the features are ranked, the closer the score is to 1. Scores close to 0.5 indicate random selection, while scores close to 0 mean that correct features are specifically not selected.

Number of features for best validation error, n_{sel}, is what the name implies. The number of features in the subset of features that provide the lowest validation error is taken, and the mean is calculated from the achieved numbers of features. This allows the observation of what portion of features are required by a feature ranking algorithm to reach its best-achieved result. Since the goal of FS is to remove as many features as possible, a smaller feature subset is preferable.

4.4. Feedforward Neural Networks

In addition to comparing *EMLM*–based *MAS-FS* to the popular feature selection methods, we also performed experiments with the MAS for FNNs (see Remark 2). We used two versions of Sequential model from Tensorflow [102]: A single hidden layer + linear output layer (denoted as *FNN-2*) and three hidden layers + linear output layer (denoted as *FNN-4*). Sizes of the Tensorflow's dense-layers were fixed to:

$$\begin{aligned} \text{FNN} - 2 &= (\lceil n_s/2 \rceil, 1), \\ \text{FNN} - 4 &= (\lceil n_s/2 \rceil, \lceil n_s/4 \rceil, \lceil n_s/8 \rceil, 1), \end{aligned}$$

where n_s is the number of features (after dataset preprocessing (removal of constant features).

5. Experiments and results

In this section, we present the experiments and their results based on the datasets and evaluation criteria depicted in the previous section. The discussion follows the steps of the overall MAS-based FS algorithm (MAS-FS) for the distance-based EMLM as follows: feature ranking component, testing against commonly known algorithms, and finally testing against *RF*.

The experiments were run on a local computation cluster on a single node (2x Intel Xeon Gold 6148) using Python 3.8. We used the following library versions: *i*) NumPy: 1.20.1 [103]; *ii*) SciPy: 1.6.1 [104]; *iii*) Scikit-learn: 0.24.1 [98]; and *iv*) Kneed: 0.7.0 [93].

5.1. Assessing the quality of feature scores and rankings

Our first step was to compare the MAS scoring with a representative set of other techniques depicted in Section 4.2. Using the synthetic datasets presented in Table 2, we compared the FS algorithms' ability to generate feature scores and the corresponding ranking to correctly order the features.

We randomly split each synthetic dataset 30 times into training and validation partitions, where 33% of the generated dataset composed the validation set. Each of the 30 training partitions were input to a feature ranking algorithm. The features were scored and ordered, which with the known ground truth allowed us to then use AUROC (presented in Section 4.3) to compute a quality measure.

In addition, we computed the validation error for each subset of the ordered features in each of the 30 splits. Specifically, we used EMLM (85% of data used as reference points) to compute the validation error $E_{\text{val}}(1)$ for the rank #1 feature, $E_{\text{val}}(2)$ for rank #1 and #2 features, $\ldots, E_{\text{val}}(n)$ for all features. Then, we found the number of features for $min(E_{\text{val}})$. Finally, we reported the mean and standard deviation for each AUROC value and each $min(E_{\text{val}})$. The results are shown in Table A.4.

The best values, the highest AUROC (denoted in table as AUC), and the lowest n_{sel}, are in bold for each dataset. In addition, we used ◆ to show that a value is statistically close to the designated best value (using the Kruskal–Wallis H test).

We will discuss the results presented in Table A.4 in the order of feature ranking algorithms.

MASvec achieved the lowest number of features in seven cases out of 10 and the lowest mean validation errors in eight cases out of 10. With these results, it had the best performance among the tested feature ranking algorithms. Further, it is the only algorithm that did not select extra features for datasets $A1 - A6$ from the false feature pool, as it did not return more than 100 features. However, it slightly missed the intended number of features with datasets $R1, R2$, and $R4$.

RreliefF is a traditional feature ranking algorithm that is often praised. However in our experiments, *RreliefF* failed to impress. Regarding datasets $A1-A6$ and $R3$, *RreliefF* performed as if it was a random selector. While the algorithm found better results with datasets $R1, R2$ and $R4$, overall its performance was found lacking. Thus, it can concluded that the algorithm is not suited for datasets where the input features closely resemble each other.

SpearmanR performed best in datasets $R1-R4$ and relatively well in datasets $A1-A6$, excluding $A2$. This indicates that it is better suited to datasets where there are relatively few correct features.

MI is a popular ranking algorithm. In our experiments, it performed better than *RreliefF*, but was still surprisingly close to a random selector with datasets $A1-A6$. However, *MI* is one of the three tested algorithms that managed to find the correct number of features with dataset $R4$. Dataset $R4$ has spiral-like data, and we expect the form of the dataset to play a role in the performance of *MI*.

Based on the AUROC values, *DT* performed in a similar manner to *MI*: it was surprisingly close to random selection with datasets $A1-A6$ and performed significantly better with datasets $R1-R4$. As *DT* is an iterative method, it may require more training time than what is provided by the default settings to be able to properly handle a situation with more than a few correct features.

RF is the third of the three that found the correct number of features with dataset $R4$, and it performed similar to *MI* and *DT*. We expect that *RF Ranker* has the same needs as *DT* and would require changes to the default settings.

Fisher-score was the worst–performing feature ranking algorithm that we tested. Unlike other methods that resembled random selection at their worst, it actively selected wrong features in the $A1-A6$ datasets. However, the same effect was not present with datasets $R1--R4$, even though *Fisher-score* was closer to random selection with datasets $R1$ and $R2$. Thus, we must conclude that *Fisher-score* is not suitable for datasets like $A1--A6$ and $R1--R4$.

Mean Absolute Difference (MAD) mostly resembled a random selector, with the exceptions being with datasets $R1-R4$. Especially of note is the result for dataset $R4$, since it completely failed

Table A.4
Feature-ranking algorithm results for the eight feature ranking algorithms and for the 10 synthetic datasets. The best mean value per dataset has been highlighted in bold. The results where the population median is the same as for the best mean value according to the Kruskal–Wallis H test are marked using ◆.

