Predicting Medical Conditions Using *k*-Nearest Neighbors

Johann Sun[1], Kaylee Hall[1], Andrew Chang[1], Jessica Li[1], Connor Song[1], Apoorva Chauhan[2], Michael Ferra[3], Theresa Sager[3], Shahab Tayeb[4]
[1]UNLV STEM, [2]AEOP UNITE, [3]RET, [4]UNLV
Las Vegas, Nevada

***Abstract* — As the healthcare industry becomes more reliant upon electronic records, the amount of medical data available for analysis increases exponentially. While this information contains valuable statistics, the sheer volume makes it difficult to analyze without efficient algorithms. By using machine learning to classify medical data, diagnoses can become more efficient, accurate, and accessible for the general public. After choosing k-Nearest Neighbors for its simplicity, we applied it to datasets compiled by the University of California, Irvine Machine Learning Repository to diagnose two conditions -- chronic kidney failure and heart disease -- with an accuracy of approximately 90%. In the future, similar methods can be used on a larger scale to bring ease of use to the field of medical diagnostics**.

Keywords — Data mining; Disease diagnoses; k-Nearest Neighbors; k-NN classification

I. Introduction

In the medical field, data mining is used to heighten efficiency and accuracy of medical diagnoses by gathering a large amount of clinical data into an easily processable form. By exploring the applications of data-mining techniques, doctors receive the benefits of a statistical perspective of their data, leading to more accurate decisions concerning the health of their patients. This data then becomes evidence doctors and patients utilize to access medical data in a comparative format, increasing accuracy and efficiency of medical diagnosis. By providing algorithm-assisted insight, individuals can begin preventative measures early, decreasing the 39.5 million global deaths in 2015 caused by preventable fatal conditions such as cardiovascular diseases, cancers, diabetes, and chronic lung diseases [1].

The motivation behind our research is to provide a quick and efficient way to diagnose an individual with minimal information and maximum accuracy. The algorithm synthesizes biometric data along with other patient attributes received in lab reports to warn patients and notify them to see their doctors. Such a method could also be applied by professionals to speed up the diagnosis and treatment processes. Furthermore, this method simplifies the diagnostic process for the average patient. Not everyone understands all the nuances of their medical reports and may not realize they are in danger until their next doctor's appointment. The convenience of the algorithm allows users to receive their diagnostics within the comfort of their homes and may even convince many self-confident users to stay up to date on their body conditions. By developing an algorithm, the average individual can be alerted about potential threats to their health without waiting for the next available spot in their doctor's schedule. This can result in faster treatment, lower medication costs, and prevent fatalities.

There is a plethora of other research involving the use of artificial intelligence to improve the efficiency and accuracy of medical care and diagnoses. The mRhythm study, consisting of medical researchers from University of California, San Francisco and software engineers from Cardiogram, are developing an algorithm that can detect heart arrhythmia through semi-supervised deep learning [2]. They use data from patients with FDA-approved monitors like AliveCor and collect data from their iOS and Android Cardiogram app to develop the algorithm. Studies done at Lumiata, an Intel-backed company, are also using deep learning to analyze health data and create models to supplement medical diagnosis [3].

A more specific study that was conducted used a multiple kernel learning algorithm to classify wrist blood pulse signals to diagnose individuals [4]. An automatic diagnostics system is noted in a data-mining project pursuing similar topics. It proposes computer-based information and/or decision-support systems that can aid in achieving clinical tests at a reduced cost. [5] aims to analyze the different predictive/descriptive data mining techniques proposed in recent years for the diagnosis of heart disease. Another study applied a fuzzy *k*-NN algorithm to diagnose Parkinson's disease more effectively [6].

To diagnose patients as being vulnerable to certain conditions, they must have symptoms that can be associated with that condition. Thus, the chosen algorithm must be able to classify a patient based on their measured medical data. After taking in biometric data such as heart rate and blood pressure, the algorithm compares this data to others already diagnosed with various medical conditions and decides whether or not the patient is similar enough to be diagnosed. Our hypothesis is that a modified yet unspecified k-NN algorithm will be able to accurately diagnose patients for multiple medical conditions without significant change to the original method using training data sets for said health conditions.

