Available online at www.sciencedirect.com

ScienceDirect

ICT Express 7 (2021) 414–420

www.elsevier.com/locate/icte

Nearest neighbour methods and their applications in design of 5G & beyond wireless networks

S.A.R. Zaidi

School of Electronics and Electrical Engineering, The University of Leeds, Leeds LS2 9JT, United Kingdom

Received 7 January 2021; accepted 12 January 2021
Available online 5 February 2021

Abstract

In this paper, we present an overview of Nearest neighbour (NN) methods, which are frequently employed for solving classification problems using supervised learning. The article concisely introduces the theoretical background, algorithmic, and implementation aspects along with the key applications. From an application standpoint, this article explores the challenges related to the 5G and beyond wireless networks which can be solved using NN classification techniques.

Keywords: Nearest neighbour search; Nearest neighbour classification; *k*-NN; 5G; Localisation; Beamforming; MIMO; Anomaly; SDN; Network Slicing; NFV; Energy efficiency

Contents

E-mail address: s.a.zaidi@leeds.ac.uk.
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.01.003

1. Introduction

1.1. Motivation

Nearest neighbour (NN) classification belongs to a class of supervised Machine Learning (ML) algorithms. The NN classification adopts NN search techniques to solve the classification problem. The task of classification involves the assignment of a class label to a given query/sample data while utilising a Training dataset which contains several classified samples. Often sample points exist in a certain metric space equipped with a well-defined distance function. The NN search techniques find the sample in training data which has the least distance to the unclassified query. This sample point is termed as NN of the query. Subsequently, the class label associated with the NN can be allocated to the unclassified query. The NN classification algorithm is: (i) simple to implement; (ii) does not require knowledge of the joint distribution of class and sample data; (iii) performs well in most problems especially for low-dimensional data-sets. Despite the popularity of NN classification approaches there is a lack of concise overview which provides a theoretical background while highlighting algorithmic and implementation aspect. Even more importantly, while classification problems are frequently encountered in the design and analysis of modern wireless networks, a concise overview of the important challenges and how they can be tackled using NN classification is missing. To this end, this article is geared towards providing a comprehensive overview of NN methods and their application for design and analysis of 5G and Beyond Wireless Networks.

1.2. Contributions & organisation

The contribution of this article is two fold:

1. First, we provide an overview (see Section 2) of the theoretical and algorithmic framework for solving NN Search and Classification problems. We also highlight how these techniques can be implemented in practice. As an example, a Jupyter Notebook [1] is provided as a supplement to this article. The notebook provides a reference implementation on real data-set.
2. Second, we highlight (see Section 3) key emerging scenarios related to 5G and beyond wireless networks which present a certain classification challenge. A comprehensive overview providing the context of the problem and how some studies have cast them into the NN framework is provided. Moreover, some of the emerging networking scenarios which present similar challenges and yet remain unexplored are briefly mentioned.

2. Theoretical framework

In this section, we provide an overview of theoretical foundations which characterise the NN classification techniques. We review mathematical preliminaries and then explore relevant algorithms that are employed to solve NN search (NNS) classification problems.

2.1. NN search & classification

NN Search Problem

Definition 1: Let S be a set of objects and $d : S \times S \to \mathbb{R}$ be associated distance metric on S metric space. Let $s_i, s_j, s_k \in S$, then the function d satisfies following three properties:
❶ Positive property: $d\left(s_i, s_j\right) > 0$ for $s_i \neq s_j$;
❷ Symmetric property: $d\left(s_i, s_j\right) = d\left(s_j, s_i\right)$; and
❸ Triangle inequality: $d\left(s_i, s_j\right) \leq d\left(s_i, s_k\right) + d\left(s_j, s_k\right)$.
Let $V \subseteq S$ be a certain subset of S of size n, then the nearest neighbour searching (NNS) problem is to build a data structure, so that for an input query point say $q \in S$ an element $v \in V$ is found with $d(q, v_i) \leq d(q, v_j)$ for all $i \neq j, v_j \in V$.