Algorithm		MASvec		RreliefF		SpearmanR		Mutual-Info	
Dataset	Var	$\overline{x}$	δ	$\overline{x}$	δ	$\overline{x}$	δ	$\overline{x}$	δ
A1	AUC	**0.93**	0.01	0.50	0.05	0.80	0.02	0.57	0.05
	n_{sel}	**86.67**	3.57	198.00	2.21	141.13	18.67	169.63	30.47
A2	AUC	0.49	0.03	0.49	0.03	0.50◆	0.04	0.51◆	0.05
	n_{sel}	78.83◆	80.62	51.03◆	41.12	54.07◆	48.60	49.57◆	35.36
A3	AUC	**0.91**	0.01	0.50	0.04	0.79	0.01	0.58	0.04
	n_{sel}	**80.20**	3.96	198.57	1.94	131.53	20.87	142.77	20.22
A4	AUC	**0.87**	0.02	0.50	0.03	0.79	0.02	0.59	0.04
	n_{sel}	**65.53**	5.00	197.17	3.53	77.50	17.43	144.93	19.59
A5	AUC	**0.93**	0.01	0.50	0.04	0.81	0.02	0.55	0.04
	n_{sel}	**85.27**	3.93	197.87	2.80	145.37	27.68	167.93	27.99
A6	AUC	**0.86**	0.01	0.49	0.04	0.77	0.02	0.57	0.05
	n_{sel}	**74.87**	10.26	197.63	2.66	106.27	20.69	147.90	25.73
R1	AUC	**0.99**	0.00	0.80	0.10	0.99◆	0.00	0.97	0.03
	n_{sel}	4.93◆	1.06	98.50	57.89	5.00◆	0.97	23.00	21.29
R2	AUC	**0.99**	0.00	0.79	0.11	0.99◆	0.00	0.99	0.00
	n_{sel}	**3.13**	0.34	20.13	28.71	3.13◆	0.43	3.93	1.39
R3	AUC	**0.99**	0.00	0.44	0.27	0.99◆	0.00	0.99◆	0.00
	n_{sel}	**1.00**	0.00	69.80	52.54	1.00◆	0.00	1.00◆	0.00
R4	AUC	0.95	0.03	0.90	0.09	0.96	0.01	**0.99**	0.00
	n_{sel}	5.37	4.21	7.83	5.77	5.17◆	4.89	**2.00**	0.00

Algorithm		DecisionTree		RandomForest		Fisher-score		MAD	
Dataset	Var	$\overline{x}$	δ	$\overline{x}$	δ	$\overline{x}$	δ	$\overline{x}$	δ
A1	AUC	0.58	0.04	0.74	0.03	0.21	0.05	0.48	0.03
	n_{sel}	195.13	8.07	174.00	21.21	197.07	4.87	199.80	0.40
A2	AUC	0.51◆	0.04	**0.52**	0.03	0.19	0.04	0.50	0.02
	n_{sel}	62.00◆	49.82	61.93◆	55.80	57.60◆	47.92	**42.13**	27.41
A3	AUC	0.56	0.04	0.76	0.02	0.20	0.05	0.46	0.02
	n_{sel}	197.13	3.19	161.40	21.66	197.67	4.56	198.90	1.14
A4	AUC	0.57	0.03	0.76	0.02	0.22	0.08	0.52	0.02
	n_{sel}	195.37	5.38	80.80	18.67	194.53	7.01	197.40	5.09
A5	AUC	0.56	0.03	0.74	0.02	0.19	0.06	0.49	0.02
	n_{sel}	196.03	6.13	167.40	22.49	197.70	4.09	198.53	1.61
A6	AUC	0.57	0.04	0.74	0.03	0.20	0.04	0.50	0.02
	n_{sel}	195.67	6.19	134.33	34.96	196.30	4.38	197.37	4.59
R1	AUC	0.99◆	0.00	0.99◆	0.00	0.52	0.05	0.35	0.07
	n_{sel}	**4.87**	0.88	5.00◆	1.03	128.53	7.66	179.00	17.40
R2	AUC	0.99◆	0.00	0.99◆	0.00	0.58	0.07	0.40	0.10
	n_{sel}	3.30◆	0.64	3.27◆	0.51	126.33	16.63	132.53	40.06
R3	AUC	0.99◆	0.00	0.99◆	0.00	0.86	0.22	0.64	0.24
	n_{sel}	1.03◆	0.18	1.00◆	0.00	27.40	46.68	71.00	50.85
R4	AUC	0.99◆	0.00	0.99◆	0.00	0.66	0.10	0.00	0.00
	n_{sel}	2.00◆	0.00	2.00◆	0.00	11.00	32.60	52.83	37.17

to find the correct feature. We can conclude that *MAD* does not work well with a spiral-like dataset.

Based on the results, we can conclude that *MAS-FS* ranked first in the test. This confirms its basic utility to be used for feature ranking with the distance-based EMLM. Moreover, it should be noted that all ranking methods had problems with the Y_{A2} dataset. This was expected as it mostly resembles random noise and is the most difficult of the synthetic datasets.

5.2. Comparison of Algorithm 1 to reference algorithms with regression datasets

In order to assess the entire FS algorithm given in Algorithm 1, we compared it to other approaches using publicly available regression datasets, which are presented in Section 4.1. Similar to experiments in Section 5.1, each dataset had 30 different training/-validation splits (with validation partition using 33% of the whole data). Because the number of reference points has a quadratic effect on the computational effort of *EMLM*, we provide results with two reference point percentages for the results presented in Appendix B, 65% and 85%, and additionally with 100% for the results presented in Appendix C. We were interested in observing how much accuracy might be lost with a reduced number of reference points.

