DOI: 10.21917/ijsc.2019.0275

SUPPORT VECTOR MACHINE BASED DISEASE DIAGNOSTIC ASSISTANT

Samuel Ndirangu and Davies Segera
Department of Electrical and Information Engineering, University of Nairobi, Kenya

Abstract

There has been a huge growth both in data and computing technology which has made it easier for the development of artificial intelligent systems that are capable of learning from this data and make medical diagnosis on their own. In this paper, Support Vector Machines (SVM) are used in implementing a multi-disease diagnostic assistant application that is able to make predictions, early detections and instant diagnosis of various illness based on given patient data. The application is implemented in an easy to use graphical user interface and contains pretrained SVM models of predicting several diseases. A medical staff creates a new patient entry and enters or uploads a patient's required diagnostic data, once done the application gives multiple diagnosis based on the diagnostic data. In case the application makes a wrong diagnosis, it can learn from its mistake through correction from the medical staff, enabling future similar diagnosis to be correct.

Keywords:

Bayesian Optimization, Kernel Function, Sequential Feature Selection, Support Vector Machine

1. INTRODUCTION

In recent times, there has been a witness to data explosion across several fields. The result of this has seen creation of varied datasets describing different illnesses and conditions. In the medical field, the continuous growth of biotechnology has seen the creation of modern devices and sophisticated biosensors that are now able to collect vast amounts of information from a person. For patients across the world receiving a quick and correct diagnosis is key for abating their suffering and in extreme cases evading death. Disease diagnosis should be a quick process that is also able to be done correctly and early on before extreme progression of an illness. However, for many patients this still proves elusive. Diagnosis for certain illnesses may take a long time or even suffer misdiagnosis. This is due to several reasons however not limited to stretch medical human resources, doctor biases and confusing symptoms. Lack of enough medical specialists in certain regions means their demand is high and hence complex diagnosis waiting on a specialist may take longer periods to complete.

This paper, proposes an application through which medical staff will have the ability to upload patient diagnostic data. Depending on the data the application will be able to diagnose up to 8 diseases and conditions using SVM based classification models. The number of diseases capable to diagnose can slowly be increased over time as more data becomes available and better prediction models are developed. If any misdiagnosis is made by the application and is later discovered, the application offers an option to correct the diagnosis by retraining the specific model by incorporating the new diagnostic data and the correct diagnosis, preventing future misdiagnosis of similar cases. SVM is a powerful machine learning method based on the Statistical Learning Theory (SLT) [1]. It is a supervised binary classification machine learning algorithm that classifies given data in one of two classes. It learns or trains by being given data already classified into two classes, hence its supervised nature. During training, an SVM creates a separation boundary referred to as the separation hyperplane between the two different classes of data. After training, unclassified data can then be fed to the SVM, where thus, it can only fall on either one side of the separating boundary and being classified [1]. In this application, SVMs will be used to classify a patient as positive or negative for a given condition based on specific diagnostic data obtained from the patient through different means.

2. RELATED WORK

Several approaches towards automating medical diagnosis using SVMs have been investigated by various researches such as in [2]-[11]. However, most of these have only focused on research and the creation of prediction models and thus there's has been limited development of practical applications and integrated prediction systems.

There still have been noted works that have been implemented practically. As an attempt to automate the medical diagnosis process, a project undertaken by Kampouraki [12] sought to develop a web based SVM for automatic medical diagnosis. This project would have medical staff add new diseases to the system from where they would input statistical diagnostic data then training it and be able to use it to make diagnosis for the particular disease. The limitation of this approach is that it would be the medical staff's job to perform model optimizations and evaluations of their trained models on their own and may not be adequately skilled to do so or even have the time to do so.

Another work performed by Juliet and Samy [13] sought to implement a smart health care monitoring system for elderly people. This project was implemented by continuous reading of data from different biosensors spread about a patient's body and through an SVM model identify abnormal conditions and have the system send an alert to near medical responders. A limitation to their project was the rigidity and lack of improvement in decision making should the system make a false decision.

Several of the mentioned studies performed using SVMs for disease diagnosis however have only been performed in the regard of one disease. There has been a gap in the development of automated SVM diagnosis systems combining diagnostic capabilities of different diseases into one system. There has also been little work done regarding the automatic learning from mistakes done by the SVM models allowing these systems and physicians to continue complimenting and learning from one another.

