K-Nearest Neighbour Algorithm for Classification of IoT-Based Edge Computing Device



Micheal Olaolu Arowolo, Roseline Oluwaseun Ogundokun, Sanjay Misra, Jonathan Oluranti, and Akeem Femi Kadri

1 Introduction

The Internet of Things (IoT) is an advanced model intended to offer enormous applications that are presently a portion of our everyday existence. Under dynamic networks, millions of smart devices are installed to provide lively functionality, including connectivity, tracking, and control of vital infrastructures [1, 3]. However, due to the shortage of bandwidth and money, this enormous creation of IoT gadgets and the resulting big data traffic caused at the edge of the network imposed further loads on the advanced centralized cloud storage model. Edge computing (EC) is now evolving as a revolutionary approach that puts data conversion and depository close to ultimate consumers, contributing to what is known as EC-aided IoT. While this model offers new functionality and quality of service (QoS), it likewise raises tremendous dangers in the areas of data protection and confidentiality [2].

Presently, the IoT scenario has attained a great deal of popularity. It encompasses a software and hardware infrastructure that ties the corporeal world to the Internet.

M. O. Arowolo · R. O. Ogundokun (✉)
Department of Computer Science, Landmark University, Omu Aran, Nigeria
e-mail: arowolo.micheal@lmu.edu.ng; ogundokun.roseline@lmu.edu.ng

S. Misra
Department of Computer Science and Communication, Ostfold University College, Halden, Norway
e-mail: sanjay.misra@hiof.no

J. Oluranti
Covenant University, Ota, Nigeria
e-mail: oluranti@covenantuniversity.edu.ng

A. F. Kadri
Department of Computer Science, Kwara State University, Malete, Nigeria
e-mail: akeem.kadri@kwasu.edu.ng

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

S. Misra et al. (eds.), *Artificial Intelligence for Cloud and Edge Computing*, Internet of Things, https://doi.org/10.1007/978-3-030-80821-1_8

The number of IoT devices has risen significantly owing to the exponential advancement of concentration in this archetype. More than 75 billion gadgets are predicted to be coupled to the Internet by 2025, contributing to a financial impact on the worldwide economy. Usually, IoT gadgets have restricted processing capacity and minor reminiscences and can produce huge quantities of data. In our homes, towns, cars, and industries, reduced-power and coupled systems, primarily sensors, would be utilized. Cloud storage can be sufficient for the growth of the IoT market, but together with potential frequency range capacity, the interruption instigated by data transmission is inappropriate for certain assignments (e.g., well-being keep tracking) [2].

Limitations in resource-scarce computer computational resources limit the application of multifaceted machine learning algorithms; multiple software agent frameworks provide stable and efficient resolutions for optimizing the implementation of various edge computing. Low-data fusion is concerned with tasks for delivering edge components, while calculation needs to be delegated to more powerful systems for a profound understanding of the data and administrative intentions. Transferring raw data to cloud servers upsurges the cost of transmission, causes device responses to be delayed, and exposes private data. A real-world approach is to examine computing data nearer to its origin and convey it to the inaccessible servers that solitary, the information required for additional cloud processing to resolve the problems. Instead of distant areas, edge computing pertains to the data transformation accomplished as close as possible to the information sources [3].

Effective artificial intelligence (AI) algorithms can be implemented on IoT gadgets, such as AI techniques include support vector machine (SVM), deep learning (DL), and, among other things, neural networks (NNs) to make this scenario possible. It is noteworthy that NNs in the application process need less computational resources than in the training phase. In terms of RAM, the AI technique can be run on a computer with low power (random access memory). AI has been used for the prediction and detection of diseases [4–6]. For various subjects, for instance, smart cities, computer vision, medical care, self-propelled, and machine learning algorithms are used [7, 8]. There are numerous instances of what way machine learning may be applied on edge gadgets in these areas [9–11].

Edge computing is increasingly linked with AI, advancing one another in the equivalent of efficiency: edge intelligence and smart edge. The intellect of the edge and the smart edge are not distinct from each other. Edge intelligence is the target, and intelligent edge DL techniques are as well a component of edge intelligence. Consecutively, the smart edge could deliver edge intelligence with advanced service material and asset utilization [12].

This research, therefore, uses machine learning systems to deploy an edge computing device and evaluate the algorithms of machine learning that can be applied in edge computing using the edge analysis approach to machine learning.

The article's remaining part is structured as thus: Unit 2 discussed the literature review on AI, ML, edge computing, and IoT devices as well as the interrelated studies. Unit 3 discussed the materials and methods. The ML procedure used for the execution of the system in the study was conferred in the unit as well. Unit 4 discussed the execution and testing of the system. Here, the findings discovered are also conferred in this unit. The study came to an end in Unit 5.

2 Background and Literature Review

In this section, we provide the background of the work which includes an architecture for IoT edge-centric, AI and edge computing, machine learning, and other related works.

Upon the accelerated spread of IoT technologies by 2025, it is estimated that there will be 77.44 billion IoT devices [13]. Different IoT architectures [14–16] were introduced by diverse organizations from diverse backgrounds for the vast number of IoT applications, and edge computing has been documented as an imperative abutment for IoT structures [15].

There are four key parts of the edge-centric IoT architecture: the cloud, the end machines of the IoT, the edge, and its apps. The sketch of the architecture carves over together the resources obtainable and every group's unique functionality. To make their lives more comfortable, consumers use intelligent IoT software, while more commonly they connect with IoT end devices via cloud or edge interactive interfaces somewhat than unswervingly communicating with IoT finishing gadgets. Deeply rooted in the real universe are the IoT end units. They feel things as they are and assume action to regulate things as they are, but in computer-heavy activities, they are not advanced. There are almost unlimited resources in the cloud; nevertheless, they are generally physically situated distant from the final gadgets. As a result, an IoT cloud-centric structure is generally unable to perform effectively [17], particularly once the structure possesses the requirements for the present. This could together coordinate the further three groups to concur, agree, and counterpart the cloud and IoT client gadgets for enhanced accomplishment, with the edge being a dominant element of the entire architecture.