Results of these experiments are given in Table B.5. *MAS-FS* was included with two reference point percentages. In almost all cases, the results between 65% and 85% were close to each other based on the Kruskal–Wallis test, the exception being *ComputerActivity* dataset, where the result for 65% was better than it was for 85%. This indicates that between the two, the higher reference point percentage did not have a meaningful effect on the outcome of the FS. This indicates that for the purposes of FS, *MAS-FS* is robust enough that a lower reference point percentage is recommended. This is also because the lower reference point percentage requires fewer computations. We point out that this conclusion is given in the context of FS: after selecting the final feature set, the portion of reference points for the corresponding EMLM model can be selected independently. *MAS-FS* had either the best RMSE value or was close to the best RMSE value based on the Kruskal–Wallis test in four datasets out of five, thus coming out on top of the tested FS methods. Altogether, MAS-FS was the best or statistically equally good as the best in 3/5 cases. *RF* had the best value or was statistically similar to the best value in two datasets out of five. *Lasso* did not receive the best RMSE values, but it is noteworthy that *Lasso* had the lowest standard deviation in the RMSE values in four datasets out of five, indicating that it was the most consistent of the tested FS methods. Of the tested methods, no other clear noteworthy results were gained.

5.3. Algorithm 1 with FNN on synthetic and regression datasets

Next we conclude the experiments where FNN and the corresponding feature sensitivity formula were used in Algorithm 1. This simply means that the EMLM and the $\mathscr{FI}$ formula in Steps 1 and 2 of Algorithm 1 were replaced with the FNNs as defined in Section 4.4 and the sensitivity formula given in (6). Each dataset had 30 different training/validation splits (with validation partition using 33% of the whole data). The results of these experiments are given in Tables C.6 and C.7. As can be seen from Table C.6 and Table C.7, the mean RMSE validation errors ($\overline{e}$ in tables) indicates that the EMLM-based MAS-FS is able to achieve lower errors than either of the tested FNN version. Another noteworthy observation, although expected, is that the deeper model FNN-4 achieves better accuracy than FNN-2 in all cases. We can also point out that the result for *FNN-4* begins to approach the result for *EMLM* with *Census* dataset. From the data-driven model construction perspective, use of *FNN* and *EMLM* have significant differences. Whereas selection of the portion of reference points is sufficient for *EMLM*, with *FNN* one could tune the number of epochs and the size of batches per epoc in training, the number of hidden layers, the number of neurons in each layer, the activation function in each neuron etc. In addition, *EMLM* is fully deterministic but *FNN* is not and may require multiple training rounds in the hopes of improving the model. Thus, we assume that the *FNN* results could be improved by significantly increasing the amount of time used in hyperparameter optimization and assessing different models. However, these results show that the feature sensitivity based FS can be generalized to completely different models compared to the kernel-like *EMLM* and that the generalization capability of the *EMLM-MAS-FS* algorithm compared to FNN-based versions is promising.

5.4. Comparison of Algorithm 1 to reference algorithms with classification datasets

For our last experiment, we compared the *MAS-FS*-based FS to the available implementation of RF for a regression task. Similar to experiments in Section 5.1, each dataset had 30 different training/validation splits (with validation partition using 33% of the whole data). Some of the classification datasets came with their own validation dataset. For these, the provided validation dataset was first combined with the rest of the data and then split into train/validation sets as was done with the other datasets.

The results for the mean RMSE validation error are given in Table D.8. The table contains results for three different reference point percentages for *MAS-FS*, as *RF* does not have the same

Table B.5
Comparison of MAS-FS to reference ML models using regression datasets. Header $\overline{e}$ denotes the mean validation error calculated with RMSE and header δ its standard deviation. Best mean validation error per dataset has been marked with bold text. Results where the population median is the same as for the best mean value (Kruskal–Wallis H test) are marked with ◆.

Dataset	ATP1D		RF2	
Algorithm	$\overline{e}$	δ	$\overline{e}$	δ
MAS-FS, 65%	$9.42e-2$	$4.02e-3$	$7.70e-2^{\blacklozenge}$	$6.21e-3$
MAS-FS, 85%	$9.40e-2$	$4.40e-3$	$\mathbf{7.68e-2}$	$6.17e-3$
DecisionTree	$8.65e-2$	$2.28e-3$	$1.16e-1$	$1.45e-2$
Lasso	$1.06e-1$	$1.00e-4$	$1.76e-1$	$3.62e-3$
LinearRegression	$5.13e2$	$2.76e3$	$4.39e-1$	$1.07e-1$
RandomForest	$\mathbf{5.97e-2}$	$1.33e-3$	$8.10e-2$	$6.42e-3$
Ridge Regression	$8.55e-2$	$5.63e-4$	$8.80e-2$	$7.55e-3$
SVM	$7.67e-2$	$7.58e-4$	$9.25e-2$	$5.06e-3$

Dataset	SCM1D		ComputerActivity	
Algorithm	$\overline{e}$	δ	$\overline{e}$	δ
MAS-FS, 65%	$3.32e-3^{\blacklozenge}$	$3.45e-4$	$2.69e-2$	$3.41e-3$
MAS-FS, 85%	$\mathbf{3.29e-3}$	$3.51e-4$	$3.33e-2$	$1.28e-2$
DecisionTree	$1.12e-2$	$3.62e-3$	$3.74e-2$	$2.09e-3$
Lasso	$2.09e-1$	$1.16e-4$	$1.86e-1$	$3.80e-5$
LinearRegression	$1.79e-2$	$1.14e-3$	$9.80e-2$	$3.07e-3$
RandomForest	$7.84e-3$	$2.26e-3$	$\mathbf{2.53e-2}$	$9.24e-4$
Ridge Regression	$2.27e-2$	$6.58e-4$	$9.80e-2$	$2.20e-3$
SVM	$5.29e-2$	$6.98e-4$	$4.76e-2$	$2.00e-3$

