A new enhancement of the k-NN algorithm by Using an optimization technique

Ehsan Altayef
Cardiff University
Cardiff School of Engineering
Wolfson Centre for Magnetic
Cardiff,United Kingdom
altayefem@cardiff.ac.uk

Fatih Anayi
Cardiff University
Cardiff School of Engineering
Wolfson Centre for Magnetic
Cardiff,United Kingdom
Anayi@cardiff.ac.uk

Michael Packianather
Cardiff University
Cardiff School of Engineering
Wolfson Centre for Magnetic
Cardiff,United Kingdom
PackianatherMS@cardiff.ac.uk

***Abstract*— Of a number of ML (Machine Learning) algorithms, k-nearest neighbour (KNN) is among the most common for data classification research, and classifying diseases and faults, which is essential due to frequent alterations in the training dataset, in which it would be expensive using most methods to construct a different classifier every time this happens. Therefore, KNN can be used effectively as it does not require a residual classifier to be constructed in advance. KNN offers ease of use and can be applied across a broad variation spectrum. Here, a novel KNN classification approach is put forward using the Bayesian Optimization Algorithm (BOA) for optimisation. This paper seeks to make classification more accurate and suggest alterations of nearest neighbour K value to use information about dataset structure and the similarity measure of distance. The findings of experimental work based on the University of California Irvine (UCI) repository datasets in general shows improved performance of classifiers compared with conventional KNN and give greater reliability without a significant time cost to speed.**

Keywords— (k-nearest neighbour algorithm, Classification, Bayesian Optimization Algorithm)

I. INTRODUCTION

Here has been a slow increase in the application of methods of machine learning both for detecting faults and diagnosing disease, and this increase is mostly due to the development of more effective systems for recognising and classifying data around both faults and medical conditions. This includes for breast cancer, which is a cancer frequently diagnosed in females and which is growing more prevalent [1].

There are various terms used for Instance-Based Learning (IBL) algorithms and each is associated with the lazy-learning algorithm based on their method of delaying generalisation/training processes until the data has been classified. IBL is characterised through 3 main functions: difference or similarity; choosing a typical instance, and the classification function. Among the most widely used of these algorithms is nearest neighbour (NN) [2][3], with KNN being an IBL algorithm. KNN uses a smaller time for training compared to eager learner approaches including the Bayes net, decision tree and neural network.

In a 2016 study, high confidentiality was achieved for the dataset's data through using KNN as the primary algorithm[18].

II. RELATED WORKS

An enhancement to k-NN, called 'condensed nearest neighbour (CNN) was suggested in 1967, focusing on reducing the size of training datasets while improving classification performance [4].

Four years later, Gates put forward another novel approach, the Reduced Nearest Neighbour (RNN) rule, which built on and improved CNN. In RNN, a training instance is deleted if it shows no impact on classification findings. This algorithm performed at a higher level than CNN, although at increased computational time. RNN showed lower complexity in the classification stage and had lower storage requirements than the older algorithm [5].

In 1997, Wettschereck and Aha sought to monitor the way that k-NN behaved for a noisy instance. The findings of the experiment showed that k-NN did not have sensitivity to k selection for larger values of k, and the authors concluded that performance of the algorithm was more reliable at smaller values[6].

In 2000, Wilson and Martinez put forward 6 novel algorithms based on k-NN, with one improving on the last in sequence. The initial algorithm developed was DROP1, which enhances RNN rule. This algorithm had as its basis the rule that removal of an instance P only occurs where some neighbouring instances are in its class. In the second algorithm, DROP2 performance is improved through deletion of certain P instances, while DROP3 brought in a filter for noise prior to S-instance sorting. A noise filtering concept was introduced in DROP3 before sorting the S instances, and further enhancements were made in this series, ending with DROP6[7].

In 2002, KNN was further enhanced, making it more efficient while maintaining accuracy in classifying data. This algorithm, Improved k-NN (IKNN) has fewer iterations than classic KNN [8].

In 2002, KNN was again enhanced through 2 novel approaches to make it faster and more accurate. The first was a novel similarity matrix, named Higher-Order Bit Similarity (HOBS), which calculated similarity taking into account similarity within the more important consequent bit positions moving from the bit with the highest order, positioned on the far left. The second method put forward was "perfect centring", using the max distance matrix with perfect centring

for expansion of the neighbourhood. This maintained the target value central while maintaining the range on the right and left with an expand function [9][10].

In 2005, a rapid and less complex KNN algorithm known as Fast Condensed Nearest Neighbour rule (FCNN) was put forward. FCNN chooses points very near the decision border without relying on data order. A further, collaborative k-NN classification algorithm was proposed in the same year in order to conserve parties' privacy, using a secure framework for non-disclosure of data [11][12].

In 2009, another enhanced KNN classification algorithm was put forward. This relied on clustering of the training dataset via k-means with the centre of each cluster treated as a new training dataset. From sampling a cluster multiple times, weighted values are produced which show how significant a sample is [13].

