Dynamic k determination in k-NN classifier: A literature review

Merkourios Papanikolaou
Dept. of Applied Informatics
School of Information Sciences
University of Macedonia
Thessaloniki, Greece
aid20006@uom.edu.gr

Georgios Evangelidis
Dept. of Applied Informatics
School of Information Sciences
University of Macedonia
Thessaloniki, Greece
gevan@uom.edu.gr

Stefanos Ougiaroglou
Dept. of Information and Electronic Engineering
School of Engineering
International Hellenic University
Thessaloniki, Greece
stoug@ihu.gr

***Abstract*—One of the widely used classification algorithms is k-Nearest Neighbours (k-NN). Its popularity is mainly due to its simplicity, effectiveness, ease of implementation and ability to add new data in the training set at any time. However, one of its main drawbacks is the fact that its performance is highly dependent on the proper selection of parameter k, i.e. the number of nearest neighbours that the algorithm examines. The most frequently used technique for the "best" k determination is the cross validation as there is no general rule for choosing the k value due to its dependency on the training dataset. However, selecting a fixed k value throughout the dataset does not take into account its special features, like data distribution, class separation, imbalanced classes, sparse and dense neighborhoods and noisy subspaces. A lot of research has been done to date in the specific field, leading to many k-NN variations. In the present research, a thorough literature review is conducted in order to summarize all the achievements made to date in this field. Specifically, a pool of twenty eight (28) approaches with their experimental results are presented, all concerning methods and techniques for dynamic "best" k selection.**

***Index Terms*—k-NN Classification, Dynamic k parameter determination, Adaptive k parameter determination**

I. Introduction

Classification algorithms or classifiers attempt to recognize, understand and group objects or data instances into preset categories. Using a set of instances - training/test dataset - for which the category or class label is known, they attempt to build/train a model that can successfully predict the class label of unlabelled instances. Generally, classifiers are divided into two main categories; eager and lazy ones [5]. Eager learners build a generalized model based on the training instances that is used for making predictions. On the other hand, lazy classifiers usually store in memory all the instances and conduct a local search. In other words, the instances are both the training dataset and the model at the same time.

The k-Nearest Neighbours (k-NN) algorithm [6] is an effective and extensively used lazy classifier. Its popularity is due to (i) its simplicity, as there is no model to train, (ii) its effectiveness, as the asymptotic classification error of 1-NN is bounded by twice the Bayesian error rate, (iii) its ease of implementation, as it is mainly based on distance computations, and, (iv) its ability to add new training data at any time as there is not an already trained model, an extremely useful characteristic, especially, for data streams. Specifically, k-NN predicts the class label of an unknown instance by conducting a local search among its k nearest neighbours and then applying a majority voting; the unknown instance is labelled with the class of the majority of the k nearest neighbours. Usually, local search is based on Euclidean distance.

Based on the above, one can easily conclude that a major drawback of lazy learners is the high demand of both storage space, as they store all instances in memory, and computational resources, especially in case of high dimensional datasets. An additional drawback of k-NN is that performance is dependent on value k, which determines the extent of the neighborhood that the search is taking place.

In recent years, a lot of research have been conducted in order to tackle the above-mentioned disadvantages. In practice, k is treated as a hyper-parameter and, consequently, the most frequently used technique for the "best" k determination is the cross validation as there is no general rule for choosing the k value due to its dependency on the training dataset. However, the k that is finally determined with the use of cross validation is a unique and fixed value for the whole dataset without taking into account the specific and unique features that each dataset may have as well as its distribution. For example, in cases either of classes that are not well separated or of noisy instances, a large k value may be more suitable in order to examine an extensive subspace (neighborhood). A large k value results in a noise tolerant classifier, as its search area is large. On the contrary, in case of distinct classes, a large k value may result both in higher computational cost and in accuracy deterioration. In such cases, a small k value may be more appropriate. A small k value results in a noise sensitive classifier, as the search area is limited.

However, the problem becomes more and more complex in cases of real-life datasets that may simultaneously contain both well and not-well separated classes, imbalanced classes, sparse and dense neighborhoods and noisy sub-spaces. Based on the above, one should consider that a globally defined fixed k value is not appropriate for a dataset. Instead, one should take into account the special features of the dataset

978-1-6654-0032-9/21/$31.00 ©2021 IEEE

and the subspace that each instance lies into and try to dynamically determine a local k value for each instance to be classified. Thus, the research has led to various approaches of k-NN classifier, which mainly combine it with various other techniques for k value determination. The purpose of the present work is to conduct a thorough literature review, gathering all the existing alternative approaches concerning both the global and the local determination of the "best" k value.

The rest of the paper is organized as follows: Section II briefly describes the methodology followed in the literature review research, while Sections III and IV briefly present the alternative approaches found in literature. Finally, Section V discusses the findings and concludes to useful remarks.