Dataset	Census	
Algorithm	$\overline{e}$	δ
MAS-FS, 65%	$3.70e-2^{\blacklozenge}$	$1.27e-3$
MAS-FS, 85%	$\mathbf{3.64e-2}$	$1.35e-3$
DecisionTree	$5.64e-2$	$1.97e-3$
Lasso	$1.46e-1$	$1.07e-3$
LinearRegression	$5.05e-2$	$1.07e-3$
RandomForest	$4.00e-2$	$1.17e-3$
Ridge Regression	$4.94e-2$	$1.07e-3$
SVM	$5.20e-2$	$1.02e-3$

parametrization, it has one result per dataset. Additionally for *MAS-FS*, we show the remaining number of features after FS as a percentage as well as the standard deviation for it. Of the eight classification datasets, *MAS-FS* has the best RMSE error in five cases. In general, the three different reference point percentages for *MAS-FS* produced similar mean validation errors, leading to the same conclusion as was made with *MAS-FS* in Section 5.2. The dataset properties as well as the number of observations, features, and unique features do not provide indications on whether there is a pattern to the observed differences in terms of the mean validation errors between *MAS-FS* and *RF*. The type of the input (integer/float) did not provide any insight either. Thus, $\overline{n_{\%}}$ was added

Table C.6
Comparison of FNN-based ranking to EMLM-based ranking using regression datasets. Header $\overline{e}$ denotes the mean validation error calculated with RMSE and header δ its standard deviation. Best mean value per dataset has been marked with bold text. Results where the population median is the same as for the best mean value (Kruskal–Wallis H test) are marked with ◆.

	EMLM 85%		FNN-2		FNN-4	
Dataset	$\overline{e}$	δ	$\overline{e}$	δ	$\overline{e}$	δ
A1	**4.51e − 2**	1.20e − 2	1.91e − 1	2.82e − 2	1.10e − 1	1.52e − 2
A2	**8.48e − 2**	2.84e − 3	1.84e − 1	2.16e − 2	1.04e − 1	1.31e − 2
A3	**4.14e − 2**	8.96e − 3	1.84e − 1	2.06e − 2	1.05e − 1	1.52e − 2
A4	**5.75e − 2**	4.49e − 3	1.84e − 1	2.45e − 2	1.08e − 1	1.64e − 2
A5	**4.52e − 2**	1.07e − 2	1.91e − 1	3.20e − 2	1.14e − 1	1.41e − 2
A6	**5.57e − 2**	7.14e − 3	1.93e − 1	2.61e − 2	1.07e − 1	1.43e − 2
R1	**7.42e − 2**	6.45e − 3	1.93e − 1	2.86e − 2	1.06e − 1	1.16e − 2
R2	**6.63e − 2**	8.77e − 3	1.83e − 1	2.33e − 2	1.07e − 1	1.26e − 2
R3	**1.59e − 1**	3.69e − 3	2.26e − 1	1.94e − 2	1.67e − 1	9.23e − 3
R4	**1.53e − 1**	3.03e − 3	2.20e − 1	1.99e − 2	1.65e − 1	9.55e − 3

Table C.7
Comparison of FNN-based ranking to EMLM-based ranking using regression datasets. Header $\overline{e}$ denotes the mean validation error calculated with RMSE and header δ its standard deviation. Best mean value per dataset has been marked with bold text. Results where the population median is the same as for the best mean value (Kruskal–Wallis H test) are marked with ◆.

	EMLM 85%		FNN-2		FNN-4	
Dataset	$\overline{e}$	δ	$\overline{e}$	δ	$\overline{e}$	δ
ATP1D	**3.85e − 2**	2.98e − 3	2.31e − 1	8.65e − 2	1.52e − 1	6.78e − 2
RF2	**1.71e − 3**	2.34e − 4	5.06e − 2	1.50e − 2	2.38e − 2	4.32e − 3
SCM1D	**1.78e − 2**	6.22e − 4	3.35e − 2	3.65e − 3	2.95e − 2	1.68e − 3
ComputerActivity	**1.45e − 2**	3.78e − 3	1.49e − 1	3.84e − 2	8.99e − 2	2.31e − 2
Census	**3.56e − 2**	4.64e − 3	5.16e − 2	4.81e − 3	4.86e − 2	4.35e − 3

Table D.8
Results of the comparison between MAS-FS and *RF*. The same notations from the tables above are used with the addition of $\overline{n_{\%}}$ (percentage of features remaining after FS) and $\delta n_{\%}$ (standard deviation for $\overline{n_{\%}}$).

Algorithm		MAS-FS				RandomForest	
Dataset	RefP	$\overline{e}$	δ	$\overline{n_{\%}}$	$\delta n_{\%}$	$\overline{e}$	δ
StudentTest	65	6.55e − 2◆	8.74e − 3	47%	9%	7.19e − 2	7.44e − 3
	85	6.56e − 2◆	8.31e − 3	43%	7%	-	-
	100	**6.48e − 2**	7.31e − 3	41%	5%	-	-
COIL	65	7.67e − 2	3.10e − 3	57%	10%	**7.30e − 2**	3.23e − 3
	85	7.53e − 2	3.37e − 3	59%	13%	-	-
	100	7.46e − 2	3.22e − 3	57%	10%	-	-
Madelon	65	5.18e − 1	4.81e − 3	88%	5%	**3.94e − 1**	5.78e − 3
	85	5.18e − 1	5.18e − 3	89%	6%	-	-
	100	5.19e − 1	5.32e − 3	87%	6%	-	-
Outdoor	65	9.52e − 2	8.24e − 3	66%	17%	1.06e − 1	4.97e − 3
	85	**8.84e − 2**	8.59e − 3	70%	18%	-	-
	100	8.90e − 2	7.70e − 3	65%	20%	-	-
OptDigits	65	8.10e − 2	1.74e − 3	91%	5%	1.05e − 1	3.93e − 3
	85	7.93e − 2◆	1.89e − 3	91%	5%	-	-
	100	**7.88e − 2**	1.82e − 3	91%	5%	-	-
ThyroidAnn	65	1.00e − 1	4.69e − 3	10%	0%	**3.93e − 2**	6.25e − 3
	85	1.00e − 1	4.68e − 3	10%	0%	-	-
	100	1.00e − 1	4.67e − 3	10%	0%	-	-
SatImage	65	1.04e − 1◆	3.10e − 3	70%	20%	1.14e − 1	3.71e − 3
	85	1.03e − 1◆	3.57e − 3	70%	19%	-	-
	100	**1.03e − 1**	4.00e − 3	71%	18%	-	-
COIL2000	65	**2.58e − 1**	4.41e − 3	83%	26%	2.61e − 1	3.32e − 3
	85	2.65e − 1	6.28e − 3	77%	31%	-	-
	100	2.71e − 1	4.83e − 3	82%	27%	-	-