A 2011 multi-stage algorithm known as Class Boundary Preserving (CBP) is designed to minimise the required training dataset. CBP preserves instances near class boundaries due to their usefulness in giving required knowledge for description of the fundamental data distribution [14].

A model which combined KNN with k-means clustering on the basis of term reweighting in classifying news in Indonesia was put forward 2012. The primary phases in the model included pre-processing, term weighting, document, k-means clustering, KNN classification, and then evaluating the model[15].

A model was put forward in 2013 using cloud-based secure k-NN computations using data encryption this prevents the data owner from sharing decryption/encryption keys with query users requesting data. The authors reported that the model gave good performance with encrypted data with no release of encryption/decryption key information for query users [16].

In 2014 a further encrypted data secure classifier was put forward, which also used the cloud [17].

The Character Frequency-Direct Word Frequency (CF-DWF) similarity formula was proposed in 2016, focusing on computation of similarity within string data and simplifying computation of similarity/distance in KNN. This approach is limited to use with string data [18].

In the same year, 2 predictive models were developed to provide strength of analysis in data related to insurance and banking. The models used Naïve Bayes and k-NN, and were compared to assess how accurate the predictions of each model were [19].

In 2020 a new model combining k-NN and k-means clustering was developed, known as Fast Hybrid Classification. This approach involves grouping training data to form multiple TLDS (Two Level Data Structure) prototypes [20].

Also in 2020, a semantic k-NN (Sk-NN) ML algorithm was developed which sought to provide solutions to the limits of classic k-NN, through leveraging semantic itemisation and a bigram model for filtering of training datasets based on related information from test datasets [2]. Figure (1) summarises related research to illustrate the various proposals to improve classic k-NN over time. It is seen from this that the majority of proposed improvements to k-NN target reductions in storage space requirements for data through removal of parts of training datasets. However, this has faced challenges, among which: removing some of a dataset can negatively affect decision borders when classifying data; and that in order to classify the data, the algorithm might require data from removed sections of the dataset (shown as issue 1 and 2 in the figure below).

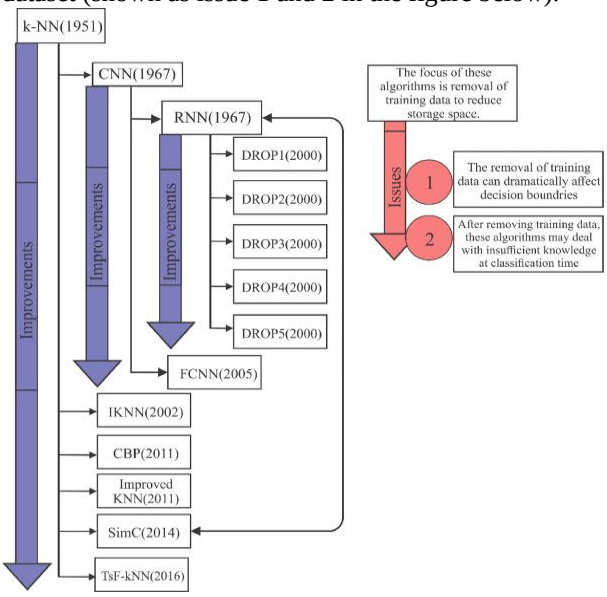


Figure (1); various enhancements of classic k-NN over time.

III. REASONS FOR SELECTING K-NN

There were various factors in the decision to preferentially use k-NN to predict labels for unclassified file data using file attributes:

- It is generally stated that the k-NN algorithm provides a comparatively effective solution in problems of "instance multi-class" [21].
- Previous literature shows extensive discussion of k-NN in which it is described as simple to implement, highly efficient [22], accurate and simple to realize [23].
- k-NN is frequently selected as simple, effective and suitable where training pattern numbers are limited [24].
- k-NN is a well-established algorithm and considered state-of-the-art in classifying English texts [25].
- k-NN shows effectiveness in machine learning for the solution of traditional problems of classification [21].
- The k-NN algorithm is usually favoured particularly where there is a requirement for a simple and accurate method [26].
- k-NN is the most suitable classifier for problems relying on semantic features [27].

A. Test dataset

In the current paper, the developed algorithm was tested by applying it to a dataset with both an encrypted requirement and no encrypted requirement. In this case, a health-based

dataset related to cancer was used, sourced from the UCI repository, which houses datasets which are frequently utilised across a broad range of research [28][29][30][31][32]. The datasets were set as critical categories, meaning that data breaches in such data could damage the organisation or individual reputationally, with negative impacts on functional operations [33].