II. Methodology

The main research objective of this paper was to gather together all the proposed approaches for dynamic k value determination when applying k-NN classifier. The adopted research methodology consisted of the following steps. The first step included a thorough literature search for potential references in academic bibliographic databases and search engines that are connected with scientific publications (e.g., Springer Link, Mendeley, Science Direct, Google Scholar, Scopus, Wiley Online, Emerald Insight, etc.). The second step included the one by one examination of the references included in the documents collected in step one. The final step involved the in depth study of the most relevant with the research objective references.

The successful execution of the first two steps led to the collection of about sixty five (65) publications. The in depth study of these sources (step three) led to a pool of twenty eight (28) publications that either propose a new methodological framework for dynamic k determination or refer to the need of a more adaptive k value by proposing, for example, alternatives where the classification process is independent of k. Seven out of twenty eight publications refer to global k value (hyper-parameter tuning) while the rest twenty one two examine it locally (i.e., for each test instance).

Then, the publications extracted from step three are grouped according to the methodological framework they are based on. In total, six (6) groups were identified, namely mapping to genetic algorithms, neural networks, prototypes or clusterings, heuristics, probabilistic and the group "other", which includes any other approach that does not fit into the previous categories. The study was conducted between December 2020 and February 2021.

III. Global Approaches

In [19], the authors propose an approach that is based on a genetic algorithm, namely Biogeography based optimization (BBO). This algorithm uses a multi-part chromosome for the simultaneous optimization of feature selection, feature weighting and the k value. The proposed algorithm was compared with six evolutionary and fourteen non-evolutionary genetic algorithms, on 10 different datasets. After conducting experiments, the BBO seems to outperform all the compared algorithms, in terms of accuracy rate, Kohen's Kappa and reduction rate, across all datasets.

A heuristic based k-NN variation is presented in [9], where the k value is selected automatically, without any user intervention. The heuristic is based on the idea that the algorithm will search for that k value that correctly classifies the majority of training instances. The proposed approach was compared, in terms of accuracy, to conventional k-NN, with k $\in [1, 51]$ (only odd numbers) on 25 datasets. In thirteen out of twenty cases, the proposed algorithm outperformed the widely used 1-NN. Moreover, in five out of the thirteen above mentioned cases, the difference was statistically significant.

Despite the fact that the approach presented in [21] does not propose a different procedure for k selection, the described heuristic makes the performance of conventional k-NN less dependent from the selected k value. Briefly, the idea behind this approach is that (for a given k) if x votes for y, then y also votes for x, even if x does not belong to $y's$ k nearest neighbours. The proposed algorithm was compared, in terms of accuracy, to 1-NN and 1-tNN, over 29 datasets. The results demonstrated that the proposed algorithm outperformed the rest ones without statistical significance.

The authors in [16] presented an analytic probabilistic k-NN variation which deals with the uncertainty on k using a prior distribution on it. The proposed approach was compared, in terms of classification error rate on a variety of datasets, to conventional k-NN algorithm, demonstrating competitive performance.

The approach presented in [14] does not propose a varied procedure for k selection but a faster cross-validation technique in order to examine a larger amount of k values within the same running time, reducing the time complexity by O($K*$), where $K*$ is the maximum k value. The proposed technique was tested on 3 datasets demonstrating its contribution in running time reduction.

A quite older empirical approach is presented in [8]. The authors argue that the optimal k value is dependent on the dimensions, the size and the structure of the sample size. Moreover, they propose some equations for k computation in function with the difference between sample proportions and the difference between con-variance matrices. No experimental tests are reported.

A general discussion about the k selection in problems with binary imbalanced class is made in [15]. In this paper the authors cite some proposed equations for k approximation, like $k \approx n^{\frac{2}{8}}$ or $k \approx n^{\frac{3}{8}}$.

IV. Local Approaches

A. Genetic Algorithms Approaches

An approach utilizing genetic algorithms is presented in [17]. In this work, authors propose a genetic alternative of conventional k-NN, namely G-k-NN, which is used for the optimization of the k value. Specifically, this technique builds decision trees that split the search space into distinct regions. Then, a fixed, unique and optimized k value is assigned to each region. In turn, this k value is assigned to each unlabelled instance (test instance), according to the position it lies in input space and the region it belongs to. The search for the k nearest neighbours can be performed either locally - within the limits of the boundaries of each region - or globally - across all regions - or in a mixed way - both globally and locally. The three alternatives of the proposed algorithm - local G-k-NN, global G-k-NN and mixed G-k-NN - were compared, in terms of accuracy, to the classic k-NN for $k = 5$, $k = 11$ and $k = 11$, on 27 different datasets. As far as the statistical significance is concerned, global G-k-NN significantly outperforms the three versions of conventional k-NN. However, this is not the case for local G-k-NN.