in a bid to provide an explanation for the results, but a clear correlation was not found. However, as some of the results indicate (see *Madelon*), the kneepoint detection algorithm Kneedle can behave conservatively, leaving a large set of features to the obtained model. This suggests that, in some cases, it could be beneficial to repeat the algorithm to the once-reduced feature set. On the other hand, this works to the strengths of the distance-based EMLM since it is robust and is capable of handling extra features without a loss of accuracy.

5.5. Summary of the experimental results

Here we discuss the experimental results as a whole. Overall, our proposed FS algorithm performed well. On a more specific note, the feature ranking component in *MAS-FS* was able to determine the feature importance rather accurately, which then allowed the kneepoint detection algorithm to perform the actual FS. In the extensive experimental comparison, it was shown that the proposed method was better than the *RF* with both regression and classification datasets. Moreover, MAS-FS with EMLM can determine the scores and rankings of features for both the original, full set of features as well as the final, selected feature subset.

Of the synthetic datasets, Y_{A2} was the most problematic for all tested feature ranking algorithms, which we expected due to how the dataset is formed. Further, the AUROC-score revealed that except for the *Fisher-score*, the features in Y_{A2} were basically selected randomly. For the $Y_{A1} - Y_{A6}$ datasets, the AUROC values for the *Fisher-score* are below 0.25, implying that reversing the feature ranking would improve performance.

We included two versions of our algorithm for the regression dataset tests, for which we used two different reference point percentages and three versions for the classification dataset tests. Based on the results, that show that the performance between the reference point percentages was so similar, we recommend using 65% for feature ranking and selection as it is computationally lighter than 85%. After the ranking process, we recommend selecting as high a reference point percentage as possible due to the tendency of *EMLM* to not overlearn [4]. The comparison between *MAS-FS* using *EMLM* vs. *FNN* provided the knowledge that using our *MAS-FS* algorithm performs better with *EMLM* at its core at similar levels of researcher setup.

6. Conclusions

In FS, filters are used due to their speed and simplicity even if they often do not possess the best possible accuracy. Meanwhile, wrappers are used for their accuracy, but they require a search component that makes them slow and computationally expensive. A common practice is to combine the two by first applying a filter to quickly reduce the workload and then finishing with a wrapper. Our FS algorithm is a wrapper since it uses the distance-based model of *EMLM*, but it is a wrapper without a search component. This makes our algorithm simple, straightforward, and efficient. Since there is no search component, there is no iterative component either, implying that the feature importance scoring is conducted using a one-shot procedure.

We discovered that regression benchmark datasets for FS (especially with the ground truth features) are rarely available in the literature. Therefore, we presented a group of synthetic datasets ($Y_{A1} - Y_{A6}$), which were designed to have easily understandable relations between the feature importances. This allows them to function as a sanity check for a FS algorithm and as an assurance that the algorithm works properly. Moreover, the availability of ground truth features allows for the usage of feature ranking based performance measures. Indeed, current literature seldom discusses FS in the regression context and has not discussed it in relation to distance-based ML models. We have positioned our umbrella review to provide a thorough background into the topic of this paper.

This paper proposed a new FS approach for a distance-based supervised machine learning model referred to as the EMLM. Subsequently, we evaluated the proposed method with an extensive set of synthetic and real datasets and compared it to popular approaches. In addition, we presented a thorough umbrella review, which is the first, on the topic of FS.

Our experimental results for a representative set of synthetic datasets showed that the regression model sensitivity-based feature importance scoring outperformed other methods in terms of feature ranking quality. Further, the proposed method can identify underlying non-linear, input–output data relations hidden in a large set of noisy features. The experimental results for the real datasets also showed that the proposed one-shot wrapper approach, which straightforwardly utilizes the model's sensitivity-based feature ranking outperformed (although with a slight margin) the popular methods like the RF.

In order to adapt the proposed FS method to other machine learning models, we derived a general MAS-FS formula for those FNN architectures which are differentiable with respect to features. We performed an experimental comparison with two off-the-shelf DL architectures, which demonstrated the adaptability of the proposed FS approach. However, the experimental results showed that the distance-based method with the one-shot wrapper outperformed these DL architectures. These results indicate that the DL architectures require more fine-tuning of parameters and data to obtain the same level of accuracy as this distance-based method.

As for future work, a natural extension of our study would be the application of the FS techniques to reduce features from the distance regression model in the first phase of the MLM [1,79]. Similarly, the encouraging initial assessment of MAS-based feature scoring and ranking for classification tasks, as given in [86], is to be extended to the full FS framework along the lines of this article. This is a prime example of a multi-output problem [105] where use of the MAS technique can produce individual sensitivities for each output variable. This would then allow for the usage of dedicated and different feature subsets for each output.