B. Fundamental information in datasets

The dataset contains 10 quantitative predictors, with one binary dependent variable which shows whether breast cancer is present or absent. Each predictor is an anthropometric data parameter able to be collected during a routine blood test. A model for prediction on the basis of these parameters which showed sufficient accuracy would have potential uses as a breast cancer biomarker.
Quantitative Attributes:
Age (years)
BMI (kg/m2)
Glucose (mg/dL)
Insulin (µU/mL)
HOMA
Leptin (ng/mL)
Adiponectin (µg/mL)
Resisting (ng/mL)
MCP-1(pg/dL)
Labels:
1=Healthy controls
2=Patients

C. Developing the algorithm:

The findings need to reflect the quality of the novel combined algorithm and record its efficiency in comparison with older algorithms, as well as demonstrating how far it is effective and feasible. This represents the piloting of an innovative approach in leveraging a hybrid algorithm using AI. The study contributes positively in terms of methodologies in classification research in algorithms and in addition offers a decisions tool, as well as providing firms with management reference. The developed data classifier is based on the k-NN algorithm, and algorithms were developed with the aim of predicting data classes on the basis of specific security policies, and thus enhancing performance in comparison to classic k-NN. The major areas for enhancement are time taken for computation, accurate classifications and decrease in overloading in training data.
The findings of the experiment reveal that k-NN slows down the testing stage, with limitations of complexity leading data to be wrongly classified. Moreover, it was shown that k-NN was more complex if training with various types of training dataset. The complex computations when using k-NN are due to its lack of a filtering stage, which means that it takes into account every training dataset in every category when determining labels for test data.

IV. Comparing Classification Algorithms

Five algorithms used in classifying data are considered here: K- Nearest Neighbours (K-NN), Naive Bayes, support vector machine (SVM), alinear discriminant algorithms, and decision tree. Comparing these classifiers allows their power and accuracy to be assessed, including how efficiently they perform and how much time is needed. This comparison is used to validate the proposed model. Firstly, these re-existing algorithms are compared, to assess how accurate they are, for subsequent comparison to the algorithm put forward here. This was achieved via Classification Learner within MATLAB.

This application is used to train a model for data classification and allows different classification models to be used to investigate supervised machine learning. In addition, it is used for data exploration, feature selection, specification of validation methods, model training, findings assessment and automated training in finding an optimal classifier model. It can also carry out supervised machine learning through provision of known datasets as inputs, which may be example data or observation data, for instance, along with associated responses including class or label data. This data is applied in training models in prediction of responses to further data. Models can then be fed new data. The app can be used for investigating programmatic classification. Also, models can be exported into a workspace, and MATLAB® code can be produced for recreation of trained models.

Figure (2) provides a flowchart illustrating typical workflows in training a classifier within Classification Learner.

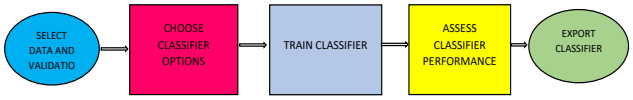


Figure (2): typical workflow to train a classifier within Classification Learner.

Classification Learner shows results from the validated model using diagnostic tools including scatterplots, accuracy measures, confusion matrices and ROC curves. Automatic training can be applied for one or multiple classification algorithms, allowing comparison of validation results, and selection of the classifier type which best fits the particular problem (including KNN, SVM, Naïve Bayes, linear discriminant models, and decision tree).

Based on the above Classification Learner was selected for use in comparing the algorithms, and the accuracy findings are presented in separate sections and figures for the different algorithms, using the ROC Curve and the Parallel Coordinates Plot Figures 3, 4, 5, 6 and 7.

Implementation and testing through the app was done using datasets from UCI's Machine Learning Repository, comparing the above-mentioned algorithms. Based on the findings, KNN gave the most accurate predictions (see Figure 8).

1 – Decision Trees:
Accuracy (68.1%), Total misclassification cost 37, Prediction speed ~10000 obs/sec, Training time 0.43094 sec.

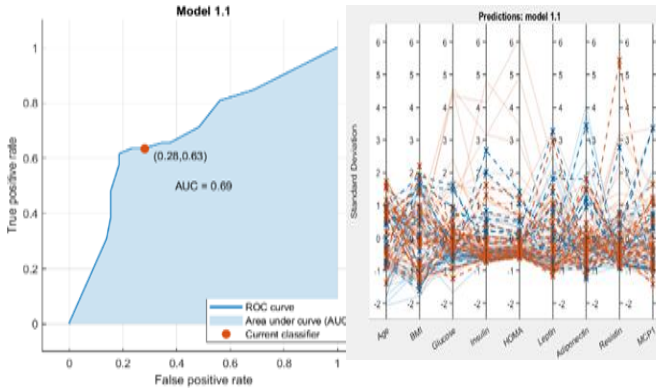


Figure (3): Decision Trees ROC Curve and Parallel Coordinates Plot.

2 – Linear Discriminant:

Accuracy (70.7%), Total misclassification cost 34, Prediction speed ~7200 obs/sec, Training time 0.62525 sec.