The NNS problem is an example of so-called **proximity problems** [2], i.e. these are class of problems whose definition requires association of a distance function with the corresponding metric space. To provide a concrete overview of the problem, we state a few well-known proximity problems in Table 1. Notice that the list is not exhaustive and interested readers are directed to [2,3] and references therein for detailed treatment.

The NNS problem has received considerable attention from research community whereby several techniques have been proposed to reduce the neighbour search time. Typically $O(\log(n))$ can be achieved by many proposed algorithms (see [4–7] for details). The NNS is frequently employed to implement NN classification. The classification problem involves assigning a label $\omega \in \mathcal{W} = \{1, 2, \ldots, M\}$ to $q \in S$ given a training set $T = \{(s_1, \omega_1), (s_2, \omega_2), \ldots, (s_n, \omega_n)\}$ where $s_i \in V \subseteq S$ and $\omega_i \in \mathcal{W}$.

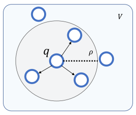
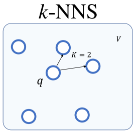
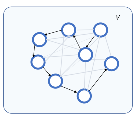
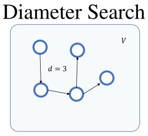
NN Classification

Definition 2: The nearest neighbour classifier performs class assignment for an unclassified sample point ($q \in S$) by assigning it the same class label (say ω) as those of the nearest set of previously classified points. Generally, when such classification is performed by inspecting a fixed number of enumerated distances say k, then this classification is known as $k-$NN classification. k-NN is one of the most frequently employed classifiers for supervised learning. For $k = 1$, the NN classification assigns $\omega = \omega_i$ where ω_i is the label associated with s_i which satisfies, $d(q, s_i) \leq d\left(q, s_j\right) \forall s_i, s_j \in V$.

2.2. Performance of NN classification

The accuracy of the classifier is measured in terms of error-rate which is computed as an average of a loss function which encapsulates the penalty of misclassification. Consider the test-set T introduced before, if the joint-distribution of $P(V, W)$ is completely known than the Bayes classifier provides a

Table 1
Proximity problems.

Proximity problem	Description
Fixed Radius NNS 	The fixed radius nearest neighbour relaxes the exact NNS problem stated in Definition 1, i.e. for a given query point $q \in S$, we are interested in finding $R \subseteq V$ such that $d(q, r) \leq \rho$ for $r \in R$, with ρ being the desired fixed radius.
k-NNS 	The k-NNS problem tries to find $k \leq n$ nearest points in V for a given query $q \in S$, i.e., the output for each q is $R = \{r_i : i = 1..k\}$ such that $d(q, r_1) \leq d(q, r_2) \leq \cdots \leq d(q, r_k)$. In other words, if q is the point in $\mathbb{R}^d$ then k-NNS is geared to find closest k points to q which are contained in set V.
Minimum Spanning Tree 	A minimum spanning tree problem deals with finding a sub-set of edges which connect all objects in V without any cycles and with the minimum possible total edge weight. In the context of proximity problems, edge weight (e_{ij}) can simply be selected distance ($d(v_i, v_j)$) between every set of objects say $v_i, v_j \in V$.
Diameter Search 	The diameter problem is geared towards finding the two objects say $v_i, v_j \in V$ which are maximally distant from each other, i.e. if distance between objects in the set V is enumerated then the maximum value for $d(v_i, v_j)$ is the diameter.

minimum probability of error R^*.[1] The NN classifier is non-parametric and does not rely on the knowledge of the joint distribution. For any number of classes, it has been shown that the probability of error of the NN classifier is bounded above by twice the Bayes probability of error [8,9]. In particular, when $\omega \in \mathcal{W} = \{1, 2, \ldots, M\}$ the probability of error R for the NN-classifier can be bounded as follows:

$$R^* \leq R \leq R^* \left(2 - {}^{(MR^*)}/_{(M-1)}\right), \tag{1}$$

these bounds are the tightest possible.