IoT users submit IoT data access queries or IoT device control commands to the edge-centric IoT architecture. Via a net or mobile application-enabled interface supported by the cloud or the edge, such requests and commands can ultimately enter the edge layer. The edge layer would then manage them, either routing them to IoT users' gadgets or managing them on behalf of IoT gadgets on the edge layer. The edge layer, not solitary, bridges them with consumers and the cloud by communicating with IoT gadgets but can also stock the gathered records submitted from IoT gadgets and unload major computational requirements, for instance, big data scrutiny and robust protection procedures from IoT users' gadgets. It is also possible to move several current IoT user's system facilities from the cloud to the edge and to configure them depending on the requirements of IoT users' gadgets. The edge can operate independently of the cloud in relation to the interaction between the cloud and the edge, or collaboratively the edge can work with the cloud. The edge is efficient enough to manage IoT application needs in the first model. It will, for example, provide storage and processing resources to satisfy all IoT device demands. Furthermore, the edge receives cloud resources to control the edge layer or to better manage the needs of the IoT framework. Besides, it is possible to move several current IoT end system facilities from the cloud to the edge and to configure them depending on the requirements of IoT users' gadgets. The edge can operate

independently of the cloud in terms of the interaction between the edge and the cloud or the edge can collaboratively work with the cloud. The edge can sufficiently manage IoT implementation requirements in the first model. It will, for example, provide storage and processing resources to satisfy all IoT device demands. In model 2, the edge receives cloud resources to control the edge layer or to better manage the needs of the IoT framework. Second, mechanically, the layer of the edge is similar to the IoT users' units. It could meet the instantaneous necessities required in the plan for security [18]. Third, data from several IoT end devices is obtained and stored by the edge layers. Therefore, the edge is a safer area to decide protection verdicts relative to end devices, so a top protection verdict relies on the algorithm's performance and the availability of appropriate information simultaneously. The edge layer, for example, will detect intrusion more effectively with more data [19–21]. Several security procedures would be transformed to steering strategies with the recognition of software-defined networks and virtualization; however, they can clash with each other. We will overcome these disputes at the edge with an analysis of the entire system coupled by the edge. Fourth, it is mostly not feasible to install and maintain firewalls on any IoT end system, given resource limitations, maintenance expense, and incredibly large size of end devices. Instead, installing firewalls on the edge layer allows approaching threats to be more easily cleared and obstructed. Fifth, the edge layer should keep track of the movement of these devices, given the versatility of end devices, and provide them with a continuous secure relation. Moreover, the very secure link between the users' gadgets and the edge layer helps to create a good trust between them. It alleviates the issues of confidence building between these devices. Sixth, with the cloud, the edge typically has a huge-velocity communication. The edge can communicate with the cloud layer for protection support whenever it is required. For instance, the cloud can offer position and role authentication for the edge, as shown in Sha, Alatrash, and Wang [22], and the cloud can create powerful protection measures to secure the edge. First, we discuss edge-centered IoT security resolutions.

The advent of edge computing in IoT networks has brought many research challenges, such as the design of dynamic computing offloading schemes, allocation of resources (e.g., computing resources, spectrum resources), and power management transmission. Over the last year, IoT generated enormous interest in science. The concept of IoTs is seen as part of the future Internet and will consist of billions of smart things communicating with each other. In the nearest future, International Networks will constitute hierarchically organized embedded devices that expand the world's borders with physical entities and artificial resources. IoT would bring new capabilities to connected things. IoT can communicate with one another without human interference. Some early IoT implementations in the healthcare, transportation, and automotive industries have already been developed [23, 24]. IoT systems are currently in their infancy; however, there have been several recent advances throughout the integration of sensor objects into the cloud-based Internet [23, 25, 26]. IoT creation includes several issues, including communication networks, applications, protocols, and standards [23, 27]. The term Internet and Things refers to a worldwide interconnected network with an emphasis

on visual technologies, networking, information-processing, and connectivity, the latest version of interaction which could be ICT [28]. Smart sensing and wireless communication technologies, in particular, became some of the IoT's components and the newest opportunities and forensic horizons appeared [29].

IoT is triggered by using Radio Frequency Identification technology, which is becoming much more popular in various industries such as manufacturing and retail [30, 31]. Disruptive technologies are the term used for technological advances that have changed the way ordinary products are done in the world. The IoT has been implemented in many aspects of daily life as part of the Disruptive Technology community. It is projected that the demand for IoT gadgets will upsurge to 20–30 billion by 2020 [32].

There have been numerous innovative pieces of machinery intended at operating at the edge of the net in the advancement of edge computing, with similar perceptions but distinct emphases, for instance, cloudlet, micro data centers (MDCs), fog computing, and mobile edge computing (now multi-access edge computing). The edge computing group, however, is yet to attain a compromise on the homogenous concepts, architectures, and edge computing protocols. For the suite of new technologies, the common word "edge computing" is used [33]. Because of its benefits of lowering information transfer, enhancing facility potential, and simplification of cloud computing strain, edge computing has become an important solution for breaking the bottleneck of emerging technologies. The architecture of edge computing will transform to being a significant accompaniment to the cloud, even substituting the position of the cloud in few instances [34]. Edge computing is an enabling resource that facilitates network IoT applications, lower data on behalf of cloud facilities, and upstream information on behalf of IoT services. They are interchangeable as opposed to fog computing, but edge computing focuses further on the stuffed side, whereas fog computing concentrates further on the infrastructure side. Both computing and network resources along the path between data sources and cloud data centers can be an "edge." Edge computing's rationale is that computing can take place near data sources. At the edge, not only do stuff request cloud service and content but also perform cloud computing tasks, as well as transmitting requests and distribution services from the cloud side to the user side. Edge can perform computing offloading, data storage, caching, and processing [35].

Edge computing consists of important metrics, such as the delay in the definition. Besides, high bandwidth, which is accomplished by restricting the flow of data. Edge computing servers, and with a lengthy latency rate for a centralized server. Energy usage is also a big concern, apart from this. External resource systems-related computational work contributes to an improved battery life of user devices. Edge computing is fast, stable, scalable, and reliable in terms of speed [36].