To remedy and improve, our proposed application utilizes pretrained and tested high accuracy models based on the various available datasets rather than have medical staff do this on their

own. Furthermore, as more diagnosis are made by the application the diagnostic data of each diagnosis is automatically stored growing the size of the existing datasets that can be used for future research works for improving the current prediction models. If any misdiagnosis occurs medical staff can query the specific patient diagnosis and correct it allowing for the application to retrain its specific prediction model incorporating the new data. Thus, allowing the application to continue improving its diagnosis and prediction accuracy continuously over time.

3. APPLICATION DESIGN AND SVM MODELS TRAINING

3.1 DESCRIPTION OF EXPERIMENTAL DATA

The disease diagnosis SVM prediction models were trained and tested using the following 7 datasets [14] - [20] obtained from the University of California, Irvine (UCI) Machine Learning Repository.

Dataset 1: The Chronic Kidney Disease (CKD) Dataset

This dataset [14] features diagnostic data of early stage chronic kidney disease collected from an Indian population. The dataset has a total of 400 samples each with 24 features. Of the features, 11 are numeric values while 13 are categorical values and certain samples do have certain missing data. Each sample is then classed in one of two classes, having CKD or not having CKD. This data set can then be used in the creation of model to diagnose a patient with chronic Kidney with the given features.

Dataset 2: Breast Cancer Diagnostic (BCD) Dataset

This dataset [18] was created to automate breast cancer detection from digitized images of a fine needle aspirate (FNA). It consists of data extracted from a digitized FNA image taken from 569 women whom each was detected with a mass within their breasts. The data has a total of 30 features which describe the characteristics of the cell nuclei present in the image. Each data sample is then placed as either malignant or benign. This data set can be used to train prediction models to accurately diagnose if a detected mass or lesion in a breast image is malignant signaling breast cancer or benign from an FNA image.

Dataset 3: Acute Inflammations Dataset

This data was created [20] to aid development of an algorithm which will perform the presumptive diagnosis of two diseases of urinary system. It features 120 samples each with 5 features that are used to diagnose two conditions at the same time. Of each of the 2 conditions there are 2 classes representing whether having the disease or not.

Dataset 4: Breast Cancer Coimbra Dataset

This data was created [19] to be able to identify a biomarker for breast cancer from a simple blood analysis. It has 116 samples each with 9 features classified into either having breast cancer or not.

Dataset 5: Lymphography Dataset

This dataset [15] features 148 samples each with 18 features. This dataset was introduced to train models to detect 3 cases within a lymphogram. Samples are classified into four classes; normal, malignant, metastases and fibrosis.

Dataset 6: Cardiotocography Dataset

This dataset [17] is used to train prediction models to identify the status of fetal health in a pregnant woman. The data is obtained from 2126 fetal cardiotocograms that were generated from an Electronic Fetal Monitor (EFM). The data was automatically processed and the respective diagnostic features measured. The dataset thus has a total 2126 samples and a total of 21 features. Each sample is classified into 1 of 10 different classes which describe the fetal current state and 1 of 3 classes which describe the overall health status of a fetus from normal, suspect and pathological. Once a model is trained using this data it can correctly make decisions regarding fetal health status saving on diagnosis time and allowing for prompt action during an emergency.

Dataset 7: Hepatocellular Carcinoma (HCC) dataset

This dataset [16] was introduced as a means of predicting the survival rate within one year of a patient diagnosed with HCC. It consists of data from 165 real patients each diagnosed with HCC. It has 49 features containing several demographics, risk factors, laboratory and overall survival features of a particular patient. Each patient is then classed accordingly whether they survived after one year. Being able to make accurate predictions can help analyze a patient's prognosis based on these features and accordingly make changes to try and improve their prognosis.

3.2 DATA PROCESSING

Some of the datasets acquired had missing values, categorical values in words and unshuffled data. To properly train SVM models, each dataset had to be processed accordingly so as to be fed to the SVM.