II. Literature Review

k-NN is commonly used as a simple and high accuracy classifier in the medical field. One such study utilized the algorithm to improve brain-computer interface technology [7]. Another study proposed a faster *k*-NN algorithm where the *k* value and the training-data prototypes are modified by a trained observer to maximize the accuracy for MRI data [8].

Multiple studies used *k*-NN to diagnose diseases and analyze results [9][10]. A study performed by [9] tested the reliability of seven different algorithms when diagnosing

ischemic heart disease. Using a fuzzy *k*-NN algorithm with an artificial immune system, another study diagnosed breast cancer [10]. Similarly, [11] improves the accuracy of *k*-NN in diagnosing breast cancer by applying a genetic algorithm (GA) that determines the best components for the *k*-NN algorithm. Implementing the GA algorithm with *k*-NN resulted in a 3% increase in the average accuracy of breast cancer prognosis.

Others have also used *k*-NN to classify locations, as Huang et al. did to predict where proteins are at the subcellular level [12]. Moreover, *k*-NN can be used to classify different motions, as demonstrated by a study comparing *k*-NN to the quadratic and linear discriminant analyses to classify wrist motions based on electromyogram signals [13].

A. *Papers Using Similar Data*

Shouman et al. proposed the use of *k*-NN to help medical professionals diagnose heart disease [14]. They used 13 of the 76 raw attributes in the benchmark dataset from the Cleveland Clinic Foundation (CCF) in the UCI Repository. Applying *k*-NN on this benchmark dataset achieved an accuracy of 97.4%, demonstrating a higher accuracy than any other published findings on that dataset [14]. Similarly, Kalaiselvi used an average *k*-NN algorithm utilizing the dataset from the CCF to diagnose heart disease [5]. Using 12 of the 76 raw attributes to predict and classify heart disease, he concluded that the accuracy of the average *k*-NN algorithm is higher than any other classification techniques such as Naive Bayes or Decision Trees [15].

We also referenced the UCI chronic kidney disease dataset in our research. A study performed by Ani et al. compared different classification approaches for the prediction of chronic renal failure [16]. The algorithm that demonstrated the most accurate results was used to develop a clinical decision support system. They used the dataset from the UCI repository which contains two attributes and 400 instances. When the random subspace classification method was applied to *k*-NN, it proved more accurate than other classifiers in predicting the disease. Following *k*-NN, ANN was 81%, Naive Bayes 78%, and LDA 76% accurate [16]. Another study conducted by Chihk et al. also used this dataset and tested the accuracy of a fuzzy *k*-NN method, which had an 89% accuracy rate [17].

B. *Preliminaries*

Several related papers include the use of comparable functions in implementations of *k*-NN in research. A similar study involved facial recognition, using the implementation of an altered *k*-NN method with a linear data structure [4]. They used this new LLK method, which represents the test sample as a linear combination of all the training samples, to test if the corresponding coefficients are nonzero. If the training samples are in the same class as the test sample, they are nonzero. Otherwise, the training sample would be labeled as zero.

To deal with issues resulting from disparity within samples involving the nonzero coefficient and inability of the residual classifier to use the Naive Bayes decision rule, the study implemented a new LLK method as follows:

$$\min\|x - Bv\|^2 + \lambda\|v\|1 + \alpha\|v - \beta d\|^2$$

Their use of Euclidean distance differs from the general *k*-NN method as well as our method due to the inclusion of a decay speed of distance measure [4]:

$$d_i = exp\left\{-\frac{1}{2\sigma^2}\|x - b_i\|^2\right\}$$

General Performance Enhancement techniques utilize the coefficients' truncating method, where only the k largest coefficients of each class are required for classification [4]:

$$c^* = \arg\max_c \sum_{\substack{(b_i \epsilon B_e)^\wedge \\ (v_i \epsilon T(k))}} v_i$$

$$c^* = \arg\min_c \left\| x - \sum_{\substack{(b_i \epsilon B_e)^\wedge \\ (v_i \epsilon T(k))}} v_i b_i \right\|_2^2$$

The algorithms above aim to standardize and more efficiently carry out the *k*-NN method while adding novel methods of implementation. The fundamental basis of these methods mirror the method detailed in this paper. Further applications of our current method for visual diagnostics could refer to these related works as pointers, with only minor changes in style and input/output.

Similarly, future diagnostics could also implement training data coefficient reconstruction. Medical conditions featuring less proximity between training points cause a high degree of distance errors, which will need reconstruction of training data points after running the algorithm. This can be done with the square loss function [18]:

$$\min_W \sum_{i=1}^{n} \|Xw_i - x_i\|_2^2 = \min_W \|XW - X\|_F^2$$

III. Proposed Approach

A. *Motivation*

This method highlights the use of artificial intelligence in enhancing medical diagnostics and preventative care measures across an extensive number of health disorders. With input as a collection of patient-recorded data, the method implements the *k*-NN supervised learning algorithm to predict whether a patient has a certain medical condition. Rather than focusing on a specific condition, this approach relies on the accuracy of a variety of training datasets covering a wide range of ailments. This allows the *k*-NN algorithm to deliver high levels of accuracy while theoretically covering a near infinite amount of ailments.

B. *k-Nearest Neighbors*

The *k*-NN algorithm is a method for classifying objects based on the proximity of objects in the training set. The algorithm takes in a patient's data and compares it with a training table of patients with various medical conditions and the medical attributes per condition. The algorithm then utilizes the *k*-NN technique to classify patients as having or not having a specific condition. This is due to the *k*-NN method's various

advantages over other supervised learning techniques. For example, *k*-NN fits this project better than Naive Bayes due to the former's non-parametric nature. *k*-NN is adaptive to relatively noisy training sets, simple to implement, and naturally handles multi-class cases. *k*-NN also has a history of higher success rates in the medical field, outperforming Quotient and Linear Discriminant Analysis [13].

To classify a new subject as positive for a condition, we find the patient in the training set who best matches the subject, and then use that patient's diagnosis as our prediction for the subject. The hypothesis is that if two points are near each other in the scatterplot, then the corresponding measurements are similar, so they can be expected to receive the same diagnosis. To diagnose a patient, we observe the *k* points that are closest to the patient's data, and use the diagnosis for each of those *k* points to predict the patient's diagnosis. In particular, the majority value among those *k* diagnoses will be used as the diagnostic predictor. *Figure 1* demonstrates why we chose *k*-NN as the classification method.

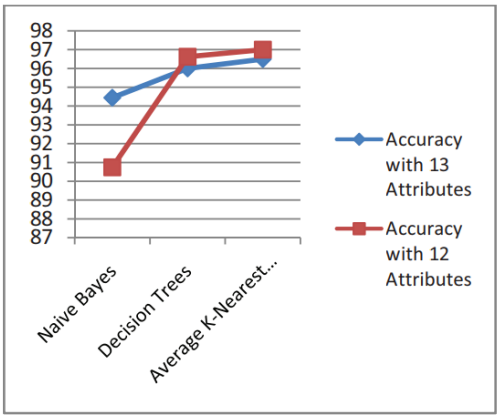

Fig 1. Classification accuracy [3]

Definition 1*:* The *training set* is a pre-established table of data with which the *test point* or *test data* is being compared. In the scope of this research, the training set represents many patients with a set number of variables/attributes. Each row represents a patient while each column represents a diagnostic attribute. The last column contains a determining factor of positive or negative for a condition. The attributes will be defined as the *n - 1* columns while the determining factor is the *nth* column. The test data is a patient with the same number of variables but no column on the determining factor of the condition.

Due to classification by proximity, the algorithm is able to form nonlinear, adaptive decision boundaries for each data point. By assuming all points are in k-dimensional space, *k*-NN uses the distance formula to determine proximity between the test point and the *k* nearest points in the training set.

Definition 2*:* Otherwise known as the Euclidian distance, the distance between two points can be found using the *distance formula.* The distance between (x_1, y_1) and (x_2, y_2) is given by:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

When applying *k*-NN to classifications involving more attributes, training samples must similarly contain the same number of attributes. Likewise, when more attributes are involved, the distance formula is adjusted for more dimensions to apply to the n number of attributes. For two points $(a_1, b_1, c_1, \ldots, n_1)$, $(a_2, b_2, c_2, \ldots, n_2)$:

$$\sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2 + (c_2 - c_1)^2 + \cdots + (n_2 - n_1)^2}$$

A point is classified by finding its nearest neighbors and picking the most widespread class among the neighbors. When a piece of patient input is entered, the *k*-NN classifier starts searching for the *k* training samples closest to the patient's input.