Edge computing may deliver several benefits. The edge computing model, for example, can be flexibly applied from a single house to society or even the size of the city. Edge computing for services requiring consistent and low latency, such as health emergencies or public safety, is an effective paradigm since it can save time for data processing as well as simplify the layout of the network. Decision and predictions are possible to diagnose and transmit from the edge of the network,

which is more effective compared to data collection and central cloud decision-making. Edge computing exceeds geographic-based applications such as transport and utility management. Due to the understanding of place, that is, cloud computing data could be obtained and processed based on geographical position in edge computing, without being loaded into the cloud [37].

The machine learning procedures that could be utilized in resource-constrained edge computing sceneries are explored in this report. The ML procedures implemented in the subsequent paragraphs are the utmost commonly utilized and present the issue of taking artificial intelligence to hardware-resource-constrained devices. Machine learning models currently require sufficient memory to store training data and computing power to train large models. New computer models have been designed to work effectively on edge equipment by using shallow models that require sufficiently low processing power to be used on IoT devices. Alternatively, reducing the size of the inputs of the model for classification applications will increase the learning speed [38].

The algorithm k-nearest neighbors (KNN) is a procedure utilized in the identification of patterns, built on the features of the artifacts similar to the one deliberated. This approach is utilized for both problems with classification and regression. There are numerous updated variants of KNN that help execute the procedure on hardware-constrained gadgets, with ProtoNN being the most creative. It is an algorithm based on KNN. The key KNN problems for edge computation are the size of the training data (the algorithm uses the entire dataset to produce predictions), the time of prediction, and the choice of distance metrics [39]. For classification and regression problems, tree-based ML procedures are utilized, which is a very known routine in the IoT area. Nevertheless, the normal tree algorithms could not be implemented on them owing to the inadequate resources of the gadgets. An algorithm that is evolving is Bonsai. The tree technique is specifically intended for IoT gadgets that are strictly resource-constrained, and it retains forecasting accuracy whereas lessening the model size and cost of estimation. It initially discovers that the size model is decreased by a single parse tree and then makes nonlinear estimation through the internal nodes and leaf ones. Bonsai finally studies the sparse matrix, jutting all the knowledge into a low-dimensional space in which the tree is taught. This makes it possible to carry the algorithm to tiny devices such as IoT ones [40]. At the embedded stage, one of the most commonly used ML algorithms is the SVM. SVM is an algorithm for supervised learning that can be utilized for problems with both classification and regression. By outlining an optimum hyperplane that split up every group, the algorithm discriminates between two or more classes of data. Help vectors are the data nearest to the hyperplane, which will appear in the hyperplane itself being redefined if removed. The critical elements of the dataset are considered for these purposes. The loss function utilized by the procedure is usually the loss of the hinge and the optimization function is the descending gradient form [2].

We assume AI and edge computing's confluence is natural and imminent. In effect, there is an engaging interaction between them. On the one hand, AI provides technology and approaches for edge computing, and edge computing can unleash its power and scalability with AI. On the other hand, edge computing provides

scenarios and platforms for AI, and with edge computing, AI can extend its applicability.

AI Provides Edge Computing with Technologies and Methods Edge computing is a disseminated computing model in general, where software-defined networks are developed to distribute data and offer sturdiness and elasticity of services. Edge computing faces resource utilization challenges in multiple layers, for instance, frequency of the CPU loop, authority of entry, radio frequency, bandwidth, and so on. As a consequence, several effective optimization methods are highly challenging to increase machine performance. AI technologies are capable of fulfilling this mission. Essentially, AI models extract from actual scenarios unrestricted optimization issues and then iteratively find the asymptotically optimal solutions using stochastic gradient descent (SGD) methods. Either mathematical methods of learning or methods of deep learning may provide the edge with support and guidance. Also, reinforcement learning, like multi-armed bandit theory, multi-agent learning, and deep Q-network (DQN), plays an increasing and significant role in edge resource distribution issues [4].

Edge Computing Provides AI with Scenarios and Platforms The explosion of IoT devices renders the Internet of Everything (IoE) a fact [41]. Rather than mega-scale cloud data centers, ubiquitous and globally dispersed smartphone and IoT devices are generating more and more data. Many further implementation examples will significantly accelerate the realization of AI from theory to reality, such as in intelligent networked cars, autonomous driving, smart house, smart city, and real-time public security data collection. Also, AI systems can be moved from the cloud to the edge with high connectivity efficiency and low computing power requirements. Edge computing, in a phrase, offers a heterogeneous framework full of rich capabilities for AI. AI chips with computational acceleration such as Field Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), and Neural Processing Units (NPUs) are now increasingly becoming possible to combine with smart mobile devices. To support the edge computing model and to promote DNN acceleration on resource-limited IoT chips, more companies are interested in the design of chip architectures. Edge hardware updates also inject vigor and energy into AI.

2.1 Related Works

The key problems related to QoS specifications for applications operating on traditional paradigms such as fog and cloud environments were explored by Guevara et al. [42]. A standard machine learning classification strategy was then presented. The paper's key gain is to concentrate professionally on facets of machine learning. However, the paper usually suffers from inadequate numbers of articles presented, and especially incomplete and insufficient records of the recently written

manuscript. Furthermore, the investigation agonizes from the deficiency of the high content of the papers studied. The organization of the study is still not very pleased.

Wang et al. [43] studied several offloading procedures that are discussed in three major groups constrained by two edges, mobile gadgets, and edge clouds, namely, contact connections called intermediate carriers. The key drawback of their investigation was that the topic entirely is considered and the complete relevant studies are checked collectively. The investigation, however, agonizes from the absence of newly issued manuscripts in the relevant research area (i.e., about 7% issued in 2018, 4 out of 54). Additionally, this survey has sacrificed specific and comprehensive facets of each individual in the two-end method referred to above for the sake of honesty. Some essential factors, for instance, fault tolerance and protection matter in the scheme, which consecutively unswervingly influence the complete competency of the system, are also neglected. This degradation, which is QoS, QoE, could deceive the operator's viewpoint. There is no systemic format for the paper to select papers, too.

Aazam et al. [44] reviewed few offloading structures in the study interest of computational models, for instance, edge, fog, cloud, and also IoT. After these paradigms, they introduced an arrangement. It also discusses permissive expertise for offloading as middleware and associated features. The evaluation was well classified as a strength in the predefined standard, with adequate instances for separately described standards. A judicious proportion of recently issued manuscripts (approximately 24% printed in 2017 – 12 out of 51) is also included in the topic, nevertheless, with certain crucial features, for instance, coarseness and flexibility in previous researches, which does not evaluate those studies. The paper does not possess a systematic setup for choosing manuscripts.