2.3. Algorithmic & implementation aspects of NN classification

It is well known that the exact computation of NNS in higher dimensional space is computationally expensive [4,10]. Specifically, when $S \subseteq \mathbb{R}^d$ and the training set has n entries, the computational complexity for NNS is $O(dn)$. Therefore, low complexity computation of k-NNS has been an open area of research for several decades. There are a variety of solutions which have been proposed so far. All these proposed approaches try to compute approximate NN rather than the precise one. Naturally, most of these algorithms have limitations, especially in higher dimensions. Consider $\boldsymbol{q} \in S$ be the query or unclassified point, let $\bar{\boldsymbol{s}}_n \in T$ be the NN of $\boldsymbol{q}$ in training set. The distance between $\bar{\boldsymbol{s}}_n$ & $\boldsymbol{q}$ be given by $d(\boldsymbol{q}, \bar{\boldsymbol{s}}_n) = \rho$, then any point (say $\bar{\boldsymbol{q}}$) in T which satisfies the relationship $d(\bar{\boldsymbol{q}}, \boldsymbol{q}) \leq (1+\epsilon)\rho$ is called ϵ-NN of the query. There have been considerable efforts from the research community to design algorithms that facilitate the low-complexity computation of such ϵ-NN search algorithms. Details of such methods have been outlined by the Breuel in [7]. The computational complexity of these approaches and distance metrics that can be applied have been summarised in [11] and are beyond the scope of this article. Nevertheless, in this section, we would like to review one very popular approach which is available in most modern ML frameworks (for instance SciKit Learn, TensorFlow, MATLAB ML Toolbox, etc.).

$K-d$ **Tree NN search:** The $K-d$ tree algorithm belongs to a class of algorithms that project higher dimensional data onto lower dimensions thus speeding up the search. The core idea is to split space into partitions that can be organised in form of trees, enabling faster and localised search for the query within these partitions. The $K-d$ tree algorithm is indeed generalisation of the binary tree to K dimensions. Each node in $K-d$ tree has one K dimensional key, two pointers (as its binary tree) which are either null or contain a sub-tree and a discriminator between 0 and $K-1$. Let $\text{Key}(P) = [\text{Key}_o(P), \ldots, \text{Key}_k(P)]$ be the K-dimensional key associate with the node P, the two pointers of P can be denoted as $\text{Sub}_1(P)$ and $\text{Sub}_2(P)$. Also, the discriminator can be denoted by $\text{Disc}(P)$ for node P. The tree is then constructed such that for any node P in a tree with $j = \text{Disc}(P)$, it is true that for any $Q \in \text{Sub}_1(P)$ it holds that $\text{Key}_j(P) > \text{Key}_j(Q)$. Similarly for any node $Q \in \text{Sub}_2(P)$ it holds that $\text{Key}_j(Q) > \text{Key}_j(P)$. All nodes on the same level in the tree have the same discriminator (root having 0 as discriminator) and it increases with the level in the tree. This could be better understood with an example. The construction can be better understood algorithmically as outlined in Procedure BUILDKDTREE (see next page). Effectively the algorithm splits the space across median values in each dimension. A realisation for $2-d$ dataset is presented in Fig. 2. As is clear from the figure, the data points lie on the lines splitting the plane. The tree representation is also shown in Fig. 2 where at each level split

[1] The optimality of Bayes rule in providing a minimum probability of error is non-trivial to proof. Interested readers are directed to [8,9].

