

Comparison of Neural Network and Machine Learning Approaches in Prediction of Chronic Kidney Disease

Shreya Nag¹ and Nimitha Jammula[#]

¹Jasper High School, Plano, TX, USA

[#]Advisor

ABSTRACT

The diagnosis of a disease to determine a specific condition is crucial in caring for patients and furthering medical research. The timely and accurate diagnosis can have important implications for both patients and healthcare providers. An earlier diagnosis allows doctors to consider more methods of treatment, allowing them to have a greater flexibility of tailoring their decisions, and ultimately improving the patient's health. Additionally, a timely detection allows patients to have a greater control over their health and their decisions, allowing them to plan ahead. As advancements in computer science and technology continue to improve, these two factors can play a major role in aiding healthcare providers with medical issues. The emergence of artificial intelligence and machine learning can aid in addressing the challenge of completing timely and accurate diagnosis. The goal of this research work is to design a system that utilizes machine learning and neural network techniques to diagnose chronic kidney disease with more than 90% accuracy based on a clinical data set, and to do a comparative study of the performance of the neural network versus supervised machine learning approaches. Based on the results, all the algorithms performed well in prediction of chronic kidney disease (CKD) with more than 90% accuracy. The neural network system provided the best performance (accuracy = 100%) in prediction of chronic kidney disease in comparison with the supervised Random Forest algorithm (accuracy = 99%) and the supervised Decision Tree algorithm (accuracy = 97%).

Introduction

Chronic kidney disease, also called chronic kidney failure, describes the gradual loss of kidney function. The kidneys filter wastes and excess fluids from the blood, which are then excreted in urine. When chronic kidney disease reaches an advanced stage, dangerous levels of fluid, electrolytes, and wastes can build up in the body. In the early stages of chronic kidney disease, there may be very few signs or symptoms. Chronic kidney disease may not become apparent until the kidney function is significantly impaired. Treatment for chronic kidney disease focuses on slowing the progression of the kidney damage, usually by controlling the underlying cause. Chronic kidney disease can progress to end-stage kidney failure, which is fatal without artificial filtering (dialysis) or a kidney transplant.

It is estimated that 37 million American adults have Chronic kidney disease and millions of others are at increased risk. Early detection can help prevent the progression of kidney disease to kidney failure. Heart disease is the major cause of death for all people with Chronic Kidney Disease. Glomerular filtration rate (GFR) is the best estimate of kidney function, and persistent proteinuria (protein in the urine) means Chronic kidney disease is present. High risk groups include those with diabetes, hypertension, and family history of kidney failure. African Americans, Hispanics, Pacific Islanders, American Indians, and seniors are at increased risk.

Current Artificial Intelligence (AI) approaches have a diagnostic performance that is comparable with medical experts, especially in image recognition-related fields. With the continued development of AI-assisted technologies, the clinical implications as captured by the clinicians' experience and guided by patient-centered health care principles should be constantly considered in future technology-based medical research [1]. The supervised machine

learning and the neural network approaches fall in the categories of AI-assisted technologies and can be utilized effectively in early prediction and diagnosis of diseases including chronic kidney disease.

The developments in computer sciences and technology, specifically in artificial intelligence (AI) and machine learning (ML), can help in addressing these challenges. Use of AI technology will result in reduction of costs in providing care, along with making the care faster and more efficient. The use of AI and ML technologies will cause changes in the medical profession with greater focus on tasks related to creativity and critical thinking than time-consuming repetitive tasks [3]. There have been many research studies which have suggested that AI techniques can perform as well as or better than humans at key healthcare tasks, such as diagnosing diseases.

Neural Networks

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling, or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text, or time series, must be translated. Neural Networks are effective in solving problems related to pattern recognition and matching, clustering, and classification [2]. Neural networks help to cluster a group of unlabeled data according to similarities among the example inputs, and to classify data when they have a labeled data set to train with. Neural networks can also extract features that are fed to other algorithms for clustering and classification. Deep learning maps inputs to outputs and finds correlations. It is known as a “universal approximator” because it can learn to approximate an unknown function $f(x) = y$ between any input x and any output y , assuming they are related by correlation or causation. In the process of learning, a neural network finds the right function f , or the correct manner of transforming x into y . Deep learning is the name that is used for “stacked neural networks”, which comprises of networks composed of several layers. The layers are made of *nodes*. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs regarding the task the algorithm is trying to learn, for example which input is most helpful is classifying data without error. These input-weight products are summed and then the sum is passed through a node’s so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome. If the signal passes through, then the neuron gets “activated.” Figure 1 shows the Neural Network message flow and architecture.

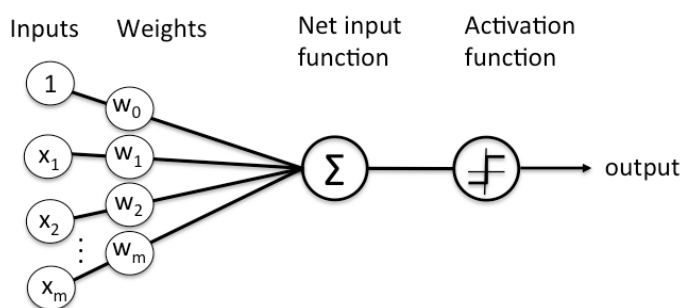


Figure 1 – Neural Network Architecture and Components

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer’s output is simultaneously the subsequent layer’s input, starting from an initial input layer receiving your data.

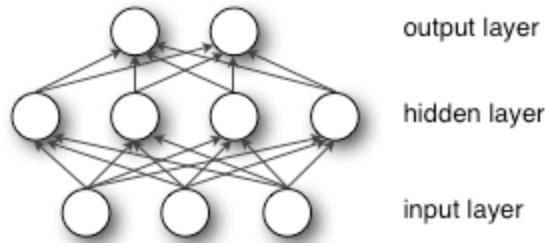


Figure 2 – Neural Network Input Layer to Output Layer Flow

Pairing the model’s adjustable weights with input features is how we assign significance to those features regarding how the neural network classifies and clusters input.

Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition. Earlier versions of neural networks such as the first perceptron were shallow, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as “deep” learning. In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer’s output.

ML algorithms can broadly be categorized according to the purposes they are designed for. The primary categories of ML algorithms include - supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning is classified into two broad categories of algorithms. The first category is classification; a classification problem is defined when the output variable is a category, such as “malignant” or “benign”, “disease”, or “no disease”. The second category is regression, which is defined when the output variable is a numerical value. Supervised learning algorithms try to model relationships and dependencies between the target prediction output and the input features such that the output values for new data can be predicted based on those relationships which it learned from the previous data sets [3].

Decision Tree Algorithm

A decision tree is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value. It partitions the tree recursively manner using recursive partitioning. This flowchart-like structure helps in decision making. The visualization is like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret [4].

The basic idea behind any decision tree algorithm is as follows:

Select the best attribute using Attribute Selection Measures (ASM) to split the records.

Make that attribute a decision node and breaks the dataset into smaller subsets.

Starts tree building by repeating this process recursively for each child until one of the conditions will match:

All the tuples belong to the same attribute value.

There are no more remaining attributes.

There are no more instances.

Figure 3 shows the flow of the Decision Tree Algorithm.

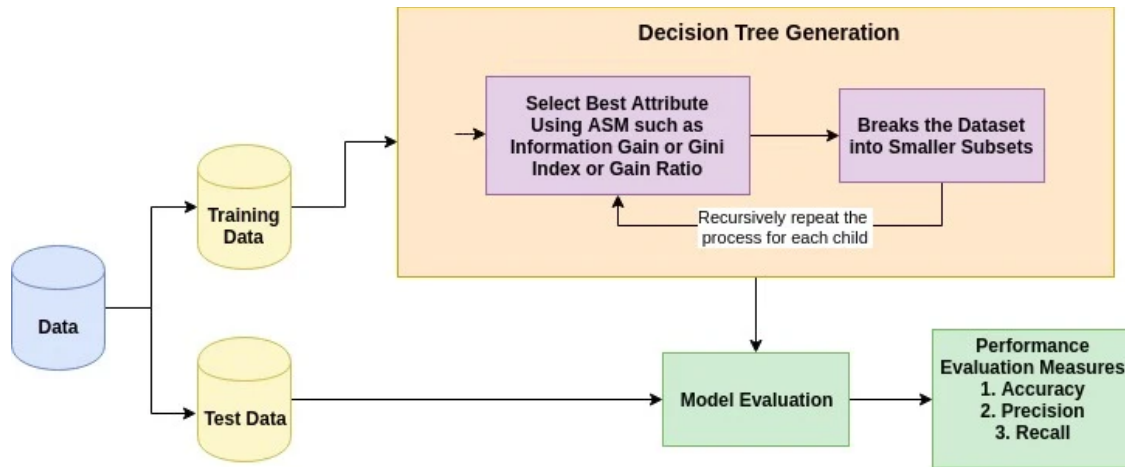


Figure 3 – Flow of Decision Tree Algorithm

Random Forest Algorithm

Random forest is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forest creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance [5].

It works in four steps:

Select random samples from a given dataset.

Construct a decision tree for each sample and get a prediction result from each decision tree.

Perform a vote for each predicted result.

Select the prediction result with the most votes as the final prediction.

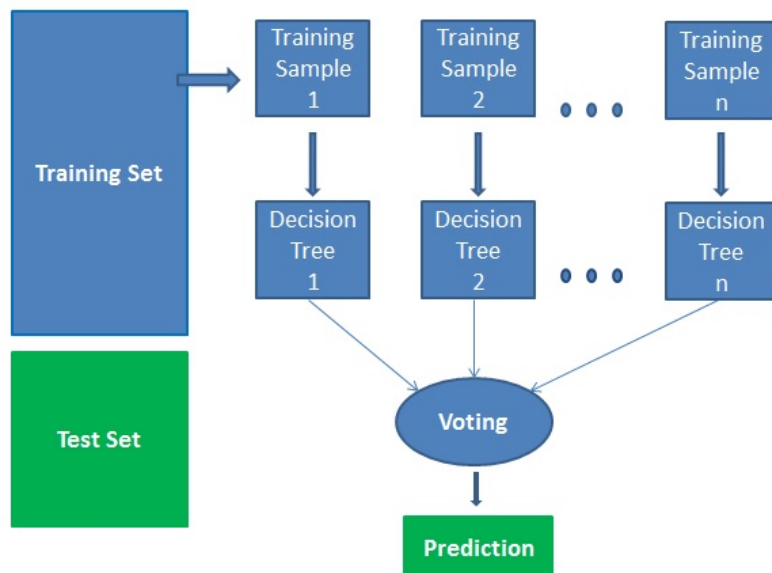


Figure 4 - Flow of Random Forest Algorithm

Chronic Kidney Disease Dataset

The Chronic Kidney Disease Dataset from the University of California, Machine Learning repository was used to design the neural networks and supervised machine learnings to perform the analysis research [6]. Table 1 shows the Chronic Kidney Dataset with the attributes, their description and value units of measurement. The input features are captured from the clinical data and are used to determine the resulting output label whether the patient has Chronic Kidney Disease (CKD) or not.

Table 1 – Chronic Kidney Disease Dataset Information

Attributes	Description	Value and Units
Input Features		
age	Age(numerical)	age in years
bp	Blood Pressure(numerical)	bp in mm/Hg
sg	Specific Gravity(nominal)	sg - (1.005,1.010,1.015,1.020,1.025)
al	Albumin(nominal)	al - (0,1,2,3,4,5)
su	Sugar(nominal)	su - (0,1,2,3,4,5)
rbc	Red Blood Cells(nominal)	rbc - (normal, abnormal)
pc	Pus Cell (nominal)	pc - (normal, abnormal)
pcc	Pus Cell clumps(nominal)	pcc - (present, notpresent)
ba	Bacteria(nominal)	ba - (present, notpresent)
bgr	Blood Glucose Random(numerical)	bgr in mgs/dl
bu	Blood Urea(numerical)	bu in mgs/dl
sc	Serum Creatinine(numerical)	sc in mgs/dl
sod	Sodium(numerical)	sod in mEq/L
pot	Potassium(numerical)	pot in mEq/L
hemo	Hemoglobin(numerical)	hemo in gms
pcv	Packed Cell Volume(numerical)	
wc	White Blood Cell Count(numerical)	wc in cells/cumm
rc	Red Blood Cell Count(numerical)	rc in millions/cmm
htn	Hypertension(nominal)	htn - (yes, no)
dm	Diabetes Mellitus(nominal)	dm - (yes, no)
cad	Coronary Artery Disease(nominal)	cad - (yes, no)
appet	Appetite(nominal)	appet - (good, poor)
pe	Pedal Edema(nominal)	pe - (yes, no)
ne	Anemia(nominal)	ane - (yes, no)
Output		
class	Class (nominal)	class - (ckd, notckd)

Information Gain

Information Gain is used to determine which feature/attribute gives us the maximum information about a class. It is based on the concept of entropy, which is the degree of uncertainty, impurity, or disorder. It aims to reduce the level of entropy starting from the root node to the leaf nodes. Entropy is calculated as shown below.

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

'p' denotes the probability and E(S) denotes the entropy. Entropy is not preferred due to the 'log' function as it increases the computational complexity.

Gini Index

Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. If all the elements belong to a single class, then it can be called pure. The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. A Gini Index of 0.5 denotes equally distributed elements into some classes. Gini Index is calculated as shown below.

$$\text{Gini} = 1 - \sum_{i=1}^n p_i^2$$

where p_i = the probability of an object being classified to a particular class.

While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

Figure 5 shows the correlations between the different input attributes in the x-axis and the y-axis. The darker red indicates stronger correlation and the darker blue indicates weaker correlation.

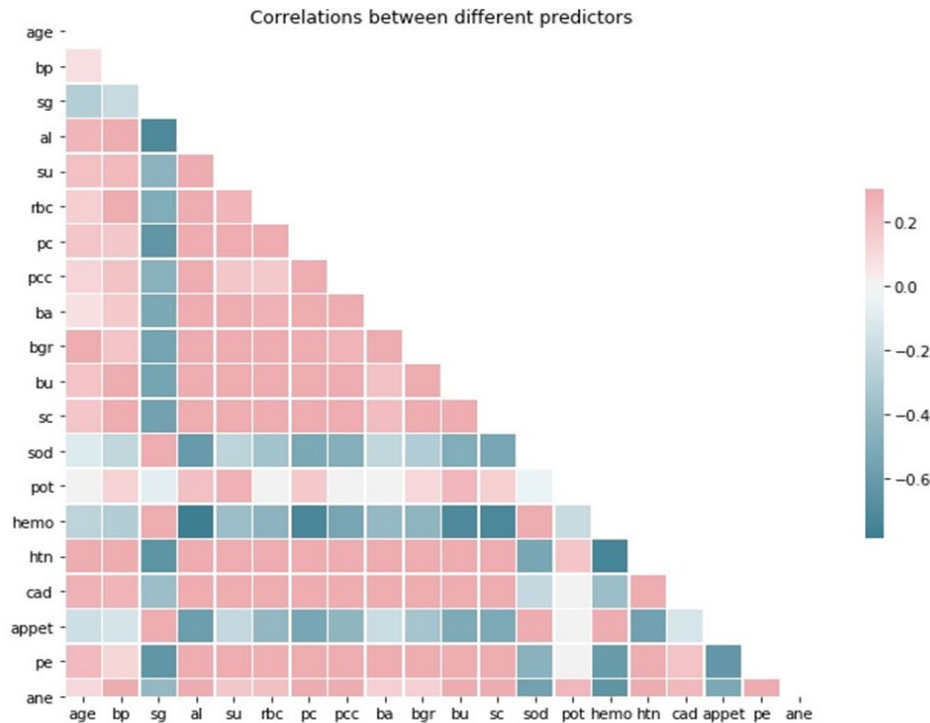


Figure 5 – Correlation between Different Predictors

Confusion Matrix

To describe the performance of a classification model (or ‘classifier’) on a set of test data, a confusion matrix was used (Table 2). The actual value indicates the correct actual outcome. The predicted value indicates the prediction of the classification model. The goal is to reliably predict true positives and true negatives.

Table 2 - Confusion Matrix

	Predicted: NO	Predicted: YES
Actual: NO	True Negatives (TN)	False Positives (FP)
Actual: YES	False Negatives (FN)	True Positives (TP)

Classification Report

The Classification Report visualizer displays the precision, recall, F1, and support scores for the model. Precision is the ability of a classifier to avoid labeling a negative instance as positive. For each class, it is the ratio of true positives (TP) to the sum of true positives (TP) and false positives (FP). Recall is the ability of a classifier to find all positive instances. For each class, it is defined as the ratio of TP to the sum of TP and FN. F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are usually lower than accuracy measures as they embed precision and recall into their computation. Support is the number of actual occurrences of the class in the specified dataset. The F1 score calculation is used to determine the accuracy of the algorithm.

ROC Curve

ROC (Receiver Operating Characteristic) curves are used to see how any predictive model can distinguish between the true positives and negatives. To do this, a model needs to not only correctly predict a positive as a positive, but also predict a negative as a negative. The ROC curve does this by plotting sensitivity, the probability of predicting a real positive will be a positive, against 1-specificity, the probability of predicting a real negative will be a positive. The best decision rule is high on sensitivity and low on 1-specificity.

AUC

AUC (Area under the ROC Curve) measures the two-dimensional area underneath the ROC curve. AUC is one of the most important evaluation metrics for checking any classification model’s performance. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0. Higher value of AUC indicates higher performance of the classification model.

Methods

To perform the analysis and design research of the neural network and supervised machine learning algorithms the following methodology was followed.