Mach et al. [45] also pronounced certain use cases, range of capabilities, standardization, and offloading of computation in MEC settings. The investigation was made as an advantage by examining many MEC-connected studies and papers, of which 23% were recently issued at the period of acceptance (e.g., available studies in 2016 – 29 out of 124). The manuscript, however, in some respects, such as the proposed granularity, is not well organized. Some important topics and methods are also not enclosed in the previous studies. By way of additional disadvantage, some aspects are simplified that lead to loss of generality. The manuscript also does not provide a methodical setup for selecting papers. It is also worth noting that this paper does not cover future paths well.

Boukerche et al. [46] examined energy sensitivity in Mobile Cloud Computing (MCC) in the area of MCC and Green Cloud Computing techniques. They studied procedures, design, setting up, and balance procedures. Next, the benefits and drawbacks of those experiments were contrasted and classified in terms of offloading mechanism and resource manager forms. The quality of their manuscript is its proficient analysis of the topic that has been sufficiently structured to address energy-conscious concerns. Nevertheless, it is a difficult task to make a firm indeterminate state amid evolving pieces of machinery which include MCC and MEC. They provide a precise sense and description for all of them as a result of the lack of principles in the discussion, and this is exactly one of the feeble goals of

this study. As another example, the analysis does not fully cover similar topics. This study also hurts from a deficiency of newly issued manuscripts in the re-examined sector (approximately 5% available in 2018, 7 out of 141, and approximately 8% in print in 2017, 11 out of 141) despite the consideration of significant numbers of checked articles.

Peng et al. [47] examined several papers relevant to MEC in terms of design, amenity acceptance, and delivery. Computing offloading and data offloading are categorized as two important facets of the MEC model for amenity acceptance. Edge server (ES), amenity delivery and its practical predictor, ES implementation, and resource utilization are checked for service provision. This survey also explores several other problems, such as MEC implementations. The analysis is supplied with reasonable recently published articles as a strength (about 30% published in 2017 – 37 out of 123). The investigation is not, however, theoretically well organized. The investigation likewise hurts from a fluent summary of the literature with inadequate technical descriptions and predefined fields in each division.

Shakarami, Shahidinejad, and Ghobaei-Arani [48] proposed a systematic analysis of game-theoretical offloading methods in the MEC setting. In the form of tables and maps, they have contrasted different critical aspects of literature. Their manuscript is the solitary comprehensive study evaluation that covers the topic in an efficient technique, as a power. The investigation likewise encompassed similar newly in print manuscripts about the relevant field. By way of a downside, the article is not linked to approaches of ML, which is not, certainly, its focus.

Cao et al. [49] suggested an investigation on basic principles in MEC with an emphasis on ML and the foremost implementations. As a strong point, their investigation emphases competently on a single issue that has no literature context. This investigation agonizes a deficiency of ample convincing papers to accurately include the field of study. As an additional downside, there is no systematic format for the paper to pick papers, too. In this article, potential instructions are still not adequately powerful and well protected.

In another work, Taleb et al. [50] presented MEC (mobile edge computing) survey that explores the major technical enablers in this domain. It discusses MEC implementation, taking into account both the experiences of individual providers and a network of mobility enabling MEC platforms. Also discussed here are various potential MEC implementation solutions. It also discusses the study of the MEC reference architecture and its core implementation scenarios that can help application developers, service providers, and third parties with multi-tenancy support. The latest standardization efforts and future open testing issues are also detailed in this work.

In the paper by Madakam et al. [51], the key aim of this study is to describe the Internet of Things, architectures, and critical innovations in our everyday lives and their applications. In this paper, the key observations are made. (a) The traditional definition of IoT is not present. (b) Universal standardization at the architectural level is needed. (c) Technologies differ from vendor to vendor and thus interoperability is essential. (d) There is a need to develop uniform protocols for better global management.

The study by Dolui et al. [52] examines the effectiveness and contrasts the feature sets of various forms of edge computing models, including fog computing, cloudlet, and mobile edge computing. Edge computing has been a field of focus for researchers, with a lot of exposure to IoT and smartphones that require real-time answers.

Mao et al. [53] presented a study of state-of-the-art mobile edge computing technology with an emphasis on radio (network) optimization and computational tools.

3 Materials and Methods

This section discussed the materials used for this study in terms of datasets used for the implementation. The machine learning algorithm adopted for this study was also discussed in this section.

3.1 Datasets

The datasets used for this study is presented as an addition to the work published in the MDPI Sensors Journal entitled "Process Management in IoT Operating Systems: Cross-Influence between Processing and Communication Tasks in End-Devices." From the perspective of Edge Computing, this thesis focuses on process management in IoT operating systems. Specifically, in sensor end devices, we carried out an empirical analysis of the cross-influence between processing and communication tasks [54].

3.2 K-Nearest Neighbor (KNN) Algorithm

The ML procedure used for this system's implementation is KNN. In several aspects, for instance, expert networks and intellectual structures, artificial learning has been widely used. An intelligent machine can hardly be considered a real intelligent system without learning capacity. To resolve the deficiencies of expert systems and smart structures, several pieces of research began with ML. The KNN procedure is a popular method of ML built on computer codes to give a run down and creäte human-acquired knowledge. It is straightforward, spontaneous, and quick to execute and possesses an extensive variety of applications. It is practically ideal for multiple kinds of data structures. Consequently, the KNN method has been commonly used as a non-parametric technique appropriate for non-inactive and nonlinear active structures for forecasting analysis in different research

areas: Davis and Nihan [55] utilized the KNN technique in interim forecasting of highway traffic. Mehrotra and Sharma [56] proposed a version of the k-nearest neighbor resampled optimized for utilization of manifold prognosticator variables. By incorporating two separate methodologies for the k-nearest neighbor approach, Bannayan and Hoogenboom [57] developed a method to forecast the comprehension of instantaneous regular climate results.