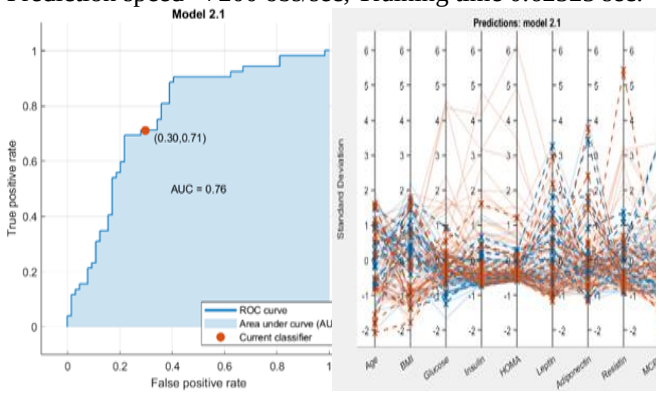


Figure (4): Linear Discriminant ROC Curve and Parallel Coordinates Plot.

3 – Naive Bayes:

Accuracy (61.2%), Total misclassification cost 45, Prediction speed ~7100 obs/sec, Training time 0.58219 sec.

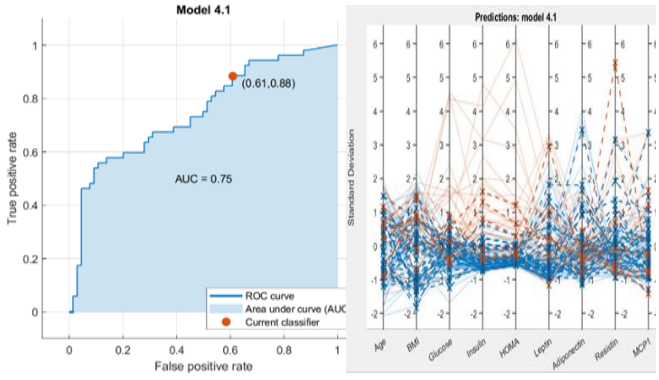


Figure (5): Naive Bayes ROC Curve and Parallel Coordinates Plot.

4 – Support Vector Machine (SVM):

Accuracy (64.7%), Total misclassification cost 42, Prediction speed ~8200 obs/sec, Training time 0.18234 sec.

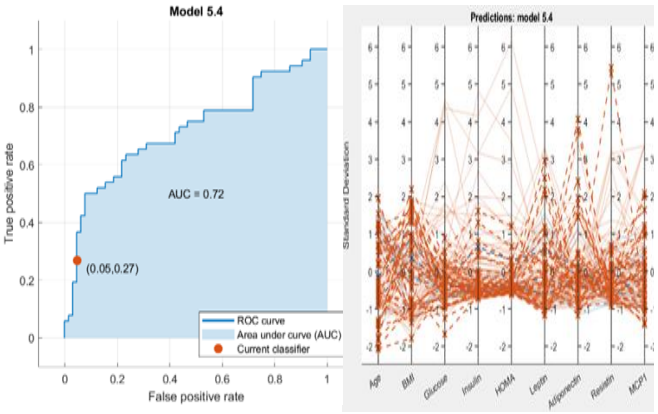


Figure (6): Support Vector Machine (SVM) ROC Curve and Parallel Coordinates Plot.

5 – K-Nearest Neighbours(K-NN):

Accuracy (71.6%), Total misclassification cost 33, Prediction speed ~5300 obs/sec, Training time 0.2025 sec.

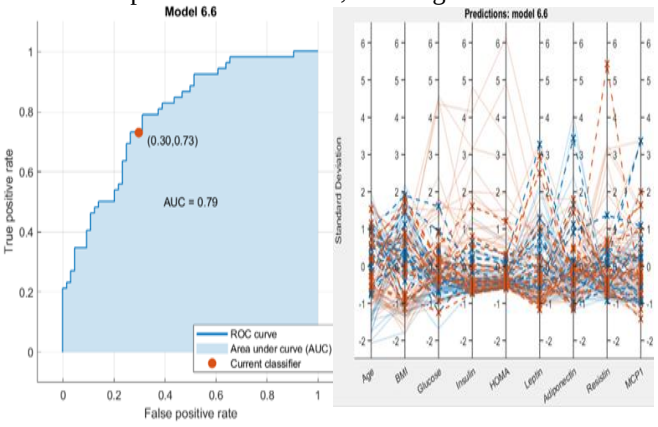


Figure (7): K-Nearest Neighbors(K-NN) ROC Curve and Parallel Coordinates Plot.

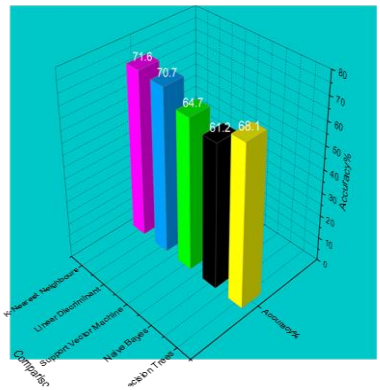


Figure (8); Comparison of (KNN, SVM, Naïve Bayes, linear discriminant).

With the KNN an accuracy rate of 71.6% for the "Euclidean" and "Cosine" distances with the same number of neighbours k = 6 was developed from the results of the previous figures.