Data was gathered about the Chronic Kidney Disease and the various parameters of the Chronic Kidney Disease Data Set from the UCI Machine Learning Repository. The chronic kidney disease data set was downloaded from the UCI Machine Learning Repository.

The Chronic Kidney Disease Data Set was split into training data and test data by using a method of the TensorFlow Machine Learning Software Library. The ideal split is considered to be a training data: test data split ratio of 75:25 [8]. The testing was performed with average split ratio of 75:25, and other test runs were performed with split ratios between 70:30 and 80:20.

The training data was used to train the system to get an accurate prediction, and the test data was used to test the model's prediction to ensure that the program works properly.

The performance of the classification algorithms, Decision Tree and Random Forest, were analyzed using Classification Reports and Confusion Matrices.

The performance of the neural networks was analyzed using an Accuracy and Loss Model. An epoch indicates the number of passes of the entire training dataset the machine learning algorithm has completed. If the batch size is the whole training dataset then the number of epochs is the number of iterations.

The accuracy of the predictions was checked by calculating the ratio of the number of correct predictions to the total number of predictions made for each of the classification algorithms.

The Python Graph plotting library is used to display the results of each classification algorithm. The Confusion Matrix and Classification Reports are generated for each classification algorithm using the Scikit Learn (sklearn.metrics) library. The ROC curves are generated using the Scikit Learn (sklearn.metrics) library and the Python Matplotlib (matplotlib.pyplot) library [7]. The results are compared to determine the best classification algorithm for the data set. The ROC curves are based on the following definitions for sensitivity and specificity.

Sensitivity is the proportion of actual positive cases which are correctly identified and is calculated as:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

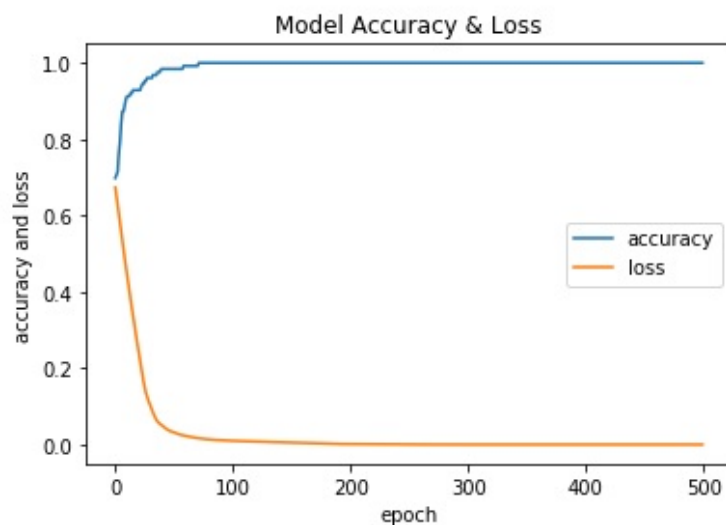
Specificity is the proportion of actual negative cases which are correctly identified and is calculated as:

$$\text{Specificity} = \frac{TN}{TN+FP}$$

Multiple trials are conducted with the test data to observe and validate the observe the accuracy of the predictions.

Results

Following are the performance results of the Neural Networks with epoch set to 500.



Shape of training data: (126, 25)
Shape of test data : (32, 25)

Figure 6 - Neural Network Model Accuracy and Loss with Epoch 500


```

Model file: ckd.best.model
32/32 [=====] - 0s 6ms/step

Original  : 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1

Predicted : 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1

Scores    : Loss = 0.030842883279547095 % Accuracy = 100.0 %
-----

Model file: ckd.model
32/32 [=====] - 0s 5ms/step

Original  : 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1

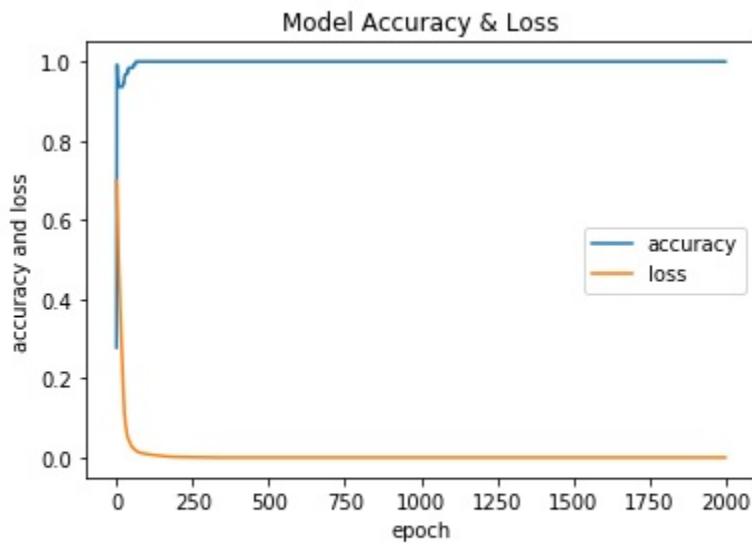
Predicted : 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1

Scores    : Loss = 0.026985176373273134 % Accuracy = 100.0 %
-----