Mangalova and Agafonov [58] considered a modeling technique aimed at forecasting the power production of wind ranch energy generators, and the KNN process is the fundamental modeling solution. An improved KNN model was suggested by Cai et al. [59] to increase prediction exactness built on spatiotemporal similarity and to accomplish multiphase prognostication. In the present centuries, in the researches for the time series prognostication utilizing KNN technique, comprehensive and elegant methods have been published, and few researchers have enhanced the KNN technique and extended it to various research areas. For short-term traffic forecasting, Cheng et al. [60] presented an adaptive spatiotemporal k-nearest neighbor technique. Martínez et al. [61] predict using a dissimilar specialized KNN learner every different season. Fan et al. [62] suggest an innovative immediate load prognostication method to evaluate the energy characteristics and regulations built on the weighted k-nearest neighbor algorithm.

Xu et al. [63] presented a procedure to simulate road traffic situations in time series dependent on kernel k-nearest neighbors (kernel-KNN). Multidimensional k-nearest neighbor (MKNN), autoregressive moving average model (ARMA), and quadratic regression model are included (2020) to develop the EEMD-FFH method. In a detailed analytical analysis using a broad data collection of M3-competition industrial time series, Kück and Freitag [64] analyze the predicted success of local-nearest neighbor methods, which are based on the concept of dynamical structures. Furthermore, the KNN procedure can also be utilized to classify (Hattori and Takahashi [65], Hattori and Takahashi [66], Jiang et al. [67], Jiang et al. [68], Miao et al. [69], Tan [70], Tan [71], Wan et al. [72], Yoon and Friel [73], Zhang et al. [74], Zhang et al. [75]).

The performance of the KNN procedure relies on the demonstrativeness and comprehensiveness of the data, comparable to further data-driven procedures. The KNN procedure has a very basic operating process as a classical regression model: k signifies the sum of closest neighbors. K training models can be initiated in the training set, built on the distance extent technique. Determine the k value that minimizes the error and the sample length l. Meaning, the basic postulation of the KNN procedure is that a condition can be found in the previous neighborhood extra or a reduced amount comparable to that projected in the upcoming. The KNN technique does not level ancient data, particularly once there are special events; the performance of the forecast is improved than the parametric process.

4 Result and Discussion

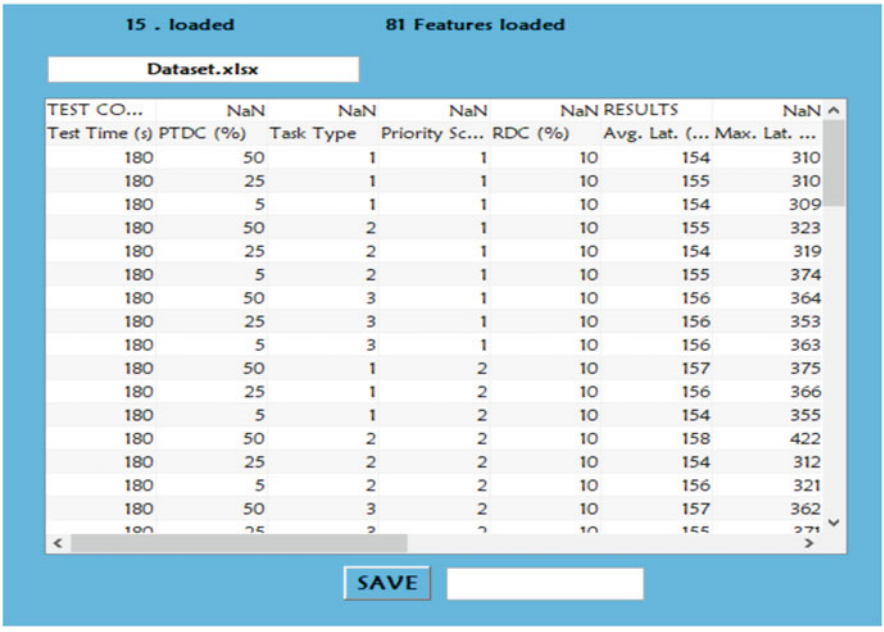
This section shows the results gotten from the implementation and the testing of the system using the k-nearest neighbor machine learning algorithm. The interfaces of the implementations are also presented in this section.

The machine-learning algorithm was implemented using MATLAB 2015 package. The datasets were inputted into the MATLAB and were then passed into the KNN classifier algorithm, and the result was obtained. The performance metrics used for the evaluation were confusion matrix, and below are some interfaces from the system implementation and testing.

Figure 1 shows the datasets used for the implementation. This particular interface in Fig. 1 is the datasets inputted into MATLAB, and the KNN classification algorithm was implemented on it.

Figure 2 shows the k-nearest neighbor scattered plot showing the classification of the datasets.

Figure 3 shows the confusion matrix for the k-nearest neighbor implemented on the system. The KNN of the system true-positive (TP) value is 35, true-negative (TN) value is 16, false-positive (FP) value is 5, and false-negative (FN) value is 4.



TEST CO...	NaN	NaN	NaN	NaN	RESULTS	NaN
Test Time (s)	PTDC (%)	Task Type	Priority Sc...	RDC (%)	Avg. Lat. (...	Max. Lat. ...
180	50	1	1	10	154	310
180	25	1	1	10	155	310
180	5	1	1	10	154	309
180	50	2	1	10	155	323
180	25	2	1	10	154	319
180	5	2	1	10	155	374
180	50	3	1	10	156	364
180	25	3	1	10	156	353
180	5	3	1	10	156	363
180	50	1	2	10	157	375
180	25	1	2	10	156	366
180	5	1	2	10	154	355
180	50	2	2	10	158	422
180	25	2	2	10	154	312
180	5	2	2	10	156	321
180	50	3	2	10	157	362