1) KNN algorithm:

The supervised KNN classifier has been implemented in various applications, as a simple and effective algorithm. Cutting edge work reveals that KNN could potentially be used in distortion mitigation. In general, this is implemented

offline, and not including any real application. Thus proposed models should involve the algorithm being applied in real time[34][35]. With experimental datasets, KNN searches for k samples in the closest samples. This approach selects a new class to be placed in, and new instances are classified within the class with the greatest number of votes within K nearest neighbors [36][37].

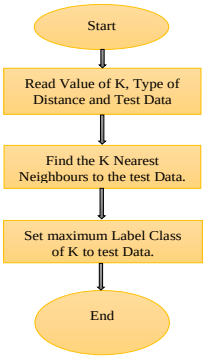


Figure (9): K-Nearest Neighbours (K-NN) flowchart.

1) Bayesian Optimization Algorithm (BOA):

The optimization of hyperparameters is seen as a black box type of problem. Black box outputs are used for assessing findings from various viable configurations of parameters in order to show how the model performs in generalization. Optimization of hyper-parameters is provided by:

$$X^* = arg_{X \in U} \max f\ (X) \qquad (1)$$

In which X* represents the optimal parameter set and U represents candidate sets.

Essentially, BOA initially assumes f(X) as the prior distribution, before applying later information for continual optimization of the guess model until this model reflects real distribution [38][39][40][41].

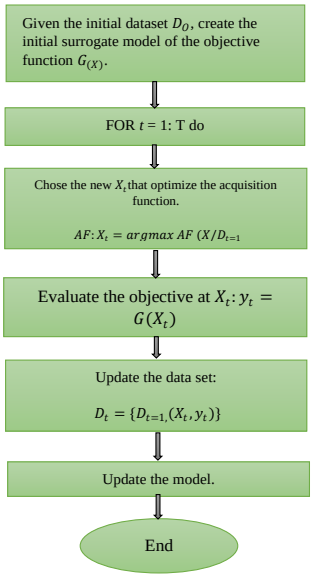


Figure (10): Bayesian Optimization Algorithm flowchart.

V. THE PROPOSED MODEL:

The model put forward in this paper applies an optimization method based on BOA in developing the KNN algorithm. Figure (11) gives an overview of KNN with BOA to represent its optimization approach for two KNN parameters, which includes the measure of similarity, called Distance, and the K value to tune parameters through application of BOA, a method with high accuracy and efficiency in comparison with other optimization approaches. This optimization approach is based on 10 techniques to assess Distance, namely: Spearman, seuclidean, Minkowski, mahala Nobis, Jaccard, hamming, Euclidean, cosine, correlation, Chebyshev and cityblock, with the best technique selected based on the type of data. The k value as the nearest neighbor number is also subject to Bayesian optimization in the model, and this is done through trying K values of 1-30 to select the optimal value. Cross-validation is performed for the dataset via the hold-out technique at a 70%/30% percentage, with selection of 70% of data and the associated labels, with the other 30% applied to validate the model following training. The final stage involves evaluation based on a metric for accuracy which uses a confusion matrix which uses the correct test data labels alongside predictions for labels based on trained data. The algorithm represents enhanced KNN.

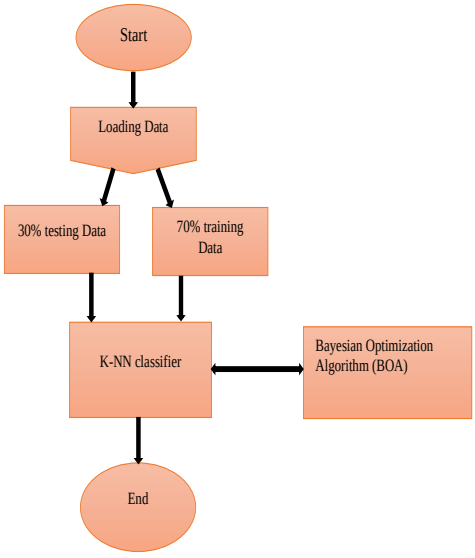


Figure (11); Flowchart for the proposed model (KNN with BOA).

1) Processes in the model:

-The primary dataset, containing labels and features, is loaded.

-his dataset is divided to form a training set and a testing set, with a 70:30 split.

- The KNN classifier is trained.

- Application of optimisation for selection of the optimal k value from 10 to 30.

- Selection of the optimal distance function for the KNN classifier.

- the test data are validated by applying the trained KNN model.

- Confusion matrix is plotted, to validate the model and measure accuracy.