```

Figure 7 - Output Results with Neural Network Model on Chronic Kidney Disease dataset

Following are the performance results of the Neural Networks with epoch set to 2000.



```

-----
Shape of training data: (126, 25)
Shape of test data    : (32, 25)
-----

```

Figure 8 - Neural Network Model Accuracy and Loss with Epoch 2000

```

Model file: ckd.best.model
32/32 [=====] - 0s 9ms/step

Original : 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1

Predicted : 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1

|
Scores    : Loss = 0.0 % Accuracy = 100.0 %
-----

Model file: ckd.model
32/32 [=====] - 0s 13ms/step

Original : 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1

Predicted : 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1

Scores    : Loss = 0.0 % Accuracy = 100.0 %
-----

```

Figure 9 - Output Results with Neural Network Model on Chronic Kidney Disease dataset

Following are the output results of running the Decision Tree Algorithm on the Chronic Kidney Disease dataset. The prediction results are evaluated with the Gini Index and the Entropy calculations. The accuracy of the predictions is 96.67% in both cases.

Table 3 - Classification Report for Decision Tree Algorithm

Confusion Matrix				
	Predicted: NO		Predicted: YES	
Actual: NO	TN = 77		FP = 3	
Actual: YES	FN = 1		TP = 39	
Classification Report				
Test Size = 0.25	Precision	Recall	F1-score	Support
ckd	0.99	0.96	0.97	80
notckd	0.93	0.97	0.95	40
accuracy			0.97	120
macro avg	0.96	0.97	0.96	120
weighted avg	0.97	0.97	0.97	120

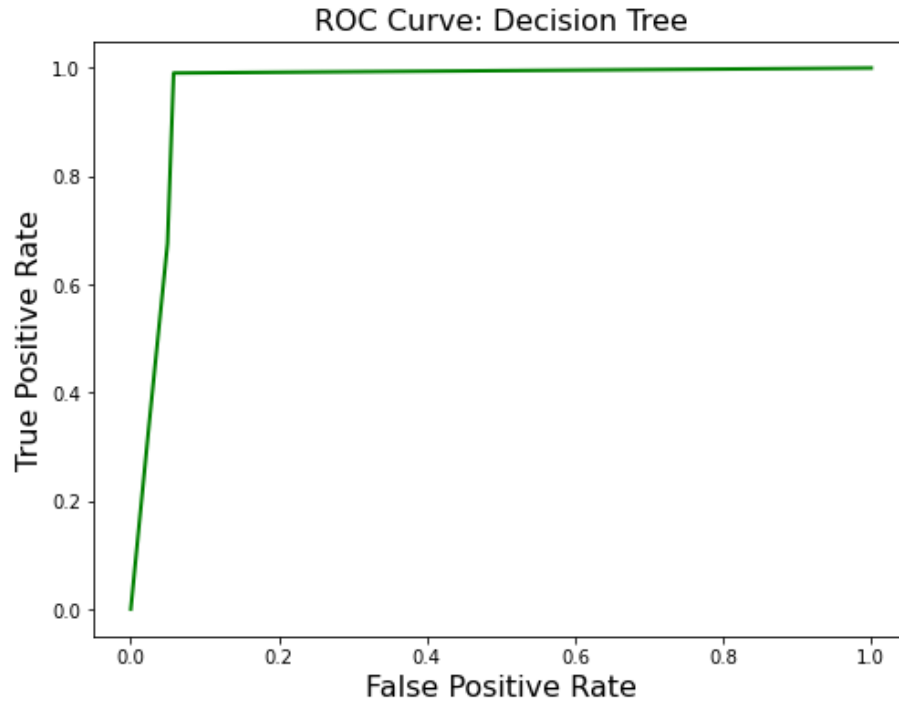


Figure 10 - ROC Curve for Decision Tree Algorithm

Following are the output results from the software program implemented with the Decision Tree Algorithm. The implementation used the Python Scikit-Learn Machine Learning library.

```

Dataset Length: 400
Dataset Shape: (400, 25)
Please select Classification Algorithm:
Decision Tree (1) or Random Forest (2): 1
[59.0 70.0 1.01 1.0 3.0 0.0 0.0 0.0 0.0 424.0 55.0 1.7 138.0 4.5 12.6 37.0
 10200.0 4.1 1.0 1 1.0 1.0 0.0 0.0]
Results Using Gini Index:
Confusion Matrix:
[[77  3]
 [ 1 39]]
Accuracy : 96.66666666666667
Report :
      precision    recall  f1-score   support

   ckd         0.99      0.96      0.97         80
  notckd        0.93      0.97      0.95         40

   accuracy          0.97         120
  macro avg         0.96      0.97      0.96         120
 weighted avg         0.97      0.97      0.97         120

Results Using Entropy:
Confusion Matrix:
[[77  3]
 [ 1 39]]
Accuracy : 96.66666666666667
Report :
      precision    recall  f1-score   support

   ckd         0.99      0.96      0.97         80
  notckd        0.93      0.97      0.95         40

   accuracy          0.97         120
  macro avg         0.96      0.97      0.96         120
 weighted avg         0.97      0.97      0.97         120

```

Figure 11 - Output Results with Decision Tree Algorithm on Chronic Kidney Disease dataset

Following are the output results of running the Random Forest Algorithm on the Chronic Kidney Disease dataset. The prediction results are evaluated with the Gini Index and the Entropy calculations. The accuracy of the predictions is 99.17% in both cases.

Table 4 - Classification Report for Random Forest Algorithm

Confusion Matrix				
	Predicted: NO		Predicted: YES	
Actual: NO	TN = 77		FP = 3	
Actual: YES	FN = 1		TP = 39	
Classification Report				
Test Size = 0.25	Precision	Recall	F1-score	Support
ckd	0.99	1.00	0.99	80
notckd	1.00	0.97	0.99	40
accuracy			0.99	120
macro avg	0.99	0.99	0.99	120
weighted avg	0.99	0.99	0.99	120