Fig. 1 Sample datasets

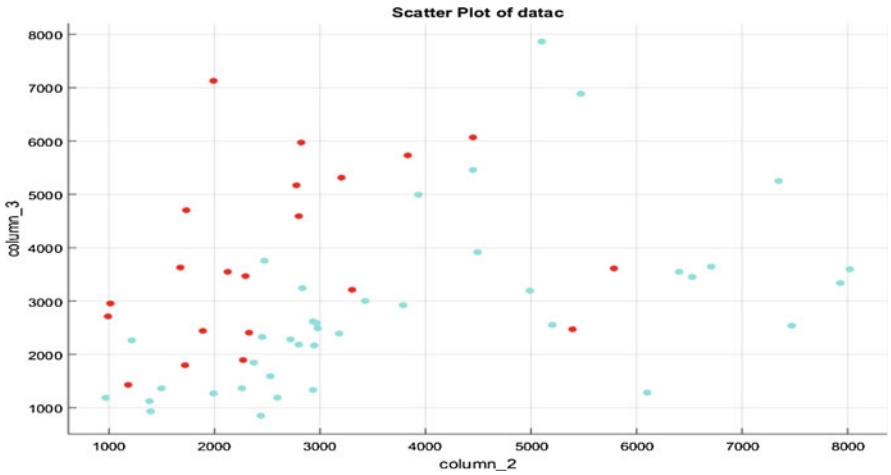


Fig. 2 Scatter plot of the dataset

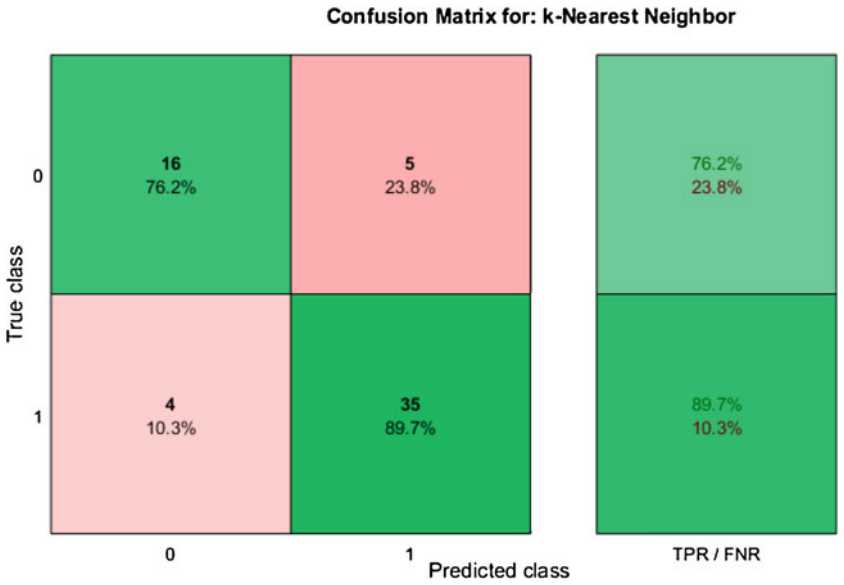


Fig. 3 K-nearest neighbor confusion matrix. Confusion matrix for KNN: TP = 35, TN = 16, FP = 5, FN = 4

4.1 System Evaluation

The developed system was appraised employing the confusion matrix metrics. The measures used in this study were sensitivity, specificity, precision, accuracy,

Table 1 Confusion matrix for proposed system evaluation

Performance metrics	Results (%)
Sensitivity	89.74
Specificity	76.19
Precision	87.50
Accuracy	85.00
F1 score	88.61
Negative predictive value	0.8000
False-positive rate	23.81
False discovery rate	12.50
False-negative rate	0.1026
Mathews correlation coefficient	66.71

Table 2 Comparative analysis with previous researches

Authors	ML methods used	Disease predicted	Accuracy
Khateeb and Usman [76]	NB, KNN, and bagging classifiers	Breast cancer	79.20%
Hashi et al. [77]	Decision tree and KNN	Diabetes	76.96%
Proposed system	KNN	COVID-19	85%

F1 score, negative predictive value, false-positive rate, false discovery rate, false-negative rate, and Matthews correlation coefficient.

Table 1 shows the confusion matrix for the proposed system evaluation, and it was revealed that the system had 88% precision, 90% sensitivity, 76% specificity, 85% accuracy, and 89% F1 score. This indicated that the system is an efficient and effective one.

4.2 System Comparison

The system was compared with previous researches that had used machine learning algorithms, as shown in Table 2 below.

4.3 Discussion

The proposed system was compared with two previous studies, and it was realized that the proposed study performance supersedes those previous works with an accuracy of 85%. Khateeb and Usman [86] employed three ML algorithms in their study and those algorithms' performance was compared, but the KNN accuracy was given as 79.20% which showed that our proposed work performs better than theirs. Hashi et al. [77] used two algorithms on the diabetes dataset, and the two accuracies were compared as well, but the KNN accuracy for the research was 76.96% which also signifies that our posed ML algorithm performs better than the

two works compared with, with an accuracy of 85%. It is finally concluded that the proposed system performed better, effectively, and efficiently in terms of accuracy being used as the performance metrics.

5 Conclusion

The implementation of ML on IoT gadgets decreases overcrowding issues in the network by permitting data to be computed near data sources, protecting data upload privacy, and decreasing energy consumption for unremitting wireless broadcasting to gateways or cloud servers. The goal of this investigation was to deliver an efficient approach to ensure the implementation of low-performance ML techniques on hardware in the IoT paradigm, allowing the means for IoT awareness. In this study, a machine learning approach using the KNN classification algorithm was developed using a process management in IoT operating systems: cross-influence between processing and communication tasks in end devices data. The result of the experiment shows an 85% accuracy which outperformed other methods that have been suggested in the literature. However, this study proves to be relevant and can be adopted for better efficiency. Future work suggests a robust dataset and combines efficient machine learning approaches such as KNN with other optimizer algorithms that can help better the technology world.