3.2.1 Imputing Missing Values:

To solve the problem of missing values within data sets, each missing value was imputed a value using the K-nearest neighbor technique. This technique enters data into an empty cell with data from the nearest neighboring column. If the nearest neighbor also has no data, it goes to the nearest neighbor of that cell and so forth. Depending on the dataset, missing values were also imputed using a value computed from adding or subtracting to the mean of a given feature using a randomly generated value that was within the variance of the feature.

3.2.2 Indexing Categorical String Data:

Certain features were categorical in nature and the data was recorded as strings, for example: "yes", "no", "Healthy" etc. SVMs only take numerical data, hence this categorical data was indexed where strings of the same category were replaced using an integer index. Once each categorical value was represented by an integer index, the data would now be entirely numerical and ready to be fed to the SVM trainer.

3.2.3 Cross Validation Partitioning:

To eliminate bias, overfitting and underfitting within trained models, the data was split into cross validation partitions. For the larger datasets having samples exceeding 400, the data was shuffled randomly then a small portion data was held out for testing. The remaining portion was then partitioned again 10 times using the *K*-fold technique and each fold was stratified ensuring an almost equal distribution of each class within each fold. Once

this was completed the data was ready to be fed to the SVM trainer after being standardized.

3.3 MODEL SELECTION

This step involved choosing the most suitable kernel function and values of the SVM model hyperparameters to achieve the highest possible accuracy.

3.3.1 Kernel Function and Feature Selection:

Kernel function selection was performed by using feature sets of ranked features by their classification strength. This was accomplished by training several models using different kernel functions with each and every feature set. Since each feature set contained a different number of features, the performance would vary for each kernel function and actively enable us to analyze the most influential set of features for a specific kernel function and the best overall kernel function. The best performing Kernel function and its best feature set was used to train the diagnosis models.

3.3.2 Hyperparameters Optimization:

To tune the values of Kernel gamma, γ, sigma σ and the penalizing factor C and try to get the best model performance the Bayesian optimization was used as the hyperparameter optimization technique. This was achieved using the Bayesian optimization algorithm within MATLAB. The obtained best Kernel function and feature set would then be used as the objective function for the optimizer. The optimizer was then allowed to iterate for 30-70 times. The generated results of the best observed penalty factor C and Kernel gamma were then selected as the hyperparameters to train the diagnosis model.

3.4 MODEL EVALUATION

This was the final step in model creation. To assess the performance and overall accuracy of the trained model, two key techniques were utilized.

3.4.1 Validation Accuracy:

For the smaller datasets which no data was held out. The model accuracy was calculated from the cross-validation loss of the model. As each data before training was partitioned using the K-Fold technique, each fold performance was computed using the test partition and then average was computed from all the folds. This gave an overall accuracy of the model. For the larger datasets having more than 400 samples of data a small portion 20-50% of the data was held out and used again to evaluate the model's accuracy. The accuracy was the percentage value of the number of times a model made a correct diagnosis over the total number of diagnosis it made.

3.4.2 Confusion Matrix:

As it might be difficult or perhaps impossible to achieve a model with 100% accuracy in all datasets, this technique was utilized to see the how each model would really perform in real life situations. An assumption was made that if a model was to perform a wrong diagnosis, it rather makes a false positive diagnosis where one is wrongly diagnosed of a disease they do not have over a false negative where a person actually suffering from a disease is wrongly diagnosed as not having the disease. Using the confusion matrix, one would be able to compare the false negative rate to the true positive rate. The model's accuracy may not be able to achieve a 100% but should have the lowest possible false negative rate. If it achieved a zero false negative rate the model was referred to as near perfect as it would be able to detect all cases of disease.

4. AUTOMATING THE DIAGNOSIS PROCESS

After training the models, they were then exported as compact models that would be utilized in making predictions and diagnosis in a standalone MATLAB graphical user interface application.