Algorithm 1 Standard *k*-Nearest Neighbors method
Input: Training table of *n* columns where *n - 1* columns are inputs and the *n*th column is a class variable with two possibilities. A test data point with one row of similar row structure as training table but with the *n - 1* columns.
Output: Predicted *n*th column value of test data.
1. Decide on value for *k*
2. Find the *k* closest point in the training set compared to the test data point
3. Classify the test data point by the most popular

Algorithm 1 is the standard *k*-NN classification method.

Algorithm 2 Distance Computation
Input: Multiple numerical training arrays $[a_1, b_1, c_1, \ldots, n_1]$ and one test patient array $[a2, b_2, c_2, \ldots, n_2]$.
Output: Euclidean distance from one input array to the others.
1. Compute multi-dimensional distance between test patient array and each training array.
2. Returns list of distances.

Algorithm 3 Table of Distances
Input: Training dataset of *n* columns where *n - 1* columns are inputs and the *n*th column is the class of the condition. Test patient array with single row of same *n - 1* columns as in the training table.
Output: Table of distances from test point to each point in training dataset.
1. Input training table.
2. Apply **Algorithm 2** training dataset and test patient array to return list of distances.
3. Return table with each row as the distance from the testing point to each training point. Taken from list of distances.

Algorithm 3 is a process that takes in the pre-determined training dataset as well as the patient data array. It applies **Algorithm 2** to calculate distances between the patient data and training data, finally entering this array of distances into a table of distances shown in *Figure 2*.

Algorithm 4 Closest *k* Points
Input: Test patient array, *k* value, and table of distances from **Algorithm 2**.
Output: Table of *k* closest rows in the training table to test patient array
1. Sort table of distances in ascending order.
2. Take the *k* first rows from the sorted table and return it as a separate table.

Algorithm 4 is a function locates the *k* closest neighbors from the table of distances shown in *Figure 2*.

Algorithm 5 Classification
Input: *k* closest rows table from **Algorithm 3**
Output: Predicted diagnosis of test patient array.
1. Count the number of positive and negative diagnoses from last column of *k* closest rows table.
2. Return predicted diagnosis as the class of the more numerous diagnoses.

Algorithm 5 is a process that classifies the patient based on the majority class of the *k* neighbors and returns the final diagnosis. Last steps in *Figure 2* exemplifies Algorithm 5.

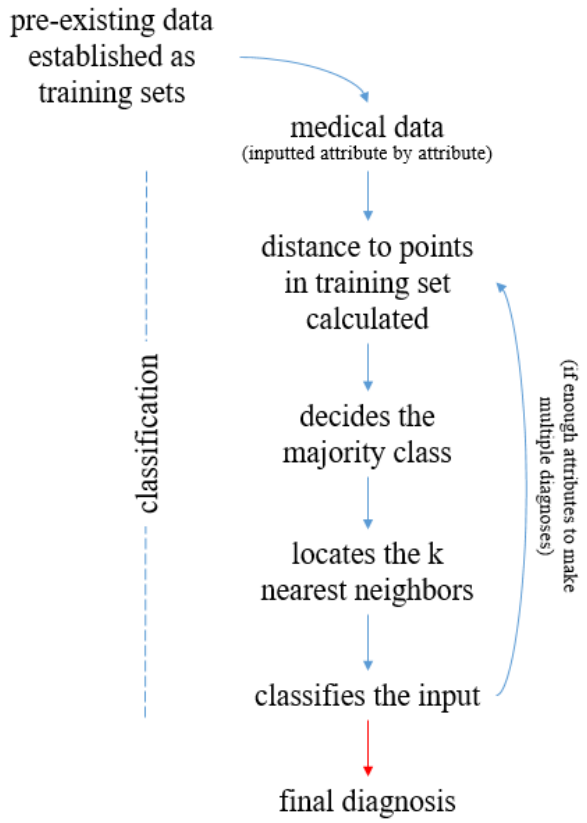

Fig 2. Methodology block diagram

C. *Accuracy*

Accuracy of the classifier will be calculated using a modified bootstrap sampling method. The original dataset will be randomly selected into a new training set and new testing set. The decision boundary of the new training set will be used to classify each point within the new testing set. The results of this test will determine the accuracy of the classifier.

Algorithm 6 Training Set and Testing Set Creation
Input: The training set for the condition.
Output: A shuffled training set and shuffled testing set sampled randomly from the original training set of points.
1. Randomly shuffle elements of the training table into a shuffled table.
2. Create shuffled training set with takes in first half of the shuffled table.
3. Create shuffled test set with rest of the shuffled table.

Algorithm 6 is a function that generates a new training set and testing set from the original training set through random sampling. This is shown in the first few elements of *Figure 3*.

Algorithm 7 Difference Comparison
Input: Two arrays.
Output: Number of similar elements in both arrays.
1. Subtract the length of each input array by the number of nonzero entries in that array to find the number of similar elements in both arrays.

Algorithm 7 is a function used to determine the similar elements in two input arrays, used in second to last element of *Figure 3*.

Algorithm 8 Accuracy
Input: Training set, *k*.
Output: Accuracy of classifier in percentage.
1. Use **Algorithm 6** to return a shuffled training set and shuffled testing set.
2. Drop the class label from the shuffled testing set and apply **Algorithm 5** to classify the shuffled testing set into the shuffled classified testing set.
3. Input the shuffled classified testing set and Training set class column into **Algorithm 7** to find the number of similar elements in both arrays.
4. Divide the number of similar elements in both arrays by the length of the training set class column to return the percentage.

Algorithm 8 is the consolidated process of the accuracy algorithm. It utilizes Algorithm 6 to return a randomly sampled or shuffled training set and shuffled testing set. Then, Algorithm 5 is applied to the shuffled testing set to determine the classification of each point within the shuffled testing set. Algorithm 7 then determines the similar elements between the classified diagnoses and the actual diagnosis from the original training dataset. This is summed up in *Figure 3*.

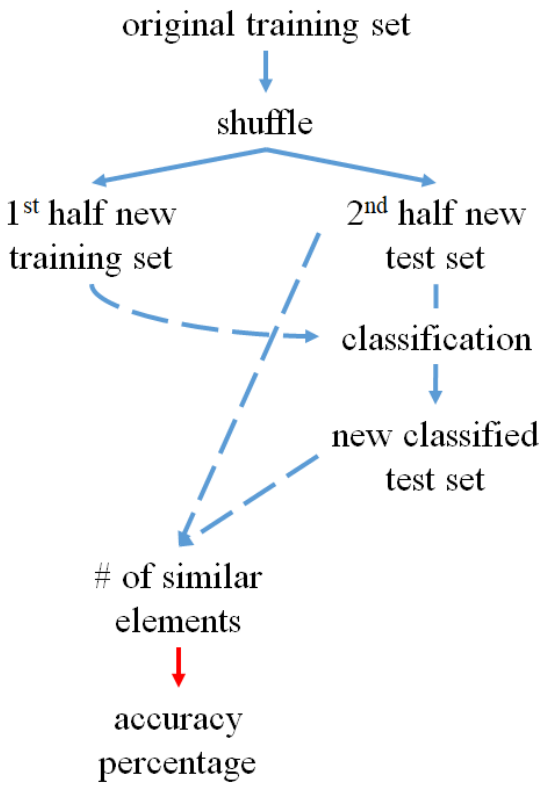

Fig 3. Flowchart of the Accuracy Calculation Process

IV. Results

A. *Datasets*

In our *k*-NN algorithm, we used two datasets from the UCI Machine Learning Repository. *Figures 4* through *7* below summarizes the UCI chronic kidney failure and heart disease datasets. It is noteworthy that we are exclusively viewing quantitative data because *k*-NN is superior in processing numerical data and inferior at processing qualitative data.