References

1. Adeniyi EA, Ogundokun RO, Awotunde JB (2021) IoMT-based wearable body sensors network healthcare monitoring system. In: IoT in healthcare and ambient assisted living. Springer, Singapore, pp 103–121
2. Merenda M, Porcaro C, Iero D (2020) Edge machine learning for AI-enabled IoT devices: a review. Sensors 20(9):2533. https://doi.org/10.3390/s20092533
3. Liu Y, Yang C, Jiang L, Xie S, Zhang Y (2019) Intelligent edge computing for IoT-based energy management in smart cities. IEEE Netw 33(2):111–117. https://doi.org/10.1109/MNET.2019.1800254
4. Odusami M, Abayomi-Alli O, Misra S, Shobayo O, Damasevicius R, Maskeliunas R (2018) Android malware detection: a survey. In: International conference on applied informatics. Springer, Cham, pp 255–266
5. Adeyinka AA, Adebiyi MO, Akande NO, Ogundokun RO, Kayode AA, Oladele TO (2019) A deep convolutional encoder-decoder architecture for retinal blood vessels segmentation. In: International conference on computational science and it applications, Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics). Springer, Cham. 11623 LNCS, pp 180–189
6. Oladele TO, Ogundokun RO, Kayode AA, Adegun AA, Adebiyi MO (2019) Application of data mining algorithms for feature selection and prediction of diabetic retinopathy. Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), International conference on computational science and it applications. Springer, Cham. 11623 LNCS, pp. 716–730

7. Ikedinachi AP, Misra S, Assibong PA, Olu-Owolabi EF, Maskeliūnas R, Damasevicius R (2019) Artificial intelligence, smart classrooms and online education in the 21st century: implications for human development. J Cases Inf Technol (JCIT) 21(3):66–79
8. Alagbe V, Popoola SI, Atayero AA, Adebisi B, Abolade RO, Misra S (2019) Artificial intelligence techniques for electrical load forecasting in smart and connected communities. In: International conference on computational science and its applications. Springer, Cham, pp 219–230
9. Xu H (2017) Machine learning based data analytics for IoT devices. Nanyang Technological University. https://doi.org/10.32657/10356/72342
10. Ieracitano C, Mammone N, Hussain A, Morabito FC (2020) A novel multi-modal machine learning based approach for automatic classification of EEG recordings in dementia. Neural Netw 123:176–190. https://doi.org/10.1016/j.neunet.2019.12.006
11. Panesar A (2021) Machine learning algorithms. Apress, Berkeley, pp 85–144. https://doi.org/10.1007/978-1-4842-6537-6_4
12. Yazici M, Basurra S, Gaber M (2018) Edge machine learning: enabling smart internet of things applications. Big Data Cogn Comput 2(3):26. https://doi.org/10.3390/bdcc2030026
13. Portal S (2018) Internet of things (IoT) connected devices installed base worldwide from 2015 to 2025. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/
14. Gubbi J et al (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Futur Gener Comput Syst 29(7):1645–1660
15. Lin J, Yu W, Zhang N, Yang X, Ge L (2017) On data integrity attacks against route guidance in transportation-based cyber-physical systems. In: Proceedings of the 14th IEEE annual conference in consumer communications and networking conference (CCNC 2017)
16. Singh D, Tripathi G, Jara AJ (2014) A survey of internet-of-things: future vision, architecture, challenges and services. In: Proceedings of 2014 IEEE world forum on internet of things (WF-IoT)
17. Chen X, Jiao L, Li W, Fu X (2016) Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans Netw 24(5):2795–2808
18. Lee I, Lee K (2015) The internet of things (IoT): applications, investments, and challenges for enterprises. Bus Horiz 58(4):431–440
19. Sha K et al (2018) On security challenges and open issues in internet of things. Futur Gener Comput Syst 83:326–337
20. Brewster T (2016) How hacked cameras are helping launch the biggest attacks the internet has ever seen. https://www.forbes.com/sites/thomasbrewster/2016/09/25/brian-krebs-overwatch-ovh-smashed-by-largest-ddos-attacks-ever/#705007235899. Sept 2016
21. Russon M-A (2016) Hackers turning millions of smart CCTV cameras into botnets for DDoS attacks. http://www.ibtimes.co.uk/hackers-turning-millions-smart-cctv-cameras-into-botnets-ddos-attacks-1525736. Accessed Sept 2016
22. Sha K, Alatrash N, Wang Z (2017) A secure and efficient framework to read isolated smart grid devices. IEEE Trans Smart Grid 8(6):2519–2531
23. Li S, Da Xu L, Zhao S (2015) The internet of things: a survey. Inf Syst Front 17(2):243–259
24. Gokhale P, Bhat O, Bhat S (2018) Introduction to IOT. Int Adv Res J Sci Eng Technol 5(1):41–44
25. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Futur Gener Comput Syst 29(7):1645–1660
26. Stergiou C, Psannis KE, Kim BG, Gupta B (2018) Secure integration of IoT and cloud computing. Futur Gener Comput Syst 78:964–975
27. Atzori L, Iera A, Morabito G (2017) Understanding the internet of things: definition, potentials, and societal role of a fast-evolving paradigm. Ad Hoc Netw 56:122–140
28. Chahal RK, Kumar N, Batra S (2020) Trust management in social internet of things: a taxonomy, open issues, and challenges. Comput Commun 150:13–46