VI. Results and Discussion:

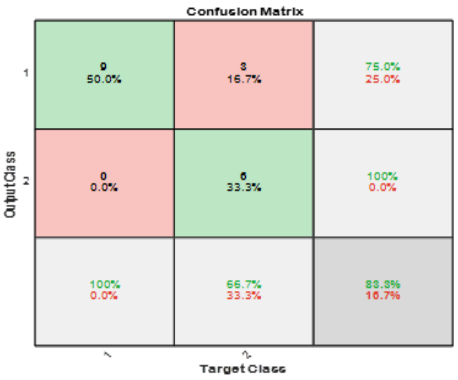
A. *Performance tests of the proposed algorithm using selected UCI datasets:*

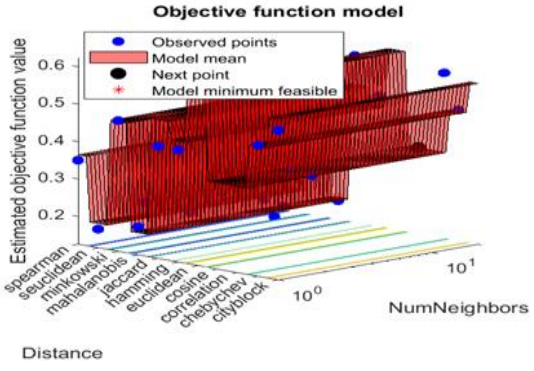
Here, the findings of performance testing of KNN with BOA on the chosen dataset from UCI are presented. Measurements are made for accuracy as a classifier, as well as time, using the nearest neighbour k value and distance as hyperparameters. Comparisons are then made to a range of classifying algorithms, including SVM, linear discrimination, Naïve Bayes and KNN to consider how powerful and how efficient the proposed algorithm is. Iin testing KNN with BOA, optimisation was conducted for the parameters to identify optimal Distance function and K value.

As part of these tests, the confusion matrix, objective function and min objective against number of function evaluations are plotted for validation of classification predictions and for accuracy. The confusion or error matrix is a table format which is used for visualising the way an algorithm performs and showing how accurately it performs. Here, correctly classified data is listed as a 'true' negative or positive, while incorrectly classified data is a 'false' negative or positive. Additional measures which can be derived include positive predictive value, specificity and sensitivity, for example, and these parameters are especially significant for one-test diagnostic testing to find whether an illness or state is present or absent.

The objective function model provides neighbour numbers and distance used as well as estimation of a value for objective function.

For min objective vs. number of function evaluation, the latter is given by the objective function's number of iterations, e.g. 1st, 2nd, 3rd iteration etc. min objective represents the minimum objective function value for the particular iteration.





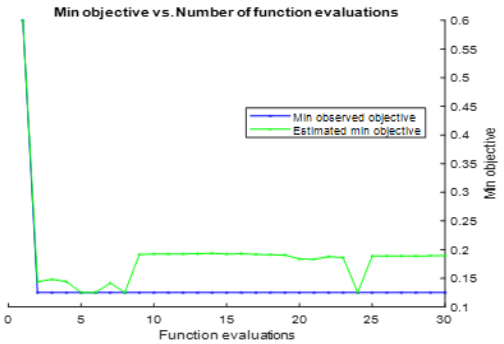


Figure (11): confusion matrix, objective function model, min objective vs. number of function evaluation findings for KNN with BOA.

The following list outlines the different types of graphics published in CPSS Transactions. They are categorized based on their construction, and use of color / shades of gray:

B. *Comparing KNN with BOA and different algorithms:*

The way in which KNN with BOA performed was then compared with other algorithms based on percentage accuracy of the classifier when used with the UCI dataset. Figure (12) gives results for each algorithm, and illustrates the superior accuracy achieved by the developed algorithm on this dataset compared to the other classifiers. While KNN with BOA performs more well in terms of classification accuracy for the breast cancer dataset, at 83.3%: however, overall, the performance exceeded other algorithms, and the findings demonstrated the capacity of the proposed algorithm to be more accurate as a classifier. Thus, KNN with BOA is shown to be effective in dealing with the classification data, performing as well or better than different classifiers.

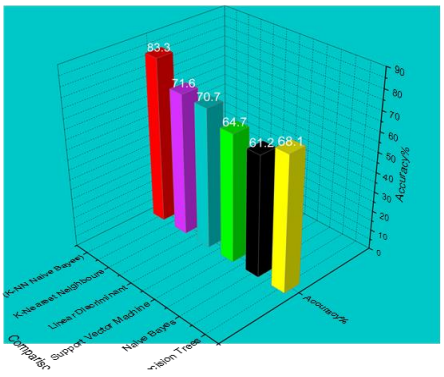


Figure (12): Comparison of KNN with BOA with other algorithms

All input vectors were tested separately for different neighbours and distance types. Based on the results obtained, the best classification accuracy rate is selected for each input vector. It is clear that the precision rate is influenced by the optimization of the parameters, which are both the type and the number of neighbours. Relatively few neighbours tend to stabilize at around 28%. For each input vector, subtract the best precision factor from the number of neighbors. The results are shown in Table 1.

Table 1KNN classifier for all input vectors.