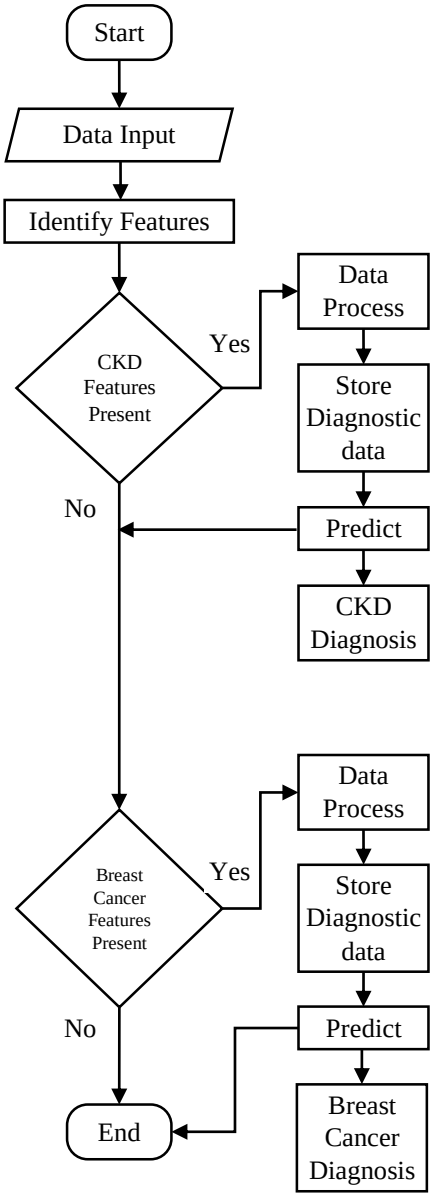


Fig.1. Simplified flow chart of the diagnosis process

The Fig.1 illustrate a flow chart of how the feature selection during diagnosis works. If data is uploaded in bulk for a particular the patient. The algorithm would detect features of a specific illness, preprocess them as required then use them to make a diagnosis. Once done for the first illnesses to goes on to the next illnesses and the processes goes on.

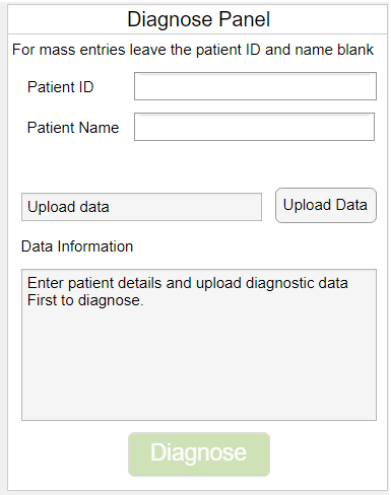


Fig.2. Data upload panel for the application

As observed above in Fig.2, a user would create a new patient entry entering their name and ID, then upload the patient's diagnostic data consisting features or attributes of the different conditions and illnesses. Once done a user presses the diagnose button. As illustrated in the flow chart shown in Fig.1, to diagnose, the application would first detect features that belong to a given illness, group them accordingly then perform any data processing techniques that are required for the particular set of features. Once done, the app would feed the required features to the respective SVM prediction model and then obtain predictions from the model which are then displayed on the user interface for the user to view. This can be viewed in Fig.4.

To work successfully, the application requires that the data to be fed in conforms to a specific format where all feature columns within the uploading file are named accordingly for each set of disease so that the app can be able to distinguish which feature belongs to which disease and specifically how to process that feature. The application also has a single patient view that allows one to view a collective disease report of only one patient. To allow learning from its mistakes of an incorrect diagnosis, a functionality was added allowing a user to mark a given prediction as wrong. This would trigger the application to retrain using new data which is the new input data and the original training set and improve individual patient overview of seeing an individual disease diagnosis report.

5. EXPERIMENTAL RESULTS

After training each model with results obtained from each section each model was done a final evaluation and the results recorded. The results are tabulated in Table.1. Within the table consists the validation accuracy obtain from the *K*-fold validation partitions and the model accuracy test set data. Another field included is the percentage accuracy of the held-out test data from the dataset. The test ratio column shows how much of the dataset was held out to be used as test set. The Table.2 displays the sensitivity, specificity and F1-scores of the 5 binary classified disease conditions.