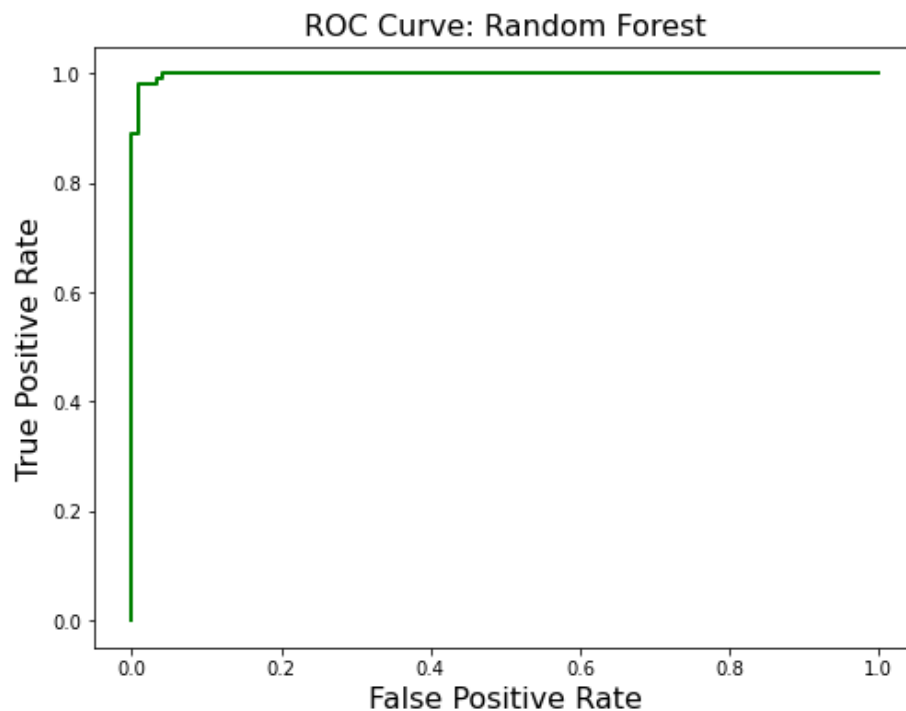


Figure 12 - ROC Curve for Random Forest Algorithm

Following are the output results from the software program implemented with the Random Forest Algorithm. The implementation used the Python Scikit-Learn Machine Learning library.

```

Dataset Length: 400
Dataset Shape: (400, 25)
Please select Classification Algorithm:

Decision Tree (1) or Random Forest (2): 2
[59.0 70.0 1.01 1.0 3.0 0.0 0.0 0.0 0.0 424.0 55.0 1.7 138.0 4.5 12.6 37.0
 10200.0 4.1 1.0 1 1.0 1.0 0.0 0.0]
Results Using Gini Index:
Confusion Matrix:
[[80 0]
 [ 1 39]]
Accuracy : 99.16666666666667
Report :
      precision    recall  f1-score   support

   ckd         0.99      1.00      0.99         80
  notckd        1.00      0.97      0.99         40

 accuracy
macro avg         0.99      0.99      0.99        120
weighted avg         0.99      0.99      0.99        120

Results Using Entropy:
Confusion Matrix:
[[80 0]
 [ 1 39]]
Accuracy : 99.16666666666667
Report :
      precision    recall  f1-score   support

   ckd         0.99      1.00      0.99         80
  notckd        1.00      0.97      0.99         40

 accuracy
macro avg         0.99      0.99      0.99        120
weighted avg         0.99      0.99      0.99        120

```

Figure 13 - Output Results with Random Forest Algorithm on Chronic Kidney Disease dataset

With a test size of 0.25 (25%), Decision Tree had an accuracy of about 96.67%. With a test size of 0.25 (25%), Random Forest had an accuracy of about 99.17%. With a test size of 0.25 (25%), the neural networks had an accuracy of 100%. The neural network system performed the best compared to the Decision Tree and Random Forest classification algorithms with an accuracy of 100%.

Conclusion

The selected supervised classification algorithms were Decision Tree and Random Forest. A Neural Network System was also created using the data. Decision Tree had an accuracy of about 96.67%. Random Forest had an accuracy of about 99.17%. The neural network had an accuracy of 100%.

All of the machine learning techniques performed well in the prediction of “ckd” (Chronic Kidney Disease positive) or “notckd” (Chronic Kidney Disease negative) and thus met the criteria of having greater than 90% accuracy. The Neural Network System was found to have higher accuracy than the Random Forest and the Decision Tree classification algorithms.

After researching the various types of classification algorithms, two algorithms were selected for design and testing. It is not known how other supervised learning classification algorithms would have performed, and they may or may not have provided better results. The amount of training data varied between 0.7 and 0.8, by splitting the data

set into training data and testing data. The amount of training data was considered adequate but may be insufficient in some cases.

The data in the data set may have some errors or outliers which could prevent the supervised machine learning system or neural network system from detecting the underlying patterns in a timely manner.

The neural networks and machine learning system can be used for the prediction of other diseases, such as cancer or Alzheimer's disease. Given new datasets from clinical test results for other diseases, the new system can use the dataset as an input and predict outcomes.

The machine learning system can be extended to use other supervised learning classification algorithms. The Random Forest algorithm can serve as a benchmark for performance, as it performed very well.

Acknowledgments

I would like to express sincere gratitude and thanks to my teachers at Plano West Senior High School, Ms. Gay Jaynes and Ms. Robin Bailey for their inspiration and guidance to conduct this research work. I take this opportunity to extend my heartfelt thanks to my research mentor, Ms. Nimitha Jammula for the valuable insights in software programming and the fields of machine learning and neural networks. I would also like to thank my parents for the resources and support to carry out the research.

References

- [1] Shen, Jiayi et al. "Artificial Intelligence Versus Clinicians in Disease Diagnosis: Systematic Review." *JMIR medical informatics* vol. 7,3 e10010. 16 Aug. 2019, doi:10.2196/10010
- [2] Dell'Aversana, Paolo "Artificial Neural Networks and Deep Learning", December 2019
- [3] Domingos, Pedro. "A Few Useful Things to Know about Machine Learning." *Communications of the ACM*, vol. 55, no. 10, 2012, pp. 78–87., doi:10.1145/2347736.2347755.
- [4] Ashari, Ahmad, et al. "Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool." *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 11, 2013, doi:10.14569/ijacsa.2013.041105.
- [5] Kirasich, Kaitlin, et al. "Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets." *SMU Data Science Review*, vol. 1, no. 9, ser. 3, 2018. 3.
- [6] Dua, D. "UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]." Irvine, CA: University of California, School of Information and Computer Science. n.p.: n.p., 12. 28 Dec. 2020.
- [7] Pedregosa, Fabian, et al. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, no. 85, 2011, pp. 2825–2830.
- [8] Guyon, I. "A Scaling Law for the Validation-Set Training-Set Size Ratio." *Semantic Scholar*, 1 Jan. 1997, www.semanticscholar.org/paper/A-Scaling-Law-for-the-Validation-Set-Training-Set-Guyon/452e6c05d46e061290feff8b46d0ff161998677.