Input vector	Vector1	Vector2	Vector3	Vector4	Vector5	Vector6	Vector7	Vector8	Vector9	Vector10
Best distance type	Spearman	seuclidean	Minkowski	mahala Nobis	Jaccard	Euclidean	cosine	correlation	Chebyshev	cityblock
Accuracy rate (%)	78.3	69.3	54.7	27.9	31.8	81.6	83.3	79.9	63.5	63.9
Best K	14	6	10	8	11	29	6	15	11	7

VII. Conclusion

This paper has introduced KNN with BOA as a good classification method using the Bayesian Optimization Algorithm to improve hyperparameters. In adaption of KNN with BOA to a classification problem, this work has discussed the stages which follow: pre-processing data, evaluation/fitness and optimization of distance and K value by BOA to make classification more accurate and time-efficient. Moreover, the method put forward here was tested using a benchmarking dataset from UCI related to breast cancer. A comparison has been made between this and previous algorithms, and it is found that KNN with BOA can achieve optimal results for the studied dataset. In terms of the hyperparameter classification task the results demonstrate that the proposed algorithm performs competitively compared to establish classifiers. Moreover, based on the findings KNN with BOA shows both accuracy and utility in classification tasks, and can produce more accurate classifications.

Further tests of the algorithm developed are needed using other datasets such as UCI benchmarks, and higher-complexity datasets. Future work will also focus on making classification more accurate and less time-consuming for the most effective predictions of unseen data, using hybrid methods for greater accuracy in applying the algorithm to big data.

Acknowledgment

THE AUTHORS THANKS WOLFSON CENTRE FOR MAGNETIC AND CARDIFF UNIVERSITY FOR THEIR SUPPORT.

References

[1] S. Şahan, K. Polat, H. Kodaz, and S. Güneş, "A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis," Comput. Biol. Med., vol. 37, no. 3, pp. 415–423, 2007.

[2] M. Ali, L. T. Jung, A. H. Abdel-Aty, M. Y. Abubakar, M. Elhoseny, and I. Ali, "Semantic-k-NN algorithm: An enhanced version of traditional k-NN algorithm," Expert Syst. Appl., vol. 151, p. 113374, 2020.

[3] D. Mena-Torres, J. Aguilar-Ruiz, and Y. Rodriguez, "An Instance based Learning Model for Classification in Data Streams with Concept Change," Proc. Spec. Sess. - Revis. Pap. 11th Mex. Int. Conf. Artif. Intell. 2012 Adv. Artif. Intell. Appl. MICAI 2012, pp. 58–62, 2012.

[4] P. H.-I. transactions on information theory and U. 1968, "The condensed nearest neighbor rule (Corresp.)," Citeseer, pp. 515–516, 1967.

[5] G. W. Gates, "The Reduced Nearest Neighbor Rule," IEEE Trans. Inf. Theory, vol. 18, no. 3, pp. 431–433, 1972.

[6] D. Wettschereck, D. W. Aha, and T. Mohri, "A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms," Artif. Intell. Rev., vol. 11, no. 1–5, pp. 273–314, 1997.

[7] W. J. et al., "Single-arm study of brentuximab vedotin in patients with relapsed or refractory Hodgkin lymphoma who are ineligible for stem cell transplantation or multiagent chemotherapy," Haematologica, vol. 101, p. 56, 2016.

[8] Y. Wu, K. Ianakiev, and V. Govindaraju, "Improved k-nearest neighbor classification," Pattern Recognit., vol. 35, no. 10, pp. 2311–2318, 2002.

[9] Q. Ding, M. Khan, A. Roy, and W. Perrizo, "T h e P - t r e e A l g e b r a 1 ' =," pp. 426–431, 2002.

[10] M. Leman, Lecture notes in artificial intelligence, vol. 1317. 2002.

[11] F. Angiulli, "Fast condensed nearest neighbor rule," ACM, no. the 22nd international conference on Machine learning, p. 25, 2005.

[12] J. Zhan, L. W. Chang, and S. Matwin, "Privacy preserving K-nearest neighbor classification," Int. J. Netw. Secur., vol. 1, no. 1, pp. 46–51, 2005.

[13] X. Li, "An Improved KNN Text Classification Algorithm Based on Clustering," pp. 436–439, 2010.

[14] K. Nikolaidis, J. Y. Goulermas, and Q. H. Wu, "A class boundary preserving algorithm for data condensation," Pattern Recognit., vol. 44, no. 3, pp. 704–715, 2011.

[15] P. WiraBuana, S. Jannet D.R.M., and I. Ketut Gede Darma Putra, "Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News," Int. J. Comput. Appl., vol. 50, no. 11, pp. 37–42, 2012.

[16] Y. Zhu, R. Xu, and T. Takagi, "Secure k-NN computation on encrypted cloud data without sharing key with query users," Cloud Comput. 2013 - Proc. 2013 Int. Work. Secur. Cloud Comput., pp. 55–60, 2013.

[17] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "K-nearest neighbor classification over semantically secure encrypted relational data," IEEE Trans. Knowl. Data Eng., vol. 27, no. 5, pp. 1261–1273, 2015.

[18] M. A. Zardari and L. T. Jung, "Data classification with k-NN using novel character frequency-direct word frequency (CF-DWF) similarity formula," 2015 Int. Symp. Math. Sci. Comput. Res. iSMSC 2015 - Proc., pp. 280–285, 2016.

[19] G. Rahangdale, M. Ahirwar, and M. Motwani, "Application of k-NN and Naïve Bayes Algorithm in Banking and Insurance Domain," Int. J. Comput. Sci. Issues, vol. 13, no. 5, pp. 69–75, 2016.

[20] N. Septiana and N. Suciati, "Combination of fast hybrid classification and K value optimization in K-NN for video face recognition," Regist. J. Ilm. Teknol. Sist. Inf., vol. 6, no. 1, pp. 65–73, 2020.

[21] T. H. Chiang, H. Y. Lo, and S. De Lin, "A ranking-based KNN approach for multi-label classification," J. Mach. Learn. Res., vol. 25, pp. 81–96, 2012.

[22] Y. Tan, "An Improved KNN Text Classification Algorithm Based on K-Medoids and Rough Set," Proc. - 2018 10th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2018, vol. 1, no. 1, pp. 109–113, 2018.

[23] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, "KNN based machine learning approach for text and document mining," Int. J. Database Theory Appl., vol. 7, no. 1, pp. 61–70, 2014.

[24] J. He, A.-H. Tan, and C.-L. Tan, "A Comparative Study on Chinese Text Categorization Methods," PRICAI 2000 Work. Text Web Min., no. August, pp. 24–35, 2000.

[25] Y. Yang and X. Liu, "A re-examination of text categorization methods," Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR 1999, pp. 42–49, 1999.

[26] M. Khan, Q. Ding, and W. Perrizo, "K-nearst Neighbor Classification on spatial Data Streams Using P-trees," Springer, Berlin, Heidelb., vol. 2336, no. 978-3-540-47887–4, pp. 978-3-540-43704–8, 2002.

[27] D. I. Ignatov and S. O. Kuznetsov, "Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia," no. May, 2012.

[28] N. García-Pedrajas and D. Ortiz-Boyer, “Boosting k-nearest neighbor classifier by means of input space projection,” Expert Syst. Appl., vol. 36, no. 7, pp. 10570–10582, 2009.

[29] H. Parvin, H. Alizadeh, and B. Minaei-bidgoli, “MKNN : Modified K-Nearest Neighbor,” Proc. World Congr. Eng. Comput. Sci. WCECS, pp. 22–25, 2008.

[30] D. Mena-Torres and J. S. Aguilar-Ruiz, “A similarity-based approach for data stream classification,” Expert Syst. Appl., vol. 41, no. 9, pp. 4224–4234, 2014.

[31] Z. Voulgaris and G. D. Magoulas, “Extensions of the k nearest neighbour methods for classification problems,” Proc. IASTED Int. Conf. Artif. Intell. Appl. AIA 2008, pp. 23–28, 2008.

[32] C. Ecient and S. Ruggieri, “Efficient C4. 5,” Chem. …, vol. 14, no. 2, pp. 438–444, 2000.

[33] S. Strauch, U. Breitenbuecher, O. Kopp, F. Leymann, and T. Unger, “Cloud data patterns for confidentiality,” CLOSER 2012 - Proc. 2nd Int. Conf. Cloud Comput. Serv. Sci., pp. 387–394, 2012.

[34] D. Marquez-Viloria, L. Castano-Londono, and N. Guerrero-Gonzalez, “A modified knn algorithm for high-performance computing on fpga of real-time m-qam demodulators,” Electron., vol. 10, no. 5, pp. 1–14, 2021.

[35] L. Sun, J. Du, and Z. He, “Machine learning for nonlinearity mitigation in CAP modulated optical interconnect system by using K-nearest neighbour algorithm,” Opt. InfoBase Conf. Pap., vol. 2, pp. 25–27, 2016.

[36] M. J. Al Dujaili, A. Ebrahimi-Moghadam, and A. Fatlawi, “Speech emotion recognition based on SVM and KNN classifications fusion,” Int. J. Electr. Comput. Eng., vol. 11, no. 2, pp. 1259–1264, 2021.

[37] J. Zhou, Y. Yang, P. Chen, and G. Wang, “"Speech emotion recognition based on rough set and SVM,” Training, 2006.

[38] M. Pelikan, D. Goldberg, and E. Cantu-Paz, “BOA: The Bayesian Optimization Algorithm,” Urbana, Univ. Illinois Urbana-Champaign, 2005.

[39] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” Adv. Neural Inf. Process. Syst., vol. 4, pp. 2951–2959, 2012.

[40] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” Proc. IEEE, vol. 104, no. 1, pp. 148–175, 2016.

[41] F. He, J. Zhou, Z. kai Feng, G. Liu, and Y. Yang, “A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm,” Appl. Energy, vol. 237, no. January, pp. 103–116, 2019.