CRediT authorship contribution statement

Joakim Linja: Methodology, Software, Formal analysis, Investigation, Data curation, Visualization, Validation, Writing – original draft, Writing – review & editing. **Joonas Hämäläinen:** Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **Paavo Nieminen:** Supervision, Validation, Writing – original draft, Writing – review & editing. **Tommi Kärkkäinen:** Software, Conceptualization, Methodology, Supervision, Project administration, Funding acquisition, Validation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported by the Academy of Finland through the projects 315550 (HNP-AI) and 351579 (MLNovCat). We acknowledge grants of computer capacity from the Finnish

Grid and Cloud Infrastructure (**FCCI**; persistent identifier urn:nbn:fi:research-infras-2016072533).

Appendix A. Comparison of feature ranking algorithms

See Table A.4.

Appendix B. MAS-FS comparison to reference methods

See Table B.5.

Appendix C. EMLM-MAS-FS comparison to FNN-MAS-FS

See Tables C.6 and C.7.

Appendix D. Tests with classification datasets

See Table D.8.

References

[1] A.H. de Souza Junior, F. Corona, G.A. Barreto, Y. Miche, A. Lendasse, Minimal Learning Machine: A novel supervised distance-based approach for regression and classification, Neurocomputing 164 (2015) 34–44.
[2] D.P.P. Mesquita, J.P.P. Gomes, A.H. de Souza Junior, Ensemble of efficient minimal learning machines for classification and regression, Neural Process. Lett. 46 (3) (2017) 751–766.
[3] D.P.P. Mesquita, J.P.P. Gomes, A.H. de Souza Junior, J.S. Nobre, Euclidean distance estimation in incomplete datasets, Neurocomputing 248 (2017) 11–18.
[4] T. Kärkkäinen, Extreme minimal learning machine: Ridge regression with distance-based basis, Neurocomputing 342 (2019) 33–48.
[5] E. Pekalska, R.P. Duin, Automatic pattern recognition by similarity representations, Electron. Lett. 37 (3) (2001) 159–160.
[6] Y. Chen, Strategies for similarity-based learning, Ph.D. thesis, University of Washington, Program of Electrical Engineering (2010).
[7] M.J.D. Powell, Radial basis function for multivariable interpolation: a review, in: Algorithms for Approximation, Clarendon Press, Oxford, 1987, pp. 143–167.
[8] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.
[9] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (9) (1990) 1481–1497.
[10] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (2) (1991) 246–257.
[11] Y. Lu, Z. Lai, Y. Xu, X. Li, D. Zhang, C. Yuan, Low-rank preserving projections, IEEE Trans. Cybern. 46 (8) (2016) 1900–1913, https://doi.org/10.1109/TCYB.2015.2457611.
[12] Y. Zhai, Y.-S. Ong, I.W. Tsang, The emerging big dimensionality, IEEE Comput. Intell. Mag. 9 (3) (2014) 14–26, https://doi.org/10.1109/MCI.2014.2326099.
[13] C.K. Fisher, P. Mehta, Bayesian feature selection for high-dimensional linear regression via the Ising approximation with applications to genomics, Bioinformatics 31 (11) (2015) 1754–1761, https://doi.org/10.1093/bioinformatics/btv037.
[14] S. Solorio-Fernández, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, A review of unsupervised feature selection methods, Artif. Intell. Rev. 53 (2020) 907–948, https://doi.org/10.1007/s10462-019-09682-y.
[15] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer, Norwell, MA, 1998.
[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[17] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 121–129.
[18] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[19] H. Zare, M. Niazi, Relevant based structure learning for feature selection, Eng. Appl. Artif. Intell. 55 (2016) 93–102.
[20] X. Wu, X. Xu, J. Liu, H. Wang, B. Hu, F. Nie, Supervised feature selection with orthogonal regression and feature weighting, IEEE Transactions on Neural Networks and Learning Systems.
[21] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[22] Z. Xu, I. King, M.R.-T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Networks 21 (7) (2010) 1033–1047.
[23] K. Benabdeslem, M. Hindawi, Efficient semi-supervised feature selection: Constraint, relevance, and redundancy, IEEE Trans. Knowl. Data Eng. 26 (5) (2014) 1131–1143, https://doi.org/10.1109/TKDE.2013.86.
[24] X. Zhang, Q. Zhang, M. Chen, Y. Sun, X. Qin, H. Li, A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method, Neurocomputing 275 (2018) 2426–2439, https://doi.org/10.1016/j.neucom.2017.11.016.
[25] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1997) 245–271.
[26] J.-X. Peng, S. Ferguson, K. Rafferty, P.D. Kelly, An efficient feature selection method for mobile devices with application to activity recognition, Neurocomputing 74 (2011) 3543–3552.
[27] J.R. Quinlan, Induction of decision trees, Machine learning 1 (1) (1986) 81–106.
[28] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[29] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern Recogn. Lett. 31 (14) (2010) 2225–2236.
[30] M. Wojtas, K. Chen, Feature importance ranking for deep learning, in: Advances in Neural Information Processing Systems (NeurIPS 2020), Vol. 33, 2020, pp. 5105–5114.
[31] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence, XAI, IEEE access 6 (2018) 52138–52160.
[32] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion 58 (2020) 82–115.
[33] N. Burkart, M.F. Huber, A survey on the explainability of supervised machine learning, Journal of Artificial Intelligence Research 70 (2021) 245–317.
[34] Y. Dimopoulos, P. Bourret, S. Lek, Use of some sensitivity criteria for choosing networks with good generalization ability, Neural Process. Lett. 2 (6) (1995) 1–4.
[35] T. Kärkkäinen, Assessment of feature saliency of MLP using analytic sensitivity, in: European symposium on artificial neural networks, computational intelligence and machine learning-ESANN2015. Presses universitaires de Louvain, 2015, pp. 273–278.
[36] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034.
[37] J. Ding, V. Tarokh, Y. Yang, Model selection techniques: An overview, IEEE Signal Process. Mag. 35 (6) (2018) 16–34.
[38] M. Dash, H. Liu, Feature selection for classification, Intelligent data analysis 1 (1–4) (1997) 131–156.
[39] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial intelligence 97 (1–2) (1997) 273–324.
[40] M.J. Grant, A. Booth, A typology of reviews: an analysis of 14 review types and associated methodologies, Health information & libraries journal 26 (2) (2009) 91–108.
[41] M. Kilpala, T. Kärkkäinen, T. Hämäläinen, Differential Privacy: An Umbrella review, Springer Nature (2021) 1–20.
[42] J. Egger, A. Pepe, C. Gsaxner, Y. Jin, J. Li, R. Kern, Deep learning–a first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact, PeerJ Computer Science 7 (2021).
[43] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering 40 (1) (2014) 16–28.
[44] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (3) (2013) 483–519.
[45] J. Miao, L. Niu, A survey on feature selection, Procedia Computer Science 91 (2016) 919–926.
[46] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Computing Surveys (CSUR) 50 (6) (2018) 94.
[47] Y. Li, T. Li, H. Liu, Recent advances in feature selection and its applications, Knowl. Inf. Syst. 53 (3) (2017) 551–577.
[48] N. Spolaôr, M.C. Monard, G. Tsoumakas, H.D. Lee, A systematic review of multi-label feature selection and a new method based on label construction, Neurocomputing 180 (2016) 3–15.
[49] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.
[50] J.C. Ang, A. Mirzal, H. Haron, H.N.A. Hamed, Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection, IEEE/ACM Trans. Comput. Biol. Bioinf. 13 (5) (2016) 971–989.
[51] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Information Fusion 52 (2019) 1–12.
[52] K. Yu, X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, X. Wu, Causality-based feature selection: Methods and evaluations, ACM Computing Surveys (CSUR) 53 (5) (2020) 1–36.
[53] P. Dhal, C. Azad, A comprehensive survey on feature selection in the various fields of machine learning, Applied Intelligence (2021) 1–39.
[54] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186.
[55] N. Dessì, B. Pes, Similarity of feature selection methods: An empirical study across data intensive classification tasks, Expert Syst. Appl. 42 (10) (2015) 4632–4642.
[56] J. Gui, Z. Sun, S. Ji, D. Tao, T. Tan, Feature selection based on structured sparsity: A comprehensive study, IEEE Transactions on Neural Networks and Learning Systems 28 (7) (2017) 1490–1507.
[57] R.J. Urbanowicz, M. Meeker, W. La Cava, R.S. Olson, J.H. Moore, Relief-based feature selection: Introduction and review, J. Biomed. Inform. 85 (2018) 189–203.

[58] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, M. Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, Computational Statistics & Data Analysis 143 (2020).
[59] J. Wang, P. Zhao, S.C. Hoi, R. Jin, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (3) (2014) 698–710.
[60] X. Hu, P. Zhou, P. Li, J. Wang, X. Wu, A survey on online feature selection with streaming features, Frontiers of Computer Science 12 (3) (2018) 479–493.
[61] R. Diao, Q. Shen, Nature inspired feature selection meta-heuristics, Artif. Intell. Rev. 44 (3) (2015) 311–340.
[62] M. Sharma, P. Kaur, A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem, Archives of Computational Methods in Engineering 28 (3).
[63] Q. Al-Tashi, H.M. Rais, S.J. Abdulkadir, S. Mirjalili, H. Alhussian, A review of grey wolf optimizer-based feature selection methods for classification, Evolutionary Machine Learning Techniques (2020) 273–286.
[64] M. Mafarja, A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, Dragonfly algorithm: theory, literature review, and application in feature selection, Nature-Inspired Optimizers (2020) 47–67.
[65] B.H. Nguyen, B. Xue, M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, Swarm and Evolutionary Computation 54 (2020).
[66] Q. Al-Tashi, S.J. Abdulkadir, H.M. Rais, S. Mirjalili, H. Alhussian, Approaches to multi-objective feature selection: A systematic literature review, IEEE Access 8 (2020) 125076–125096.
[67] P.Y. Lee, W.P. Loh, J.F. Chin, Feature selection in multimedia: The state-of-the-art review, Image Vis. Comput. 67 (2017) 29–42.
[68] S. Salcedo-Sanz, L. Cornejo-Bueno, L. Prieto, D. Paredes, R. García-Herrera, Feature selection in machine learning prediction systems for renewable energy applications, Renew. Sustain. Energy Rev. 90 (2018) 728–741.
[69] K. Tadist, S. Najah, N.S. Nikolov, F. Mrabti, A. Zahi, Feature selection methods and genomic big data: a systematic review, Journal of Big Data 6 (1) (2019) 1–24.
[70] D. Pavlyuk, Feature selection and extraction in spatiotemporal traffic forecasting: a systematic literature review, European Transport Research Review 11 (1) (2019) 1–19.
[71] X. Deng, Y. Li, J. Weng, J. Zhang, Feature selection for text classification: A review., Multimedia Tools & Applications 78 (3).
[72] V. Bolón-Canedo, B. Remeseiro, Feature selection in image analysis: a survey, Artif. Intell. Rev. 53 (4) (2020) 2905–2931.
[73] S. Kashef, H. Nezamabadi-pour, B. Nikpour, Multilabel feature selection: A comprehensive review and guiding experiments, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2) (2018).
[74] R.B. Pereira, A. Plastino, B. Zadrozny, L.H. Merschmann, Categorizing feature selection methods for multi-label classification, Artif. Intell. Rev. 49 (1) (2018) 57–78.
[75] P. Raatikainen, J. Hautala, O. Loberg, T. Kärkkäinen, P. Leppänen, P. Nieminen, Detection of developmental dyslexia with machine learning using eye movement data, Array 12 (2021).
[76] M. Cherrington, F. Thabtah, J. Lu, Q. Xu, Feature selection: filter methods performance challenges, in: in: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019, pp. 1–4.
[77] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: 2009 IEEE Symposium on Computational Intelligence and Data Mining, IEEE 2009 (2009) 389–395.
[78] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1) (2006) 489–501, https://doi.org/10.1016/j.neucom.2005.12.126.
[79] J. Hämäläinen, A.S.C. Alencar, T. Kärkkäinen, C.L.C. Mattos, A.H. Souza Júnior, J. P.P. Gomes, Minimal Learning Machine: Theoretical results and clustering-based reference point selection, Journal of Machine Learning Research 21 (2020) 1–29.
[80] T.F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoret. Comput. Sci. 38 (1985) 293–306.
[81] D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis II, An analysis of several heuristics for the traveling salesman problem, SIAM J. Comput. 6 (3) (1977) 563–581.
[82] J. Linja, J. Hämäläinen, P. Nieminen, T. Kärkkäinen, Do randomized algorithms improve the efficiency of minimal learning machine?, Machine Learning and Knowledge Extraction 2 (4) (2020) 533–557, https://doiorg/10.3390/make2040029.
[83] A. Pihlajamäki, J. Hämäläinen, J. Linja, P. Nieminen, S. Malola, T. Kärkkäinen, H. Häkkinen, Monte carlo simulations of au38(sch3)24 nanocluster using distance-based machine learning methods, The Journal of Physical Chemistry A 124 (23) (2020) 4827–4836, https://doi.org/10.1021/acs.jpca.0c01512.
[84] T. Kärkkäinen, Model selection for extreme minimal learning machine using sampling, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, 2019, pp. 391–396.
[85] J.E. Dennis Jr, R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, vol. 16, SIAM, 1996.
[86] T. Kärkkäinen, On the role of Taylor's formula in machine learning, Springer Nature, 2022, Ch. Impact of scientific computing on science and society, (18 pages, to appear).
[87] P.J. Huber, Robust statistics, vol. 523, John Wiley & Sons, 2004.
[88] T. Kärkkäinen, S. Äyrämö, On computation of spatial median for robust data mining, in: Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN, Munich, 2005, p. 14.
[89] G.-B. Huang, What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle, Cognitive Computation 7 (3) (2015) 263–278.
[90] T. Kärkkäinen, MLP in layer-wise form with applications to weight decay, Neural Comput. 14 (6) (2002) 1451–1480.
[91] R.L. Thorndike, Who belongs in the family, Psychometrika 18 (4) (1953) 267–276.
[92] R.C. Yates, A Handbook on Curves and their Properties, JW Edwards, 1947.
[93] V. Satopaa, J. Albrecht, D. Irwin, B. Raghavan, Finding a kneedle in a haystack: Detecting knee points in system behavior, in: 2011 31st International Conference on Distributed Computing Systems Workshops, 2011, pp. 166–171. doi:10.1109/ICDCSW.2011.20.
[94] Y. Sun, J. Yao, S. Goodison, Feature Selection for Nonlinear Regression and its Application to Cancer Research, 2015, pp. 73–81. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611974010.9, doi:10.1137/1.9781611974010.9. URL:https://epubs.siam.org/doi/abs/10.1137/1.9781611974010.9.
[95] D. Dua, C. Graff, UCI machine learning repository (2017). URL:http://archive.ics.uci.edu/ml.
[96] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: treating targets as inputs, Machine Learning 104 (1) (2016) 55–98, https://doi.org/10.1007/s10994-016-5546-z.
[97] University of Toronto, Delve datasets (1996). URL:http://www.cs.toronto.edu/ delve/data/datasets.html.
[98] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[99] A.G. Barnston, Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score, Weather and Forecasting 7 (4) (1992) 699–709, https://doi.org/10.1175/1520-0434(1992)007<0699:CATCRA>2.0.CO;2, URL: https://journals.ametsoc.org/view/journals/wefo/7/4/1520-0434_1992_007_0699_catcra_2_0_co_2.xml.
[100] W.H. Kruskal, A nonparametric test for the several sample problem, Ann. Math. Stat. 23 (4) (1952) 525–540, URL: http://www.jstor.org/stable/2236578.
[101] P. Teisseyre, Feature ranking for multi-label classification using Markov networks, Neurocomputing 205 (2016) 439–454, https://doi.org/10.1016/j.neucom.2016.04.023.
[102] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015). URL: https://www.tensorflow.org/.
[103] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.
[104] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nat. Methods 17 (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.
[105] J. Hämäläinen, T. Kärkkäinen, Problem transformation methods with distance-based learning for multi-target regression, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, 2020, pp. 691–696.

Joakim Linja received his M.Sc degree from the Department of Physics in University of Jyväskylä, Finland. He is currently working as a PhD researcher at the Faculty of Information Technology in University of Jyväskylä, Finland. His main fields of interests are machine learning, computational science and nanophysics.

Joonas Hämäläinen received the B.S. degree in physics, the M.S. degree in applied physics, and the Ph.D. degree in mathematical information technology from University of Jyväskylä, Jyväskylä, Finland, in 2012, 2013, and 2018. He is currently working as a Postdoctoral Researcher at the Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland. His main research interests include data mining and machine learning.

Paavo Nieminen, PhD, is currently working as a Senior Lecturer at the Faculty of Information Technology, University of Jyväskylä, where he has received also his own academic education and degrees. Alongside teaching and curriculum work, he collaborates in research and supervision on topics of machine learning and computer science education.

Tommi Kärkkäinen (TK) received the Ph.D. degree in Mathematical Information Technology from the University of Jyväskylä (JYU), in 1995. Since 2002 he has been serving as a full professor of Mathematical Information Technology at the Faculty of Information Technology (FIT), JYU. TK has led 50 different R&D projects and has been supervising 60 PhD students. He has published over 190 peer-reviewed articles. TK received the Innovation Prize of JYU in 2010. He has served in many administrative positions at FIT and JYU, leading currently a Research Division and a Research Group on Human and Machine based Intelligence in Learning. The main research interests include data mining, machine learning, learning analytics, and nanotechnology. He is a senior member of the IEEE.