B. *Discussion of Results*

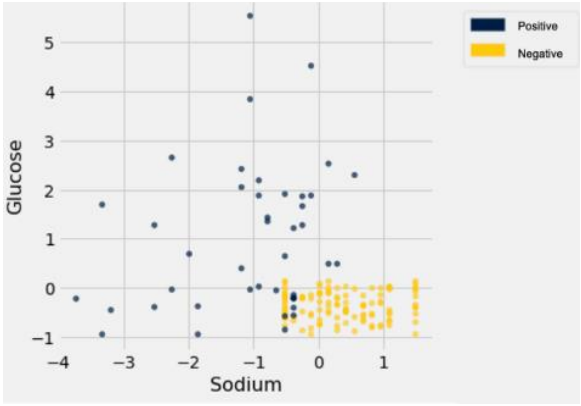

Fig 4. 2-Attribute Scatterplot

The Chronic Kidney Dataset, published as an open source dataset by the UCI Machine Learning Repository, uses the data of 400 patients measuring 25 distinct attributes over the course of 2 months to determine if a patient is suffering from chronic kidney disease. All 25 attributes included in the dataset are utilized in our *k*-NN algorithm [19]. The UCI Machine Learning Repository Heart Disease Dataset uses 76 attributes to measure if 303 subjects have or do not have heart disease. This dataset is a combination of three distinct datasets: the Cleveland, Hungazerland, and VA Long Beach datasets. In the aforementioned dataset, there are 76 attributes applied to the set, however, all but 14 attributes cannot be used because they either cannot be analyzed by *k*-NN, or they are incomplete [20].

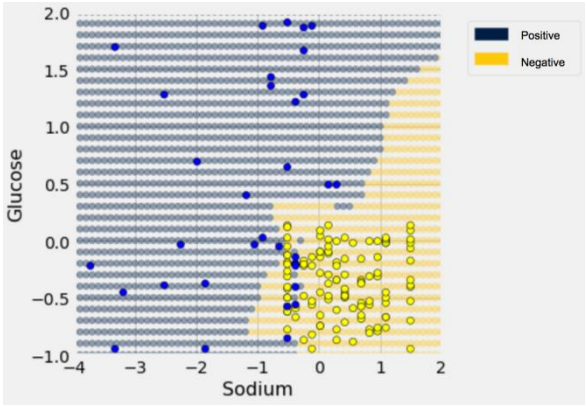

Fig 5. 2-Attribute k-NN Decision Boundary Scatterplot

Figure 4 is a visual representation of the chronic kidney disease classification with two attributes. Blue dots are patients within the training set positive for the disease while yellow dots are patients in the training set negative for the condition. *Figure 5* is the same as *Figure 4* but with a superimposed decision boundary.

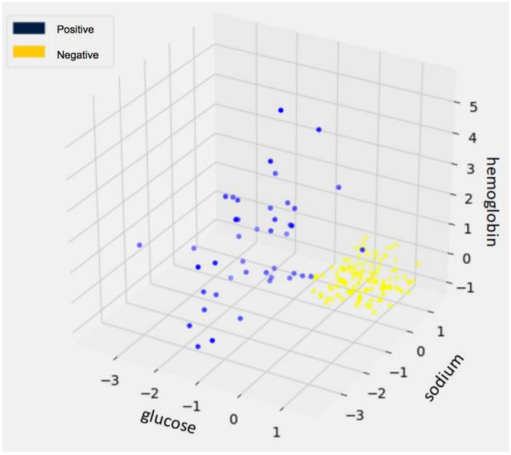

Fig 6. 3-Attribute Scatterplot

Figure 6 is an example of multiple attribute classification for chronic kidney disease. Actual classifications will utilize over 10 attributes, which are impossible to visualize.

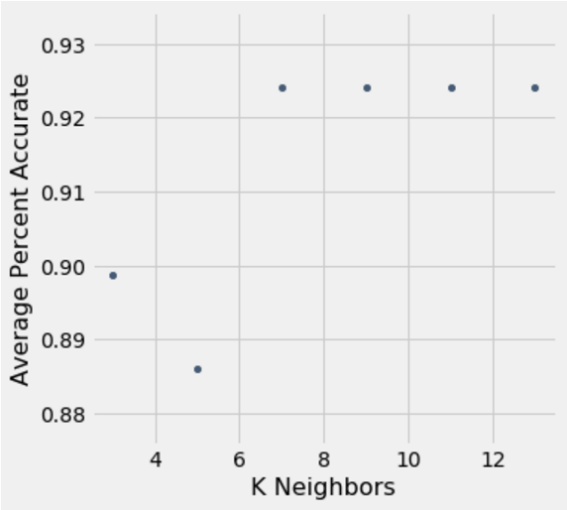

Fig 7. Percentage Accuracy after n numbers of k Neighbors

Figure 7 displays method accuracy based on number of *k* neighbors used. Each point represents the average of a hundred runs of the methods with the respective number of *k* neighbors.

B. Overview

The rest of the results utilize generated data assuming the patient inputs enough data to receive a prediction for one or more medical conditions. *Figure 4* is a graph depicting classification with 2 attributes and *Figure 5* is *Figure 4* with a superimposed decision boundary.

Definition 3: The decision boundary labels the boundary at which an inputted user is declared positive or negative for the medical condition.

Classification results are about 90% accurate in regards to the two datasets used in method. The choice for the *k* number of neighbors has been decided by running the method a hundred times with each odd number of *k* up to 13. The average percentage for the one hundred trials for each value of *k* is plotted in *Figure 7*. As depicted by *Figure 7*, using less than $k = 5$ neighbors results in a fluctuating decline of accuracy. Using more than $k = 5$ neighbors returns a steady accuracy rate of ~90%. Due to the variability of the dataset, there is a 5% margin of error which will be exemplified in the conclusion. In comparison with similar studies, our accuracy percentage lies in the ballpark of other *k*-NN research papers, slightly higher than [17] and lower than [14] as shown in *Figure 8*.

C. Limitations

Learning within the *k*-NN method is very simple, but the classification process is quite time consuming. *k*-NN has certain disadvantages, including a high computation cost due to the need to determine the distance of each test instance to all training samples. The algorithm also requires large memory proportional to the size of the training set. This may cause significant accuracy drops in medical conditions with low proximity between points [19]. Low accuracy rates also come into play when considering multidimensional datasets with irrelevant features.

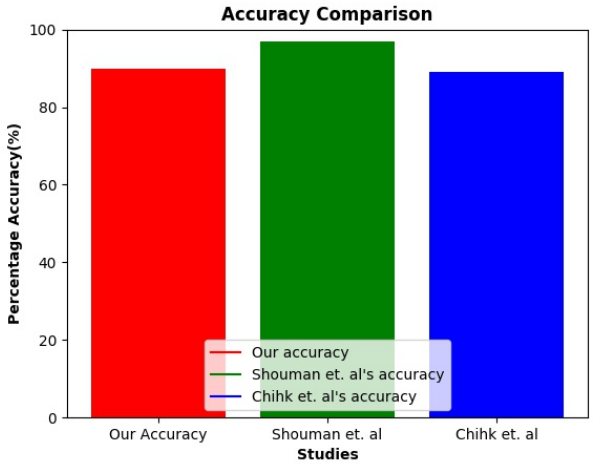

Fig 8. Comparison Chart

V. CONCLUSION

This research presents how artificial intelligence applied to medical diseases yields efficient diagnoses. This is achieved using *k*-NN to diagnose current or potential diseases. We test the accuracy of this algorithm with UCI Machine Learning Repository datasets, specifically heart disease and chronic kidney failure. The results show a diagnostic accuracy of 90% give or take a 5% margin of error due to the presence of outliers within the training dataset. This accuracy percentage is relatively close to [14]'s study of applying *k*-NN to heart disease diagnosis, and shows improvement to [17]'s research on a hybrid algorithm to diagnose diabetes. Due to various constraints, we had to generate patient input and test data for diagnosis, derailing possible practical applications. Using real-patient data when running the algorithm brings our methods and algorithms into line for practical use. Further, the simplicity and widespread nature of the algorithm makes this method unique and effective. Inputting additional training sets allows more medical conditions to be classified with minimal changes to the overall algorithm.

Future research aims at larger scale applications of medical diagnoses. For instance, datasets that measure variables for a more diverse range of diseases, such as neurological disorders or cancer. Using different algorithms to tackle new datasets that do not share the memory and multidimensional limitations of *k*-NN can also be explored further. Furthermore, by converting qualitative data using similarity score functions, such attributes as age and gender can be used to classify individuals. This can be combined with discrete time procedures such as the Fast Fourier Transform to decrease numbers of attributes while increasing number of diagnosable conditions. Another upcoming goal is to install this algorithm in a wearable device for easy accessibility. Future studies to improve the accuracy of this method can result in its use in a professional setting, allowing medical professionals to make efficient diagnoses and quickly move on to treatment.

ACKNOWLEDGMENT

This material is based upon work supported in part by the Department of Defense under Army Educational Outreach Program (AEOP) and the National Science Foundation under Grant No. 1710716. The authors thank the UNLV writing center for helping with revising the manuscript.

REFERENCES

1. Sohail, S., Clark, K., & Fagan, J. M. (2016). Fitness Gadgets as a Form of Preventative Healthcare.
2. Marcus, G., & Sanchez, J. (n.d.). Can a smart watch save you from a stroke? Retrieved June 19, 2017.
3. Im, A. (n.d.). Imagine a world of perfect health risk awareness. Retrieved June 19, 2017.
4. Q. Liu, C. Liu, "A Novel Locally Linear KNN methoecognition." IEEE Trans. pp. 1–12, May. 2016.
5. Kalaiselvi, C. (2016, March). Diagnosing of heart diseases using average k-nearest neighbor algorithm of data mining. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* (pp. 3099-3103). IEEE.
6. Chen, H., Huang, C., Yu, X., Xu, X., Sun, X., Wang, G., & Wang, S. (2013). An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. Expert Systems with Applications, 40(1), 263-271. Retrieved July 17, 2017.
7. Kaykcioglu, T., & Aydemir, O. (2010). A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data. Pattern Recognition Letters, 31(11), 1207-1215. Retrieved July 11, 2017.
8. Warfield, S. (1996). Fast k-NN classification for multichannel image data. *Pattern Recognition Letters*, 17(7), 713-721. Retrieved July 11, 2017.
9. Kononmenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine, 23(1), 89-109. Retrieved July 11, 2017.
10. Pawlovsky, A. P., & Matsuhashi, H. (2017, March). The use of a novel genetic algorithm in component selection for a kNN method for breast cancer prognosis. In *Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE), 2017* (pp. 1-5). IEEE.
11. Sahan, S., Polat, K., Kodaz, H., & Gunes, S. (2007). A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis. Computers in Biology and Medicine, 37(3), 415-423. Retrieved July 11, 2017.
12. Huang, Y. (2004). Prediction of protein subcellular locations using fuzzy k-NN method. Bioinformatics, 20(1), 21-28. Retrieved July 11, 2017.
13. Kim, K. S., Choi, K. H., Moon, C. S., & Mun, C. W. (2011). Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions. Current Applied Physics, 11(3), 740-745. Retrieved July 11, 2017.
14. Shouman, M., Turner, T., & Stocker, R. (2012). Applying k-nearest neighbour in diagnosing heart disease patients. *International Journal of Information and Education Technology*, 2(3), 220. Retrieved July 11, 2017.
15. Kalaiselvi, C. (2016, March). Diagnosing of heart diseases using average k-nearest neighbor algorithm of data mining. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* (pp. 3099-3103). IEEE.
16. Ani, R., Sasi, G., Sankar, U. R., & Deepa, O. S. (2016, September). Decision support system for diagnosis and prediction of chronic renal failure using random subspace classification. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on* (pp. 1287-1292). IEEE.
17. Chikh, M. A., Saidi, M., & Settouti, N. (2012). Diagnosis of Diabetes Diseases Using an Artificial Immune Recognition System2 (AIRS2) with Fuzzy K-Nearest Neighbor. Journal of Medical Systems, 36(5), 2721-2729. Retrieved July 13, 2017.
18. Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R., "Efficient kNN Classification With Different Numbers of Nearest Neighbors" IEEE Transactions on Neural Networks and Learning Systems. Retrieved July 14, 2017.
19. Chronic Kidney Disease Data Set. (n.d.). Retrieved July 13, 2017.
20. Heart Disease Data Set. (n.d.). Retrieved July 13, 2017.