B. Neural Networks Approaches

[20] presents two alternative versions of classic k-NN; the Adaptive k-NN (Ada-k-NN) and the the Adaptive k-NN2 (Ada-k-NN2). Both methods utilize the data density and distribution in order to find an appropriate k value for each unlabelled instance. On the one hand, Ada-k-NN uses artificial neural networks in order to learn this suitable k. On the other hand, Ada-k-NN2 is a simplified version in the light of using a heuristic as indicator of the local density around the unlabelled instance. The heuristic is based on the idea of searching for that k that correctly classifies the majority of test instance's neighbors. Two sets of experiments were conducted. The first set was oriented in comparing the algorithms with eight other classifiers on 17 datasets. The datasets were divided into two categories; the small/medium ones and the large ones. The second set was oriented in comparing the proposed algorithms in the presence of imbalanced classes. As far as the first set, the results revealed that Ada-k-NN2 outperformed the other classifiers in terms of average accuracy. It, also, achieved the best Average Rank for Small and Medium-Scale datasets. Respectively, Ada-k-NN attained the fifth place. In case of large datasets, Ada-k-NN2 attained the second place in terms of average accuracy and the first place in terms of Average Rank for Large-Scale dataset. On the contrary, Ada-k-NN attained the sixth and eighth place, respectively, which shows that it suffers from scalability. As far as the second set is concerned, Ada-k-NN2 in combinations with GIHS - a simple weighting scheme - outperformed the rest classifiers.

C. Prototypes and Clustering based Approaches

An interesting approach is presented in [10] where the search space is divided into sub-spaces. Each sub-space constitutes a neighbour which, in turn, is represented by a prototype. Each prototype is assigned with a k value, which is considered to be the optimum within the neighbour borders. A greedy approach is adopted in order to find these optimum k values; for each prototype the local performance is tested, with k varying within an interval [kmin, kmax], with most common values k_{min} = 1 and k_{max} = 100. Eventually, the k value with the highest performance is assigned to the corresponding prototype. Then, the classification process for each unlabelled (test) instance is conducted using the k value of its nearest prototype (training) instance. Given that the the proposed approach can be adopted in almost every k-NN variation, the authors conducted experiments using the standard k-NN, the adaptive k-NN and the symmetrical k-NN in combination with the proposed algorithm for k selection versus the same k-NN variations in combination with ten fold cross validation for k selection. In both cases, the interval was set to k_{min} = 1 and k_{max} = 100. The experiments were conducted using 80 datasets for regular problems and 65 datasets for imbalanced problems. The algorithms were compared in terms of accuracy and Kohen's Kappa (for regular datasets) and G-mean and auROC (for imbalanced datasets). The experiments results showed that the proposed algorithm performed better both for regular and imbalanced datasets.

An approach, based on clustering, is demonstrated in [5], namely the One Nearest Cluster (1NC) approach. According to this approach, the user pre-defines the number of clusters (l) and picks the M closest samples around the unlabelled instance. All the distances among the test instance and the M closest samples are calculated and, according to the distance, all the M samples are laid on an one-dimensional axis, from the closest to the farthest. Then, clustering is applied in order to split the M samples into l clusters. Finally, the majority voting technique is applied within the closest cluster. That way, the approximation of k can be expressed as $k \approx \frac{M}{l}$. This procedure is repeated for every test instance. The proposed algorithm (with $k \approx \frac{15}{3}$) was compared with 1-NN and 5-NN, in terms of accuracy, on 36 datasets. Moreover, a T-test is performed in order to check statistically the pairs 1NC - 1-NN and 1NC - 5-NN. 1NC outperformed nine times, 1-NN ten times and 5-NN seventeen times. In terms of averaged accuracy over the 36 datasets, 1NC demonstrated the highest accuracy. The statistical test showed that 1NC outperforms 1-NN significantly. The authors argue that M and l should have small integer numbers because this leads not only to reduced computational time but also to the same results in comparison with larger values (e.g. $\frac{10}{2}$ gives the same outcome with $\frac{100}{20}$).

A k-NN variation is presented in [13]. The fundamental idea behind this approach is that some representative instances will represent the whole training set. After the selection of representatives, a procedure based on the training set's local features which is beyond the scope of this paper, the k value is chosen automatically - without user intervention - for each representative; k value equals to the number of instances covered by each representative. In other words, the local search for applying the majority voting technique, is conducted within the limits of the local neighborhood of each representative instance. The proposed algorithm was compared, in terms of accuracy, with C5.0 and k-NN over 6 datasets. Due to the fact that its average classification accuracy was the highest one ($85,15\%$) comparing to C5.0 ($81,35\%$) and k-NN ($80,90\%$ for $k = 1$, $83,52\%$ for $k = 3$ and $83,67\%$ for $k = 5$), its performance is considered satisfactory.

The concept of representing the whole training set with a few representative instances (prototypes) is farther exploited in [22]. Specifically, in this approach, namely Subspace

Homogeneity based Dynamic k-NN classifier (shd-k-NN), the authors apply repetitively the k-means clustering procedure until all created clusters are homogeneous. This repetitive procedure produces a kind of tree of clusters. Each leaf node represents a homogeneous cluster whose depth (d) provides information about the region where the unclassified instance lies. When an unlabelled instance needs to be classified, the algorithm finds the nearest homogeneous cluster's centroid and its corresponding d value. Then, the authors suggest five heuristics based on which the k is calculated as a function of d and apply the majority voting technique. Based on the above, it is easily concluded that this k-NN variation is independent of "best" or "optimum" k selection as it is dynamically selected for each instance in an automated way. The proposed algorithm (using all different heuristics) was compared to some common used k-NN variations; "best" k-NN (k was estimated using 5-fold cross validation), 1-NN, 5-NN, 10-NN, $\sqrt{N}$-NN and $\sqrt{\frac{N}{2}}$-NN, where N represents the number of instances included in training set. The comparison was made in terms of accuracy on 14 original datasets and 19 variations of them with random "noise" addition. Generally speaking, all heuristics demonstrate not only competitive performance to the "best" k-NN but also outperform it in some cases. Comparing the shd-k-NN to conventional k-NN (with fixed k), the experiments showed that the proposed algorithm's performance is better in almost all cases.

D. Heuristic-based approaches

The exploitation of heuristics is a quite common technique for conventional k-NN improvement. One such case is presented in [23], where authors apply incremental computation of nearest neighbors in R-Trees. However, the nearest neighbor search breaks if some criteria induced by a heuristic are satisfied. Thus, each unlabelled instance is classified by a non-fixed k. The most important property of the incremental search is that the neighbors are discovered in their order of their distance from the query instance. This allows the discovery of the $(k+1)$-th nearest neighbor if we have already discovered the previous k nearest neighbors. Three different heuristics for k-NN early break, independently of the selected k value. This way, conventional k-NN preserves its simple concept and implementation while, simultaneously, reduces the required computational cost. Briefly, the three proposed heuristics for k-NN early break are the following: (a) The majority voting technique breaks when a predefined threshold for the dominant class is met; (b) The majority voting technique breaks when the remaining votes are less than the difference k-(current votes for the majority class). For example, it's pointless to continue searching in case of a binary classification problem, with $k = 9$ and after searching 7 neighbours, the dominant class has already 5 votes; (c) The majority voting technique breaks when a predefined number of consecutive neighbours is met which all vote for the dominant class. The aforementioned heuristics were tested on 2 datasets in terms of accuracy and computational cost. As far as the accuracy is concerned, the second heuristic outperforms the rest. As far as the the computational cost is concerned, the first heuristic outperforms the rest.

Another similar approach, using a heuristic, is presented in [25]. According to this, a different k value is assigned to each instance of the training set. This k value is equal to the minimum number of neighbours needed to be included in the majority voting procedure in order the instance in question be classified correctly. Then, for each unlabelled instance, the proposed algorithm finds its nearest neighbour and its assigned k value. The k-NN algorithm is the applied with this k value. The above mentioned approach was compared, in terms of accuracy, to conventional k-NN, with k $\in [1, 9]$ and Ada-k-NN [20] on 15 datasets. Its performance is considered competitive as it is classified second on six datasets and third on one dataset. Moreover, the authors support that the Ada-k-NN overall outperforms the rest conventional k-NN tested algorithms.

The authors in [2] have developed a k-NN variation, oriented in text categorization. The proposed algorithm does not suggest a new procedure for k selection but develops a k-NN variation less dependent on the k value. Specifically, the authors suggest that the k should be proportional to the number of training instances that belong to the category the test instance is going to be classified. In order to classify a test instance in a category with more training points, the algorithm should use a larger k value in comparison with classifying the same test instance in another category with less training points. The experiments conducted on 2 different text datasets revealed that the proposed variation is less sensitive to k selection. As a consequence, it can effectively deal with class imbalance.

E. Probabilistic Approaches

The concept of "spheres of confidence" is presented in [18] that exploits the notion of Laplace estimator $P_{ClassA} = \frac{k+1}{N+C}$, where, k is the number of training instance belonging to class A, N is the total number of training instances belonging within the sphere, and, C is the total number of classes. The cornerstone of this approach is that each training instance is surrounded by a sphere of confidence that includes other training instances. In order to construct the sphere of confidence, the algorithm searches the nearest neighbours one by one, starting from the closest one. The procedure breaks either when the Laplace estimator's value starts decreasing or when all training instances are examined. The first case is called "eager construction" and the generated sphere of confidence includes all the points examined by the algorithm until the break. The second case is called "total construction" and as sphere of confidence is selected that with the highest Laplace estimator value. When an unlabelled instance needs to be classified, the algorithm applies the majority voting technique either among all instances within the spheres covering the test instance (instance aggregation) or among all spheres covering the test instance (sphere aggregation), considering each sphere as one class (the class of the majority instances within the sphere). The above mentioned approach was compared, in terms of accuracy and auROC, to the conventional k-NN with fixed values $k = 5, 11, 17$ on 18 datasets. Briefly, the results showed that the proposed algorithm significantly outperformed the rest ones, demonstrating better performance on thirteen out of eighteen datasets.

The concept of exploiting the local data distribution is also utilized by the approach presented in [3] and in [4]. The authors suggest the construction of a hypersphere around each unlabelled point in order to capture the distribution of the training instances around it. Moreover, they introduce the concept of hubness weight, i.e., the probability a point belongs to the specific test point's neighbourhood. Combining the above mentioned notions, a unique k value is assigned to each unlabelled point. As referred in [3], the proposed algorithm was compared in terms of accuracy, both to the conventional k-NN with fixed values $k = 1, 3, 5, 7, [\sqrt{N}]$ and to other k-NN variations like these presented in [18] and in [26]. The comparison was made on 15 datasets. The results showed that the proposed algorithm mainly outperformed the competitors and thus can be considered as an effective k-NN variation.

In the research presented in [29] the authors suggest that the neighbourhood size should be versatile in order to handle the imbalanced classification problem. Thus, for a given k, they propose that the examined neighbourhood should be increased until $\frac{k}{2}$ instances of the rare class are included. The above mentioned approach was compared, in terms of auROC and Convex Hull analysis, to some others k-NN variations (ENN and CCW-k-NN) and techniques for handling imbalanced datasets (SMOTE re-sampling and MetaCost) on 12 datasets. Briefly, the experiments showed that the proposed approach demonstrated quite competitive performance as it outperformed both the k-NN variations and the widely used techniques for handling the imbalanced classification problem. A general discussion about the k selection in problems with binary imbalanced class is made in [15]. In this paper the authors cited some proposed equations for k approximation, like $k \approx n^{\frac{2}{8}}$ or $k \approx n^{\frac{3}{8}}$.

The authors in [24] argue that parameter k is not sufficient enough to decide the neighbourhood's size. Thus, they introduce a new parameter, namely I, which is measuring the number of informative instances within the selected neighbourhood. A training point is considered to be informative if it is close to the test instance and far enough from instances from other classes. According to the same authors, k and I selection is made using cross validation techniques. The proposed approach was compared, in terms of error rate, to some k-NN variations and other popular classifiers, like Support Vector Machines (SVM). The comparison was made on a variety of datasets, including text categorization and object recognition. The results showed that the proposed algorithm is less sensitive to k selection (comparing to conventional k-NN algorithm) while it demonstrated competitive performance compared to widely used classifiers.

Another probabilistic approach is presented in [12]. According to this approach, a unique k value is selected for each unlabelled instance based on the class distribution around it. The proposed algorithm demonstrated remarkable performance after experimenting on several datasets, in terms of standard error, in comparison with some k-NN variations. Similarly, the same author in [11] proposed a Bayesian approach in order to optimize the k selection based on data distribution. After conducting experiments, the proposed procedure seemed to outperformed the commonly used cross validation techniques.

The concept of exploiting the local statistical confidence for deciding the neighborhood size is presented in [27]. Specifically, the idea behind this approach is that instead of choosing a fixed k (like in conventional k-NN), the k is tuned, utilizing the local data distribution, until a predefined level of confidence is met. The algorithm was tested on 5 datasets. The comparison was made in terms of classification error rate with the conventional k-NN rule. After conducting experiments, the conventional k-NN demonstrated slightly better performance (lower error rates) than the proposed approach.

Finally, the authors in [7] formed expressions in order to explain statistically the k-NN classifier's behavior. These expressions can be used in order to deal with the problem of k selection. In fact, the authors argue that it is preferable to built multiple distributions, based on the above mentioned expressions, for the unlabelled instances and then select a k that decreases the error rate in as many distributions as possible. This way, the noise within the dataset can be taken into account.

F. Other Approaches

An approach for dynamic k estimation is presented in [30]. Briefly, according to this approach, an iterative algorithm outputs an interval $[k_{min}, k_{max}]$, where k is searched dynamically. Then, the algorithm creates variation tendency curves, according to the proportion of correctly classified instances for each k value within the aforementioned interval. The final k value is selected based on three criteria that are dependent on the shape of the variation tendency curves. The proposed algorithm is compared, in terms of precision, recall and F-score, to the conventional k-NN with fixed values $k = 1, 5, 7, \sqrt{N}$, where N =sample size, on a Facebook dataset. The approach in question outperformed the rest ones, in terms of recall and F-score, while it had the second best performance in terms of precision.

Another approach where k is selected for each data point is presented in [1]. Adopting a weighting scheme, the authors tried to minimize the distance between the generated prediction and the ground truth, optimizing at the same time the number of the selected nearest neighbours for each unique unlabelled instance. The proposed algorithm was compared, in terms of standard error metric, to the conventional k-NN and the Nadaraya-Watson estimator on 8 datasets. According to the results, the proposed approach outperformed the rest ones on seven out of eight datasets while on three out of seven datasets, the outperformance was statistically significant.

The authors in [28] presents an analytic optimization framework, namely Graph Sparse k-NN (GS-k-NN), in order to learn a different (optimum) k value for each unlabelled (test) instance, utilizing the already known data distribution. The proposed approach was compared, in terms of accuracy and root mean square error (RMSE), to conventional k-NN (with fixed $k = 5$) and another k-NN variation (L-k-NN) on 12 datasets. The experiments conducted for classification, regression and missing values imputation. In summary, the GS-k-NN is considered by the authors a quite competitive approach as it outperformed the rest algorithms, demonstrating higher accuracy and lower RMSE.

V. Discussion and Conclusions

Based on Table I, the aforementioned approaches cover the time period from 1986 till 2020 (with median value 2009). An interesting observation is that the last three years (from 2018 till 2020) only two publications have been recorded.

The majority of the approaches uses probabilistic frameworks in order to dynamically select the proper k value. However, these approaches are mostly the older ones, with median year of publication 2007. As far as the most recently published works (within the last five years), the majority of them utilize the prototype and clustering technique (three out of six publications), following the "other" (two out of six publications) and neural networks (one out of six publications) technique.

As far as the level of analysis for k value selection, either the search space is divided in distinct subspaces (level of analysis: the specific region or prototype), each of which is assigned with a unique k value (and, consequently, each unlabelled instance is assigned with the k value of the subspace it belongs to) or each test instance is assigned with a unique k value (level of analysis: the test instance itself).

The median amount of datasets, used for testing the aforementioned techniques, is twelve (12), something that is considered insufficient to reliably evaluate the performance. We support that one should use a wide variety of datasets, like datasets containing noise, imbalaced datasets, real life datasets, etc., in order to evaluate in a holistic manner each approach's performance. In any case, this should be supplemented with the appropriate statistical tests in order to ascertain if performance superiority is significant. Unfortunately, only in twelve out of twenty eight cases, at least one statistical test was conducted.

The last two columns of the summarizing table deals with the number of citations received by each paper in an attempt to identify the most influential papers. In order to achieve this, it is considered appropriate not to just report the total number of citations each paper has received so far, but, also to compute a normalized index, i.e., the average citations per year. All the needed information was retrieved from Google Scholar in June 2021. The median number of citations is 36.5 while the median number of the average citations per year is 4.035. Nine (9) out of twenty eight (28) papers have received more than one hundred citations, while six (6) out of twenty eight (28) papers have more than ten (10) citations per year (on average). The three most influential papers - both in terms of the total number of citations and the average citations per year - are [13] (621 / 34.5), [27] (322 / 23) and [24] (270 / 19.28). Unfortunately, one should note that none of the above-mentioned papers conduct statistical tests in order to ascertain the performance superiority, especially, in combination with the insufficient number of datasets used for testing (6, 5 and 12).

The authors of this paper feel the need to distinguish [10] as a complete approach, which could be adopted in almost every k-NN variation. The proposed algorithm, a prototype based variation, was tested using 80 datasets for regular and 65 for imbalanced problems in terms of three different metrics (Kohen's Kappa, G-mean and auROC). The algorithm's outperformance was significant, based on Wilcoxon tests that were conducted for all different situations (regular and imbalanced datasets) and metrics. In light of all the above, the specific approach have received seventy-one (71) citations within only four (4) years (17.75 citations per year).

All the above makes it clear that a wide variety of approaches have been proposed in the literature, in an effort to build an even more effective k-NN algorithm as it is now evident that a fixed k value is not efficient enough to lead to optimal performance. This is due to the special features that every sub-space of each dataset has, e.g., each subspace within a dataset does not have same density or noise.

However, in our opinion, researchers who try to overcome one of the main k-NN disadvantages, namely, the dependency on the k selection, by creating variations of the standard algorithm, should have in mind not to alleviate its advantages, which were mentioned in the first section. In other words, one should strike a balance between achieving higher classification rate (due to a "better" k selection) and keeping the algorithm relatively simple.

This paper aimed at gathering together not only the state-of-art publications, which are mostly cited by the majority of recent publications, but also whatever exists in the literature concerning the research field in question. This way, researchers will have the opportunity to form a clear picture of the achievements made to date, probably inspiring their future efforts. he majority of trends, found and presented in the present work, referred to the exploitation of heuristics, prototypes and clusterings and probabilistic frameworks in order to dynamically estimate the appropriate k-values for each sub-space of each dataset, in contrast to the prevailing (in practice) technique of choosing a fixed "best" k using cross-validation. In most cases, and after conducting experiments on standard benchmarking datasets, the developed variations outperformed the conventional k-NN algorithm, with a fixed k value throughout the dataset.

VI. Acknowledgement

This paper is a result of research conducted within the "MSc in Artificial Intelligence and Data Analytics" of the Department of Applied Informatics of the University of Macedonia. The publication/presentation of the paper is funded by the University of Macedonia Research Committee.

TABLE I
SUMMARY TABLE OF PROPOSED APPROACHES

paper	year	approach	level of k selection	#datasets	statistical tests	#citations	average citations per year
[19]	2013	Genetic Algorithm	Global	10	No	9	1.12
[17]	2008	Genetic Algorithm	Region	27	Yes	3	0.23
[20]	2018	Neural Networks/Heuristic	Region	17	Yes	39	13
[10]	2017	Prototypes & Clustering	Prototype	145	Yes	71	17.75
[5]	2017	Prototypes & Clustering	Prototype	36	Yes	26	6.5
[13]	2003	Prototypes & Clustering	Prototype	6	No	**621**	**34.5**
[22]	2020	Prototypes & Clustering	Prototype	33	No	0	0
[9]	2001	Heuristics	Global	25	Yes	5	0.25
[21]	2003	Heuristics	Global	29	Yes	35	1.94
[25]	2010	Heuristics	Test Instance	15	No	103	9.36
[2]	2004	Heuristics	Test Instance	2	No	142	8.35
[23]	2007	Heuristics	Test Instance	2	No	57	4.07
[16]	2002	Probabilistic	Global	6	No	172	9.05
[18]	2008	Probabilistic	Test Instance	18	Yes	6	0.46
[3]	2014	Probabilistic	Test Instance	15	Yes	18	2.57
[4]	2015	Probabilistic	Test Instance	15	No	9	1.5
[29]	2013	Probabilistic	Test Instance	12	Yes	28	3.5
[24]	2007	Probabilistic	Test Instance	12	No	270	19.28
[7]	2013	Probabilistic	Test Instance	5	No	38	1.63
[12]	2007	Probabilistic	Region	14	Yes	31	2.21
[11]	2006	Probabilistic	Region	11	Yes	123	8.2
[27]	2007	Probabilistic	Region	5	No	322	23
[14]	2012	Other	Global	3	No	9	1
[8]	1986	Other	Global	0	No	120	4.8
[15]	2003	Other	Global	0	No	100	5.55
[1]	2016	Other	Test Instance	8	Yes	50	10
[28]	2014	Other	Test Instance	12	No	19	2.71
[30]	2017	Other	Region	1	No	6	4

REFERENCES

[1] Oren Anava and Kfir Y. Levy. k-Nearest neighbors: From global to local. *Advances in Neural Information Processing Systems*, pages 4923–4931, 2016.

[2] Li Baoli, Lu Qin, and Yu Shiwen. An adaptive k -nearest neighbor text categorization strategy. *ACM Transactions on Asian Language Information Processing*, 3(4):215–226, dec 2004.

[3] Gautam Bhattacharya, Koushik Ghosh, and Ananda S. Chowdhury. Test Point Specific k Estimation for kNN Classifier. In *2014 22nd International Conference on Pattern Recognition*, pages 1478–1483. IEEE, aug 2014.

[4] Gautam Bhattacharya, Koushik Ghosh, and Ananda S. Chowdhury. A probabilistic framework for dynamic k estimation in kNN classifiers with certainty factor. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–5. IEEE, jan 2015.

[5] Faruk Bulut and Mehmet Fatih Amasyali. Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster. *Pattern Analysis and Applications*, 20(2):415–425, may 2017.

[6] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, jan 1967.

[7] Amit Dhurandhar and Alin Dobra. Probabilistic characterization of nearest neighbor classifier. *International Journal of Machine Learning and Cybernetics*, 4(4):259–272, aug 2013.

[8] Gregory G. Enas and Sung C. Choi. Choice of the Smoothing Parameter and Efficiency of k-Nearest Neighbor Classification. In *Statistical Methods of Discrimination and Classification*, volume 12, pages 235–244. Elsevier, 1986.

[9] Francisco J. Ferrer-Troyano, Jesús S. Aguilar-Ruiz, and José C. Riquelme. Non-parametric Nearest Neighbor with Local Adaptation. In P. Brazdil and A. Jorge, editors, *Progress in Artificial Intelligence. EPIA 2001. LNCS*, volume 2258, pages 22–29. Springer, Berlin, Heidelberg, 2001.

[10] Nicolas Garcia-Pedrajas, Juan A. Romero del Castillo, and Gonzalo Cerruela-Garcia. A Proposal for Local k Values for k -Nearest Neighbor Rule. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):470–475, feb 2017.

[11] Anil K. Ghosh. On optimum choice of k in nearest neighbor classification. *Computational Statistics and Data Analysis*, 50(11):3113–3123, jul 2006.

[12] Anil K. Ghosh. On Nearest Neighbor Classification Using Adaptive Choice of k. *Journal of Computational and Graphical Statistics*,

16(2):482–502, jun 2007.

[13] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. KNN Model-Based Approach in Classification. In R. Meersman, Z. Tar, and D.C. Schmidt, editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888, pages 986–996. Springer, 2003.

[14] Greg Hamerly and Greg Speegle. Efficient Model Selection for Large-Scale Nearest-Neighbor Data Mining. In L.M MacKinnon, editor, *Data Security and Security Data. BNCOD 2010. Lecture Notes in Computer Science*, volume 6121, pages 37–54. Springer, Berlin, Heidelberg, Springer, Berlin, Heidelberg, 2012.

[15] David J. Hand and Veronica Vinciotti. Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern Recognition Letters*, 24(9-10):1555–1562, jun 2003.

[16] C. C. Holmes and N. M. Adams. A probabilistic nearest neighbour method for statistical pattern recognition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):295–306, may 2002.

[17] U. Johansson, R. König, and L. Niklasson. Evolving a locally optimized instance based learner. In *Proceedings of the 2008 International Conference on Data Mining, DMIN 2008*, pages 124–129, 2008.

[18] Ulf Johansson, Henrik Boström, and Rikard König. Extending nearest neighbor classification with spheres of confidence. *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21*, (Feb 2014):282–287, 2008.

[19] Ahmad A. Kardan, Atena Kavian, and Amir Esmaeili. Simultaneous feature selection and feature weighting with K selection for KNN classification using BBO algorithm. In *The 5th Conference on Information and Knowledge Technology*, pages 349–354. IEEE, may 2013.

[20] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Adaptive Learning-Based k-Nearest Neighbor Classifiers With Resilience to Class Imbalance. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):1–13, 2018.

[21] Richard Nock, Marc Sebban, and Didier Bernard. A Simple Locally Adaptive Nearest Neighbor Rule With Application To Pollution Forecasting. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(08):1369–1382, dec 2003.

[22] Stefanos Ougiaroglou, Georgios Evangelidis, and Konstantinos I. Diamantaras. Dynamic k-NN Classification Based on Region Homogeneity. In J. Darmont, B. Novikov, and R. Wrembel, editors, *Communications in Computer and Information Science*, volume 1259 CCIS, pages 27–37. Springer, Cham, 2020.

[23] Stefanos Ougiaroglou, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Manolopoulos, and Tatjana Welzer-Druzovec. Adaptive k-Nearest-Neighbor Classification Using a Dynamic Number of Nearest Neighbors. In *Advances in Databases and Information Systems*, pages 66–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[24] Yang Song, Jian Huang, Ding Zhou, Hongyuan Zha, and C. Lee Giles. IKNN: Informative K-Nearest Neighbor Pattern Classification. In *Knowledge Discovery in Databases: PKDD 2007*, pages 248–264. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[25] Shiliang Sun and Rongqing Huang. An adaptive k-nearest neighbor algorithm. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, volume 1, pages 91–94. IEEE, aug 2010.

[26] Nenad Tomašev, Miloš Radovanović, Dunja Mladenić, and Mirjana Ivanović. Hubness-based fuzzy measures for high-dimensional k-nearest neighbor classification. *International Journal of Machine Learning and Cybernetics*, 5(3):445–458, jun 2014.

[27] Jigang Wang, Predrag Neskovic, and Leon N. Cooper. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28(2):207–213, jan 2007.

[28] Shichao Zhang, Ming Zong, Ke Sun, Yue Liu, and Debo Cheng. Efficient kNN Algorithm Based on Graph Sparse Reconstruction. In X. Luo, J.X. Yu, and Z. Li, editors, *Advanced Data Mining and Applications. ADMA 2014. LNCS*, volume 8933, pages 356–369. Springer, Cham, 2014.

[29] Xiuzhen Zhang and Yuxuan Li. A Positive-biased Nearest Neighbour Algorithm for Imbalanced Classification. In J. Pei, V.S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining. PAKDD 2013*, volume 7819, pages 293–304. Springer, Berlin, Heidelberg, 2013.

[30] Xiao-Feng Zhong, Shi-Ze Guo, Liang Gao, Hong Shan, and Jing-Hua Zheng. An Improved k-NN Classification with Dynamic k. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 211–216, New York, NY, USA, feb 2017. ACM.