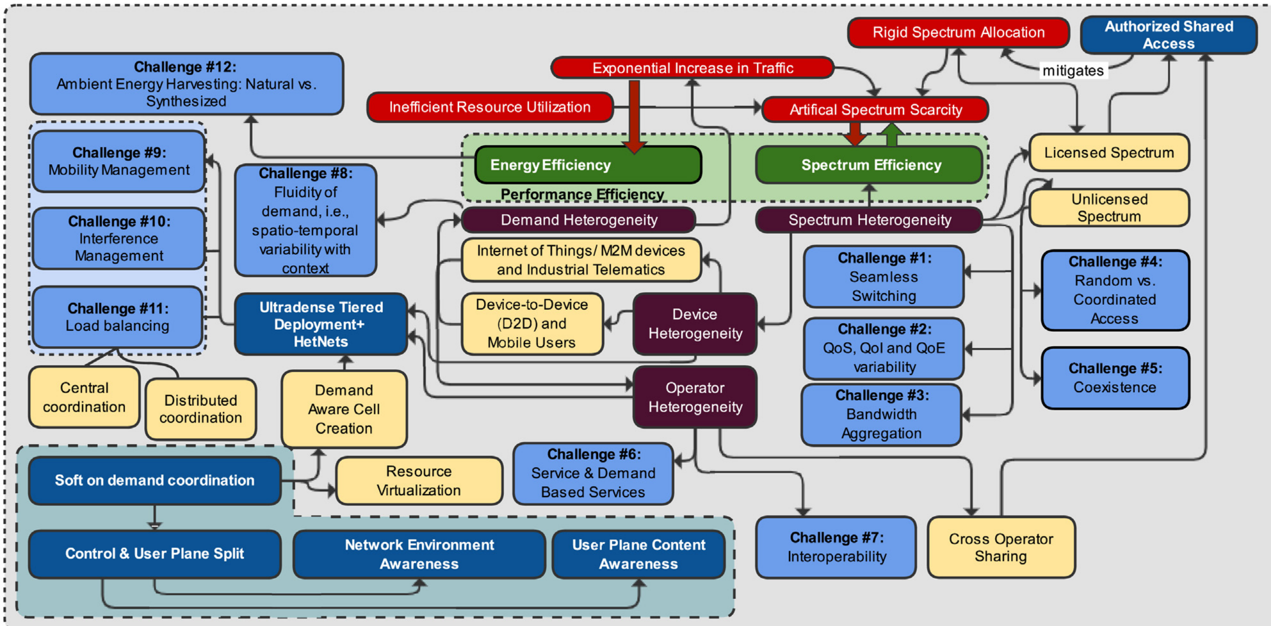


Fig. 1. Performance efficiency metrics, challenges, solutions and evolution of 5G & beyond networks.

is performed around the median value of x- or y-axis in an alternative fashion. The splits of space and organisation of the dataset in tree enable searching in smaller sets speeding up the operation. $K-d$ tree can attain search complexity of $O(\log(n))$. However, the building process has $O(n\log(n))$ time and $O(n)$ storage complexity. Further variants of $K-d$ include the Ball tree method.

Implementation issues: Finally, we briefly want to discuss the implementation of the k-NN algorithm. Let us consider an indoor localisation scenario (see Section 3 for a detailed discussion). We are often interested in identifying the propagation mode for the received signal, i.e. whether the fixed anchor nodes (ANs) with known locations and the tag (a device that needs to be localised) are in Line-of-Sight(LoS) or Non-Line-of-Sight (NLoS). This binary classification problem can easily be solved using the k-NN framework. The process for accomplishing this is described in CLASSIFY() procedure. A Jupyter notebook to accomplish this task in Python using SciKit Learn on real-life data is provided with this article and can be downloaded from [1].

Procedure BUILDKDTREE(T,*depth*)

```
input  : A set of points T and the current depth depth
output: The root of a K − d Tree storing T
initialisation;
if |T| == 1 then
    return Leaf(T) ;
    /* if there is only one point in T return a Leaf
       containing that point                          */
else
    if depth%2 == 0 ;                     // if depth is even
    then
        Split T into two subsets Sub1 and Sub2 with a
        vertical line l through median x-coordinate of
        the points.
    else
        Split T into two subsets Sub1 and Sub2 with a
        horizontal line l through median y-coordinate of
        the points.
    end
    v_left ← BUILDKDTREE( SUB1,depth + 1) ;
    v_right ← BUILDKDTREE( SUB2,depth + 1) ;
    Create a node Node(v) storing l, make v_left the left
    child and v_right the right child of v ;
    return v
end
```

3. Applications in design & analysis of emerging communication networks

3.1. Orchestration, management and allocation of resources for 5G network slicing

5G & beyond networks are envisioned to support three main service classes: (i) enhanced Mobile Broadband (eMBB); (ii) massive machine type communication (mMTC); and ultra-reliable low-latency communication (URLLC) [12]. Each of these service classes is suitable for a different set of verticals, for instance: (i) eMBB services can be employed for HD mobile video streaming; (ii) mMTC services can be employed for a sensor network solution for smart cities; and (iii) URLLC services can be employed for industrial automation. Each of these service classes has a different service blueprint, i.e., different throughput, latency, reliability, energy efficiency requirements. Furthermore, each service type is geared to vary