Table.1. Model evaluation accuracy performance results

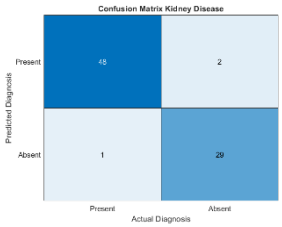
Dataset	Test Ratio	Validation Accuracy	Holdout Accuracy
Acute Inflammation: Bladder	50%	100%	100%
Acute Inflammation: Nephritis	50%	100%	100%
Chronic Kidney Disease	20%	99.46%	98.78%
Breast Cancer FNA	20%	98.5%	96.70%
Lymphography	20%	85.13%	94.3%
Cardiotocogram	20%	90.12%	92.24%
Breast Cancer Coimbra	20%	84.15%	82.35%
HCC Survival	20%	82.24%	77.56%

Table.2. Model evaluation accuracy performance results

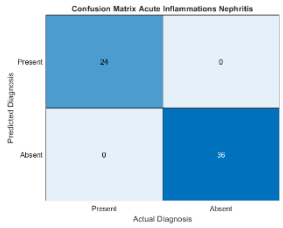
Dataset	Sensitivity	Specificity	F1-Score
Acute Inflammation: Bladder	100%	100%	1
Acute Inflammation: Nephritis	100%	100%	1
Chronic Kidney Disease	98%	93.6%	0.9697
Breast Cancer FNA	96.7%	97.1%	0.9749
Breast Cancer Coimbra	83.3%	100%	0.9091
HCC Survival	83.3%	75.7%	0.6452

5.1 CONFUSION MATRIX

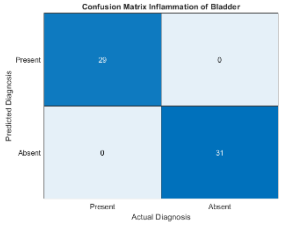
The Fig.3(a)-Fig.3(f) shows confusion matrices for the 6 datasets that were generated to view how accurately each dataset model performed. It can be noted the poor performance in the HCC dataset at Fig.3(f) which had a higher false positive rate and the excellent performance of the acute inflammation dataset represented in Fig.3(b) and Fig.3(c).



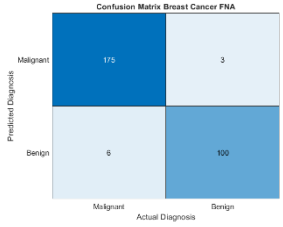
(a)



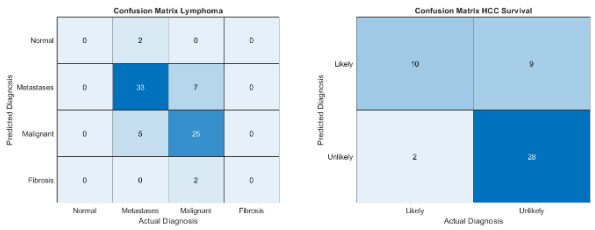
(b)



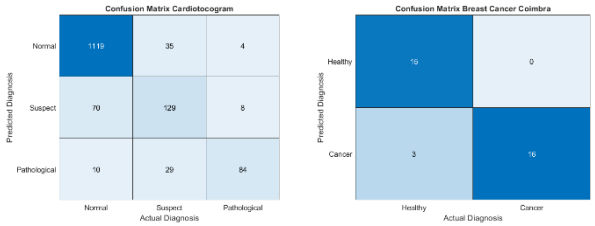
(c)



(d)



(e) (f)



(g) (h)

Fig.3. Confusion Matrices for the Various Datasets, (a) Chronic Kidney Disease (b) Inflammation of Bladder, (c) Nephritis, (d) Breast Cancer FNA, (e) Lymphography (f) HCC Survival (g) Cardiotocogram

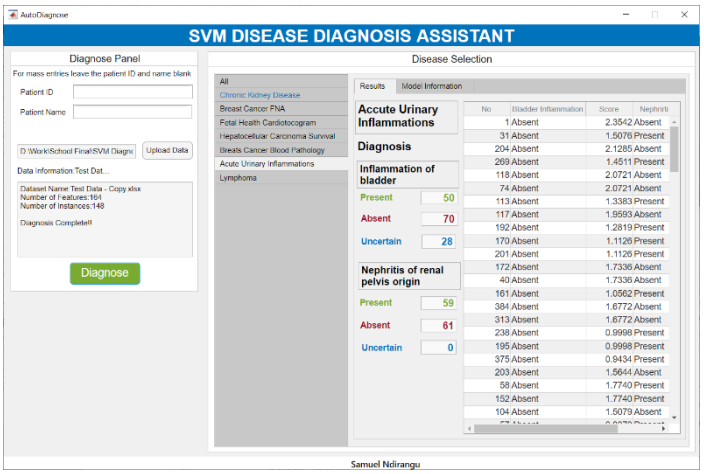


Fig.4. Diagnosis of inflammation conditions from models trained using the acute inflammations dataset.