29. Srivastava G, Parizi RM, Dehghantanha A (2020) The future of blockchain technology in healthcare internet of things security. In: Blockchain cybersecurity, trust and privacy. Springer, Cham, pp 161–184
30. Shafique MN, Khurshid MM, Rahman H, Khanna A, Gupta D (2019) The role of big data predictive analytics and radio frequency identification in the pharmaceutical industry. IEEE Access 7:9013–9021
31. Manavalan E, Jayakrishna K (2019) A review of internet of things (IoT) embedded sustainable supply chain for industry 4.0 requirements. Comput Ind Eng 127:925–953
32. Georgakopoulos D, Jayaraman PP, Fazia M, Villari M, Ranjan R (2016) Internet of things and edge cloud computing roadmap for manufacturing. IEEE Cloud Comput 3(4):66–73
33. Bilal K, Khalid O, Erbad A, Khan SU (2018) Potentials, trends, and prospects in edge technologies: fog, cloudlet, mobile edge, and micro data centers. Comput Netw 130:94–120. https://doi.org/10.1016/j.comnet.2017.10.002
34. Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: a complete survey. J Syst Archit 98:289–330. https://doi.org/10.1016/j.sysarc.2019.02.009
35. Wang Y, Meng W, Li W, Liu Z, Liu Y, Xue H (2019) Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. Concurr Comput Pract Exp 31(19). https://doi.org/10.1002/cpe.5101
36. Wang S, Zhao Y, Xu J, Yuan J, Hsu C-H (2019) Edge server placement in mobile edge computing. J Parallel Distrib Comput 127:160–168. https://doi.org/10.1016/j.jpdc.2018.06.008
37. Wang Y, Xie L, Li W, Meng W, Li J (2017) A privacy-preserving framework for collaborative intrusion detection networks through fog computing, pp 267–279. https://doi.org/10.1007/978-3-319-69471-9_20
38. Agarwal P, Alam M (2019) A lightweight deep learning model for human activity recognition on edge devices. Journal title Sensors and page 1–17
39. Makkar A (2020) Machine learning techniques. In: Machine learning in cognitive IoT. CRC Press, pp 67–85
40. Gope D, Dasika G, Mattina M (2019) Ternary hybrid neural-tree networks for highly constrained iot applications. arXiv preprint arXiv:1903.01531
41. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W (2017) A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J 4(5):1125–1142
42. Guevara JC, Torres RDS, da Fonseca NL (2020) On the classification of fog computing applications: a machine learning perspective. J Netw Comp Appl 159:102596
43. Wang J, Pan J, Esposito F, Calyam P, Yang Z, Mohapatra P (2019) Edge cloud offloading algorithms: issues, methods, and perspectives. ACM Comput Surv (CSUR) 52(1):2
44. Aazam M, Zeadally S, Harras KA (2018) Offloading in fog computing for IoT: review, enabling technologies, and research opportunities. Futur Gener Comput Syst 87:278–289
45. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor 19(3):1628–1656
46. Boukerche A, Guan S, Grande RED (2019) Sustainable offloading in mobile cloud computing: algorithmic design and implementation. ACM Comput Surv (CSUR) 52(1):11
47. Peng K, Leung VC, Xu X, Zheng L, Wang J, Huang Q (2018) A survey on mobile edge computing: focusing on service adoption and provision. Wirel Commun Mob Comput 2018., Article ID: 8267838:1–16. https://doi.org/10.1155/2018/8267838
48. Shakarami A, Shahidinejad A, Ghobaei-Arani M (2020) A review on the computation offloading approaches in mobile edge computing: a game-theoretic perspective. Softw Pract Exp 50:1719–1759
49. Cao B, Zhang L, Li Y, Feng D, Cao W (2019) Intelligent offloading in multi-access edge computing: a state-of-the-art review and framework. IEEE Commun Mag 57(3):56–62

50. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D (2017) On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun Surv Tutor 19(3) Third Quarter:1657–1681
51. Madakam S, Ramaswamy R, Tripathi S (2015) Internet of things (IoT): a literature review. J Comput Commun 3:164–173
52. Dolui K, Datta SK (2017) Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. IEEE
53. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. IEEE
54. Rodriguez-Zurrunero R, Ramiro U (2019) Dataset of process management in IoT operating systems: cross-influence between processing and communication tasks in end-devices. https://doi.org/10.17632/rxsdfg8ct9.1
55. Davis GA, Nihan NL (1991) Nonparametric regression and short-term freeway traffic forecasting. J Transp Eng 117(2):178–188
56. Mehrotra R, Sharma A (2006) Conditional resampling of hydrologic time series using multiple predictor variables: a K-nearest neighbour approach. Adv Water Resour 29(7):987–999
57. Bannayan M, Hoogenboom G (2008) Weather analogue: a tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach. Environ Model Softw 23(6):703–713
58. Mangalova E, Agafonov E (2014) Wind power forecasting using the k-nearest neighbors algorithm. Int J Forecast 30(2):402–406
59. Cai P, Wang Y, Lu G, Chen P, Ding C, Sun J (2016) A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transp Res Part C Emerg Technol 62:21–34
60. Cheng S, Lu F, Peng P, Wu S (2018) Short-term traffic forecasting: an adaptive ST-KNN model that considers spatial heterogeneity. Comput Environ Urban Syst 71:186–198
61. Martínez F, Frías MP, Pérez-Godoy MD, Rivera AJ (2018) Dealing with seasonality by narrowing the training set in time series forecasting with kNN. Expert Syst Appl 103:38–48
62. Fan GF, Guo YH, Zheng JM, Hong WC (2019) Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. Energies 12(5):916
63. Xu D, Wang Y, Peng P, Beilun S, Deng Z, Guo H (2020) Real-time road traffic state prediction based on kernel-KNN. Transportmetrica A TranspSci 16(1):104–118
64. Kück M, Freitag M (2020) Forecasting of customer demands for production planning by local k-nearest neighbor models. Int J Prod Econ 231:107837
65. Hattori K, Takahashi M (1999) A new nearest-neighbor rule in the pattern classification problem. Pattern Recogn 32(3):425–432
66. Hattori K, Takahashi M (2000) A new edited k-nearest neighbor rule in the pattern classification problem. Pattern Recogn 33(3):521–528
67. Jiang S, Pang G, Wu M, Kuang L (2012a) An improved K-nearest-neighbor algorithm for text categorization. Expert Syst Appl 39(1):1503–1509
68. Jiang JY, Tsai SC, Lee SJ (2012b) FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. Expert Syst Appl 39(3):2813–2821
69. Miao D, Duan Q, Zhang H, Jiao N (2009) Rough set based hybrid algorithm for text classification. Expert Syst Appl 36(5):9168–9174
70. Cui B, Shen HT, Shen J, Tan KL (2005, December) Exploring bit-di® erence for approximate KNN search in high-dimensional databases. In: Conferences in research and practice in information technology series, vol 39, pp 165–174
71. Tan S (2006) An effective refinement strategy for KNN text classifier. Expert Syst Appl 30(2):290–298
72. Wan CH, Lee LH, Rajkumar R, Isa D (2012) A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. Expert Syst Appl 39(15):11880–11888
73. Yoon JW, Friel N (2015) Efficient model selection for probabilistic K nearest neighbour classification. Neurocomputing 149:1098–1108

74. Zhang H, Berg AC, Maire M, Malik J (2006, June) SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 2. IEEE, pp 2126–2136
75. Zhang S, Cheng D, Deng Z, Zong M, Deng X (2018) A novel kNN algorithm with data-driven k parameter computation. Pattern Recogn Lett 109:44–54
76. Khateeb N, Usman M (2017). Efficient heart disease prediction system using K-nearest neighbor classification technique. In Proceedings of the international conference on big data and internet of thing, pp 21–26
77. Hashi EK, Zaman MSU, Hasan MR (2017) An expert clinical decision support system to predict disease using classification techniques. In: 2017 International conference on electrical, computer and communication engineering (ECCE). IEEE, pp 396–400