Procedure CLASSIFY(T)

input : A set of points $T = (x_1, y_1), (x_2, y_2), ldots, (x_n, y_n)$ with labels and a test point q
output: Class of q

initialisation;
Normalise (x, y) ;
Split T into Training αT and Testing Sets $(1 - \alpha)T$;
Fit $k-$NN on αT ;
Use $k-$NN to validate performance on Test data $(1 - \alpha)T$;
Use $k-$NN predict class label of q ;

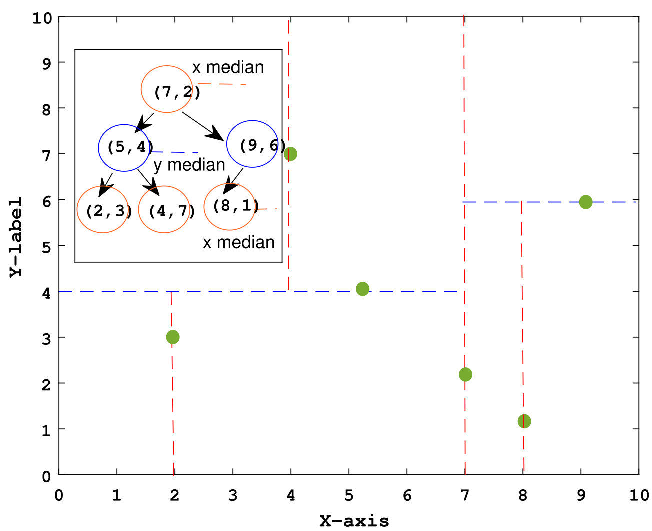


Fig. 2. 2-d Tree for a sample data $T = \{(7, 2), (5, 4), (9, 6), (2, 3), (4, 7), (8, 1)\}$.

in terms of connection densities, duty cycles, traffic profiles, and mobility dynamics. In order, to address such demand, spectrum, and operator heterogeneity, 5G networks aim to adopt flexible and soft resource allocation. Fig. 1 graphically outlines performance efficiency challenges addressed by 5G & beyond networks and tools which are utilised to mitigate the scarcity of resources. As demonstrated by the figure, demand aware cell creation (see [13]) and dynamic allocation of resources is key for the cost-effective delivery of such services. Dynamic and flexible resource allocation is enabled through two key enablers, i.e., Network function virtualisation (NFV) and Software Defined Networks (SDNs). Resources are managed, orchestrated, and allocated in form of Network Slices (NSs) which are effective logical partitions of physical compute, storage, and communication resources [14].

The orchestration of NS and its management while fulfilling the Quality-of-Service (QoS) demands can be further enhanced by proactive demand forecasting [15]. In order to realise forecast aware slicer, the first and important task is to classify the traffic and map it to pre-defined service level agreement (SLA) blueprints (as articulated in 3GPP Specification [16]). The classification of traffic profiles can then be employed by the forecasting engine which can request orchestration of resources from the resource manager. In [15,17] and various other studies such architecture is proposed. The authors have shown that k-NNs provide robust and low complexity classification for the traffic profile. A variant of k-NN scheme is shown to achieve around 95% accuracy.

3.2. Localisation & indoor positioning

Location-based services are core for enabling several 5G applications. Indoor and outdoor localisation scenarios are characterised by different propagation conditions and are of equal importance for different applications. For instance, simultaneous localisation and mapping (SLAM) is a key functionality required for the operation of industrial mobile robots (MRs) and autonomous ground vehicles (AGVs). While outdoor localisation can be enabled by current Global Navigation Satellite Systems (GNSS) such as Global Positioning System (GPS), Galelio Positioning systems etc. The current generation of GPS systems provide around 4.7 m and currently, industry giants (SpaceX, Lockheed Martin) are developing a GPS III, which will be capable of providing 0.22–0.71 m. The positioning performance can be further improved by exploiting the dense deployment of mmWave small cells. Indoor localisation albeit is more challenging than outdoor due to harsh propagation conditions. Indoor positioning has become more imminent due to the COVID-19 pandemic and various solutions are currently being explored to enable contact tracing.

The rapid proliferation of WiFi networks and dense deployments in the urban indoor environment has triggered a lot of interest in exploring indoor localisation based on WiFi Received Signal Strength (RSS). The RSS based localisation schemes can be further classified into two categories: (i) Range-based localisation method, i.e., a method which simply estimated RSS and path-loss exponent and then solve the path-loss equation to find the distance from the anchor; (ii) RSS fingerprinting based localisation, i.e., a method which requires training data which can be exploited to provide localisation estimate when sample data at the unknown location is provided. The RSS fingerprinting method is more robust as RSS readings can experience large variations in indoor environments leading to higher positioning errors.

RSS Fingerprinting as NN Classification

The RSS based fingerprinting can be easily caste into the NN classification framework. Consider that, a user equipment (UE) device at a certain location needs to be localised, then it is possible to build a vector $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$ where $\boldsymbol{x} \in \mathbb{R}^N$ is the vector containing RSS readings from N access points (APS) within the range of UE. If a training set T exists such that $T = \{(\boldsymbol{x}_1, \boldsymbol{\omega}_1), \ldots, (\boldsymbol{x}_1, \boldsymbol{\omega}_M)\}$ where $\boldsymbol{\omega}_i \in \mathbb{R}^2$ is the vector containing coordinates and $\boldsymbol{x}_i \in \mathbb{R}^N$ be observed RSS vector at this coordinates, then by performing NNS for the given $\boldsymbol{x}$, we can find the class label $\bar{\boldsymbol{\omega}}_n$ associated with the NN $\bar{\boldsymbol{x}}_n \in T$ and therefore can estimate the location of the AP. Of course, this requires T to be large enough with fine spatial resolution.

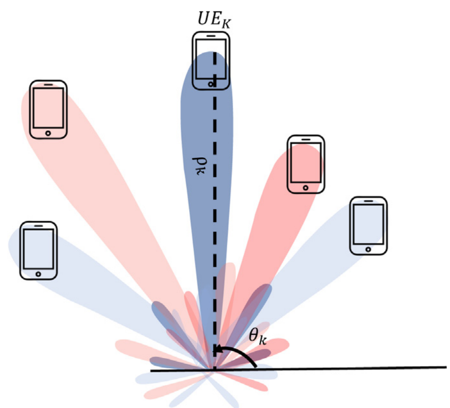


Fig. 3. Illustration of Beam allocation in Multiuser Massive MIMO [22]. The Kth user is located at ρ_k distance and at an angle θ_k.

The NN algorithm can be further improved by selecting k-NNs and selecting a maximally occurring label. This has indeed been explored in [18,19]. Further, improvement is possible by considering APs with antenna arrays and utilising the channel state information (CSI) instead of RSS. This has been recently explored in [20]. Similar approaches can be adopted for finger-printing RSS from Bluetooth enabled devices.

Ultra-wideband (UWB) technology provides a very accurate localisation with up to 10 cm accuracy. However, such accuracy in an indoor environment is only obtained when prior knowledge about the propagation condition is present [21]. For instance, classification based on RSS value can serve as a first step to establish whether there exists line-of-sight or non-line-of-sight propagation between the anchor and the UE. This can then be used to refine time-difference-of-arrival (TDoA) estimates. Naturally, the RSS classification can be cast into k-NN framework.

3.3. Beam allocation multiuser massive MIMO

Massive multiple-input multiple-output (MIMO) cellular networks are envisioned as a key enabler for enhancing spectral efficiency in 5G and beyond systems. Consider a beam allocation problem for the multiuser Massive MIMO system as outlined in [22] and shown in Fig. 3. There are K users each with a single receive antenna. The base station (BS) is furnished with $N \gg K$ fixed beam formed by Butler network with a linear array of N identical and isotropic radiating elements. The problem is how to allocate these beams optimally to K users to maximise the spectral efficiency. Authors in [22] propose the use of cloud resources to store training information. This training information is comprised of beam allocation in response to a certain spatial configuration of UEs. Then for any given set of K users, k-NN classification can be used to characterise the spatial pattern of users (encoded in (ρ_k, θ_k)). The k-NN classifier will mark the pattern with the best class according to NNS and the class vector can basically encode configuration pattern for beam allocation.

More recently, Reflecting Intelligent Surfaces (RIS) have been proposed to enhance coverage and capacity in 6G wireless networks [23]. The k-NN framework can also be adapted to optimise the configuration of RIS elements. This however remains an open issue and has not been extensively investigated.

3.4. Sleeping cell anomaly detection

The sleeping cell problem (SCP) is a well-known problem in the Long Term Evolution (LTE) cellular networks [24,25]. The SCP leads to cell outage and consequently lack of network services. The SCP can manifest due to the Random Access Channel (RACH) failure [25]. Cell outage detection and identification of SCP conditions therefore are one of the core features of big-data empowered self-organising networks (SONs) [24]. Under SCP cell outages cannot be detected as the outage event manifests without triggering the alarms. Consequently, no SON compensation function such as self-healing can be deployed unless either multiple user complaints are received or the outage is detected through drive testing. In such a scenario either UEs can be configured to periodically report signal strength parameters or neighbouring cells can log such information. The measurement can then be utilised by a k-NN classifier for anomaly detection, i.e. labelling the readings as anomalous or normal. Anomaly detection can lead to the rapid identification of SCP as shown in [24]. It has been shown by the authors that this approach yields 94% detection accuracy.

Anomaly detection is of importance in the context of network security as well. Often intrusion is identified by observing anomalous traffic. k-NN classifiers have been extensively used for such functionality. Interested readers are directed to [26] and references therein.

3.5. Energy saving for smart devices

Mobile devices have increasingly become a popular channel for digital engagement. With ever-growing applications (Apps), energy consumption and battery life-time are becoming core issues for manufacturers. It is possible to exploit contextual information by building a device and application profile to predict optimal configurations for the wireless connectivity and location interfaces on the mobile device. Such a technique can provide 24% energy savings on average as shown in [27]. The authors demonstrate that the classification and profiling can be attained by the k-NN classifier with 90% accuracy. Similar techniques can be employed for the Internet-of-Things (IoT) devices. Since most of the IoT devices are connected to the cloud through the gateway, k-NN classification can be offloaded to the cloud. This way optimisation of the configurations for energy savings can be attained without incurring significant energy penalty for obtaining such configuration on devices.

4. Conclusion

In this paper, we presented a brief overview of nearest neighbour methods. We provided a concise review of the mathematical background and highlighted the achievable performance for such methods when applied for solving a classification problem. We briefly discussed algorithmic aspects of such methods and highlighted key challenges in applying these techniques. Based on the developed statistical and algorithmic framework, we review some emerging applications where these classification techniques can be employed.

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.

Acknowledgement

This work is supported by the EPSRC, UK EP/S016813/1, EP/N010523/1, and Royal Academy of Engineering, UK 122040 grants.

References

[1] S. Zaidi, A jupyter notebook for knn los/nlos classification, 2020, URL https://salirzaidi.github.io/K-Nearest-Neighbour-&-Propagation-Mode-Classification/.

[2] M.T. Dickerson, D. Eppstein, Algorithms for proximity problems in higher dimensions, Comput. Geom. 5 (5) (1996) 277–291.

[3] K. Clarkson, Nearest-neighbor searching and metric space dimensions, 2005.

[4] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: Proceedings of the thirtieth annual ACM symposium on Theory of computing, 1998, pp. 604–613.

[5] M. Muja, D.G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, IEEE Trans. Pattern Anal. Mach. Intell. 36 (11) (2014) 2227–2240.

[6] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the twentieth annual symposium on Computational geometry, 2004, pp. 253–262.

[7] T.M. Breuel, A note on approximate nearest neighbor methods, 2007, arXiv preprint cs/0703101.

[8] R. Castro, Cal learning theory, in: Lecture Notes, 2018, URL https://www.win.tue.nl/rmcastro/2DI70/files/2DI70_Lecture_Notes.pdf.

[9] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[10] K.L. Clarkson, An algorithm for approximate closest-point queries, in: Proceedings of the Tenth Annual Symposium on Computational Geometry, SCG '94, Association for Computing Machinery, New York, NY, USA, 1994, pp. 160–164, http://dx.doi.org/10.1145/177424.177609.

[11] M. Reza, B. Ghahremani, H. Naderi, A survey on nearest neighbor search methods, Int. J. Comput. Appl. 95 (25) (2014) 39–52.

[12] P. Popovski, K.F. Trillingsgaard, O. Simeone, G. Durisi, 5g wireless network slicing for embb, urllc, and mmtc: a communication-theoretic view, Ieee Access 6 (2018) 55765–55779.

[13] U.S. Hashmi, S.A.R. Zaidi, A. Imran, User-centric cloud ran: An analytical framework for optimizing area spectral and energy efficiency, IEEE Access 6 (2018) 19859–19875.

[14] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, V.C. Leung, Network slicing based 5g and future mobile networks: mobility, resource management, and challenges, IEEE Commun. Mag. 55 (8) (2017) 138–145.

[15] V.P. Kafle, Y. Fukushima, P. Martinez-Julia, T. Miyazawa, Consideration on automation of 5g network slicing with machine learning, in: 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), 2018, pp. 1–8.

[16] 3GPP, 3gpp tr 28.801: Study on management and orchestration of network slicing for next generation network (release 15), 2018.

[17] N. Salhab, R. Rahim, R. Langar, R. Boutaba, Machine learning based resource orchestration for 5g network slices, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

[18] Y. Xie, Y. Wang, A. Nallanathan, L. Wang, An improved k-nearest-neighbor indoor localization method based on spearman distance, IEEE Signal Process. Lett. 23 (3) (2016) 351–355.

[19] J. Oh, J. Kim, Adaptive k-nearest neighbour algorithm for wifi fingerprint positioning, ICT Express 4 (2) (2018) 91–94, http://dx.doi.org/10.1016/j.icte.2018.04.004, sI on Artificial Intelligence and Machine Learning. URL http://www.sciencedirect.com/science/article/pii/S240595951830050X.

[20] A. Sobehy, E. Renault, P. Muhlethaler, Csi-mimo: K-nearest neighbor applied to indoor localization, in: ICC 2020: IEEE International Conference on Communications, 2020.

[21] F. Che, A. Ahmed, Q.Z. Ahmed, S.A.R. Zaidi, M.Z. Shakir, Machine learning based approach for indoor localization using ultra-wide bandwidth (uwb) system for industrial internet of things (iiot), in: 2020 International Conference on UK-China Emerging Technologies (UCET), 2020, pp. 1–4.

[22] J.-B. Wang, J. Wang, Y. Wu, J.-Y. Wang, H. Zhu, M. Lin, J. Wang, A machine learning framework for resource allocation assisted by cloud computing, IEEE Netw. 32 (2) (2018) 144–151.

[23] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, Z. Qin, M. Di Renzo, N. Al-Dhahir, Reconfigurable intelligent surfaces: Principles and opportunities, 2020, arXiv preprint arXiv:2007.03435.

[24] A. Imran, A. Zoha, A. Abu-Dayya, Challenges in 5g: how to empower son with big data for enabling 5g, IEEE Netw. 28 (6) (2014) 27–33.

[25] S. Chernov, M. Cochez, T. Ristaniemi, Anomaly detection algorithms for the sleeping cell detection in lte networks, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), 2015, pp. 1–5.

[26] Y. Liao, V.R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, Comput. Secur. 21 (5) (2002) 439–448.

[27] B.K. Donohoo, C. Ohlsen, S. Pasricha, Y. Xiang, C. Anderson, Context-aware energy enhancements for smart mobile devices, IEEE Trans. Mob. Comput. 13 (8) (2014) 1720–1732.