In Fig.4, illustrates the application after performing a diagnosis on uploaded data of several patients. It features tabs of each disease where a user can be able to view a summary of all the diagnosis of patients regarding the specific disease.

The Fig.5 illustrates the application view for a single patient. A user is able to view each diagnosis for a particular patient by entering their unique number. To retrain the model in case of misdiagnosis a user enables the feature through a switch as illustrated in Fig.6. A user then selects the correct diagnosis then presses retrains which retrains the model incorporating the new data and allowing it to improve it diagnosis accuracy for similar future cases.

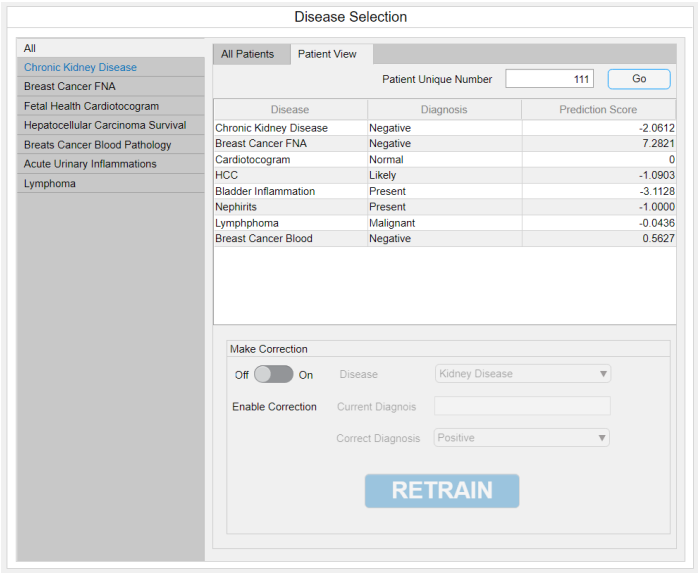


Fig.5. A single patient view of each diagnosis results and options to correct a wrong diagnosis.

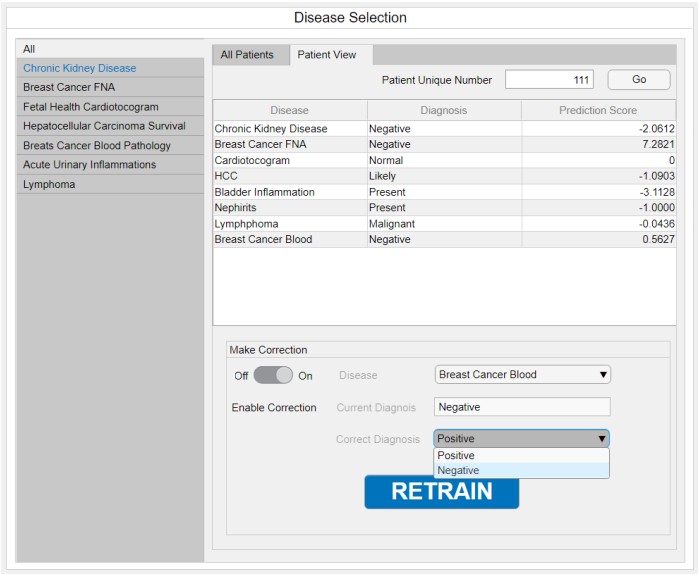


Fig.6. Selecting a patient and correcting misdiagnosis by pressing the retrain button.

The Fig.6 shows the view when the make correction panel is activated. A user selects a disease of the specific patient. It automatically displays the current diagnosis and offers options to select the new diagnosis and retrain the specific model

6. DISCUSSION

The model performances obtained were of high accuracy and satisfactory. The Acute Inflammation dataset, models trained were able to achieve a 100% accuracy in detecting the two conditions the data represented. The rest of the data sets were able to achieve over 75%. Ranking and grouping of features into different feature sets together with using the Bayesian optimizer proved to be highly beneficial in selecting kernels and the most optimal hyperparameters enabling us to yield high accuracy performing models. The automated disease diagnostic assistant

system was also able to perform as expected. Due to its ability to store diagnostic data of patients. It was easy to query a given diagnosis for reference and correction. After performing diagnosis correction on given previous given diagnosis and retraining. The application was fed the same diagnostic data as before to see if it would now predict the right diagnosis. In this regard it was able to predict correctly during the second time in most of the test cases.

7. CONCLUSION AND RECOMMENDATIONS

We have implemented an automated disease diagnosis assistant system using SVMs trained using available disease datasets. The application was implemented using the MATLAB software which was used to train the SVM models and create a graphical user interface through which users would easily interact with by feeding data and receiving a diagnosis from the system. Testing of the system yielded high accuracy and satisfactory results for most of the diseases. More needs to be done in diagnostic data collection that is varied across different races and population as current datasets tend to have a bias towards people from one locale. Other classification machine learning techniques that may offer better accuracy performance for certain datasets can also be explored and included within a diagnostic assistant improving its overall accuracy, practicality and reliability to medical diagnosis.

REFERENCES

[1] C. Corinna, and V. Vapnik. "Support-Vector Networks", *Machine Learning*, Vol. 20, No. 3, pp. 273-297, 1995.

[2] V.A. Kumari and R. Chitra, "Classification of Diabetes Disease using Support Vector Machine", *International Journal of Engineering Research and Applications*, Vol. 3, No. 2, pp. 1797-1801, 2013.

[3] Z. Gao, L. Po, W. Jiang, X. Zhao and H. Dong. "A Novel Computerized Method based on Support Vector Machine for Tongue Diagnosis", *Proceedings of 3rd International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pp. 849-854, 2007.

[4] H. Mezrigui, F. Theljani and K. Laabidi. "Decision Support System for Medical Diagnosis using a Kernel-Based Approach", *Proceedings of IEEE International Conference on Control, Automation and Diagnosis*, pp. 303-308, 2017.

[5] Emre Gurbuz and E. Kilic, "Diagnosis of Diabetes by using Adaptive SVM and Feature Selection", *Proceedings of IEEE 19th International Conference on Signal Processing and Communications Applications*, pp. 42-45, 2011.

[6] A.B. Rabeh, F. Benzarti and H. Amiri, "Diagnosis of Alzheimer diseases in early step using SVM (Support Vector Machine)", *Proceedings of IEEE 13th International Conference on Computer Graphics, Imaging and Visualization*, pp. 364-367, 2016.

[7] L. Huang, Z. Pan and H. Lu, "Automated Diagnosis of Alzheimer's Disease with Degenerate SVM-Based Adaboost", *Proceedings of IEEE 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 298-301, 2013.

[8] T. Mu and A.K. Nandi, "Detection of Breast Cancer using V-SVM and RBF Networks with Self-Organized Selection of Centers", *Proceedings of IEEE 3rd International Seminar on Medical Applications of Signal Processing*, pp. 47-52, 2005.

[9] C. Sowmiya and P. Sumitra, "Analytical Study of Heart Disease Diagnosis using Classification Techniques", *Proceedings of IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing*, pp. 1-5, 2017.

[10] H. Mezrigui, F. Theljani and K. Laabidi, "Decision Support System for Medical Diagnosis using a Kernel-Based Approach", *Proceedings of IEEE International Conference on Control, Automation and Diagnosis*, pp. 303-308, 2017.

[11] A. Singh, "Detection of Brain Tumor in MRI Images, using Combination of Fuzzy C-Means and SVM", *Proceedings of IEEE 2nd International Conference on Signal Processing and Integrated Networks*, pp. 98-102, 2015.

[12] A. Kampouraki, D. Vassis, P. Belsis and C. Skourlas, "E-Doctor: A Web-Based Support Vector Machine for Automatic Medical Diagnosis", *Procedia-Social and Behavioral Sciences*, Vol. 73, pp. 467-474, 2013.

[13] N.M.J. Augusstine and S.R.N. Samy, "Smart Healthcare Monitoring System using Support Vector Machine", *Australian Journal of Science and Technology*, Vol. 2, No. 3, pp. 1-8, 2018

[14] UCI Machine Learning Repository, "Chronic_Kidney_Disease Data Set", Available at: https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease