A cluster-genetic programming approach for detecting pulmonary tuberculosis

Adane Nega Tarekegn, Tamir Anteneh Alemu*, Alemu Kumlachew Tegegne

Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar University, Ethiopia

ABSTRACT

Tuberculosis (TB) remains a global health concern. It commonly spreads through the air and attacks low immune bodies. TB is the most common and known health problem in low and middle-income countries. Genetic programming (GP) is a machine learning model for discovering useful relationships among the variables in complex clinical data. It is more appropriate in a circumstance when the form of the solution model is unknown a priori. The main objective of this study was to develop a model that can detect positive cases of TB suspected patients using genetic programming approach. In this paper, Genetic Programming (GP) is exploited to identify the presence of positive cases of tuberculosis from the real data set of TB suspects and hospitalized patients. First, the dataset is pre-processed, and target variables are identified using cluster analysis. This data-driven cluster analysis identifies two distinct clusters of patients, representing TB positive and TB negative. Then, GP is trained using the training datasets to construct a prediction model and tested with a separate new dataset. With the 30 runs, the median performance of GP on test data was good (sensitivity=0.78, specificity=0.95, accuracy=0.89, AUC=0.91). We find that GP shows better performance in predicting TB compared to other machine learning models. The study demonstrates that the GP model might be used to support clinicians to screen TB patients.

Keywords: Cluster analysis, Genetic programming, Tuberculosis, Prediction
DOI: https://dx.doi.org/10.4314/ejst.v14i1.5

INTRODUCTION

Tuberculosis (TB) continues to be a substantial global problem as it is a common and deadly infectious disease that can occur at any age. It was assessed that 10 million persons world-wide were newly infected in 2017, including 3.2 million women, 5.8 million men and 1 million children (WHO, 2018). TB is caused by

[*] Corresponding author: tamirat.1216@gmail.com

single pathogen infection, and its mortality rate in 2017 reached 16% (World Health Organisation, 2018). Early screening and diagnosis of TB is the most important accomplishment that should be considered (Wang, 2019). However, the occurrence of TB infection is commonly challenging to predict, and delays in identification and diagnosis are common. Delay in diagnosis may lead to drug resistance, multi drug resistance (MDR), where an isolate shows resistance to two first line drugs, rifampicin and isoniazid, and extensive drug resistance (XDR) which include multiple drug resistance (MDR) and also show resistance to fluoroquinolones and at least one of the injectable drugs (Gandhi *et al.*., 2010). In medical science, lab tests require expensive microscopic examination of sputum and other profiles of patients. Identifying positive cases of tuberculosis is often complicated and time-consuming. The symptoms of patients are usually unclear, and the similarities in symptoms of some tuberculosis diseases are difficult to distinguish based on decision boundaries or discriminating rules. This creates many difficulties in reaching the right decision or diagnosis. In spite of the challenges, the diseases require rather immediate medical treatment to prevent serious consequences.

Machine learning approaches can support health professionals in making decisions for the diagnosis or prediction of TB. Many statistical and machine learning methods have been used for modelling and prediction of TB disease (Mello *et al.*., 2006; Aguiar *et al.*., 2012; Bobak *et al.*., 2019; Khan *et al.*., 2019). These techniques include logistic regression, neural network, support vector machines, decision trees, naïve Bayes, etc. Despite the contribution of these techniques, there are still some issues that prevent them from becoming adopted in modelling practical problems. One of the main limitations of the traditional methods is that a specific model form must be assumed that demands strong theoretical knowledge. For example, in a regression problem, the task is often limited to finding a set of model coefficients for the linear or polynomial functions that best describe the input variables. Moreover, prior knowledge about the statistical distribution of the data is essential in such models. Evolutionary algorithms can be used as a remedy for solving highly complex, nonlinear problems (Smith, 2018). Genetic programming (GP) is one of the evolutionary algorithms that allows searching for a suitable model more differently and intelligently. It is a general methodology for the development of mathematical models rather than a specific technique for solving particular problems (Angeline, 1994; Bannister *et al.*., 2018). In modelling or supervised learning, GP is preferable to other machine learning methods in circumstances where the form of the solution model is unknown a priori. GP has been successful in automatically evolving variable-length computer programs to solve medical problems (Hu *et al.*., 2015; Wang *et al.*., 2017).

In this paper, a cluster analysis followed by genetic programming (GP) is proposed for modelling and prediction of pulmonary TB. In particular, GP is suitable for detecting the presence of positive cases of tuberculosis based on data collected from TB suspects and hospitalized patients. Detection of tuberculosis cases is a challenging task due to the presence of nonlinear interactions between many variables. One main reason for using GP is to get the advantage of its global search mechanism in a considerable space of possible solutions. Many decision tree generation algorithms (e.g., CART and C4.5) perform a greedy local search to generate classification rules, while GP performs a more global search through the space of a large number of possible solutions. To date, various literature on TB prediction pays special attention to the traditional, statistical and machine learning methods to predict TB; however, evolutionary algorithms, such as GP, could also have the capability to model TB problems.

METHODS

Study framework

The study adopted an approach that comprised a combination of two phases. The first phase is data preparation and cluster analysis (Figure 1). First, data were pre-processed, i.e., data cleaning, integration, and feature selection. Then multiple correspondence and clustering analyses were done to identify homogenous groups of cases in the dataset. In this first phase, the study aimed to explore the potential presence of coherent clusters of patients. The high density-based spatial clustering of applications with noise (HDBSCAN) algorithm was used to discover the patient groups from a data-driven perspective. This stage realised the discovery of two well-separated patient clusters. The second phase includes the development of machine learning models that can make predictions on the clustered data. Specifically, a genetic programming model was built and trained for predicting tuberculosis cases. This phase also includes the learning and evaluation phases. The learning phase is designed to evolve the GP classification model for the tuberculosis dataset, which comprises train GP with the training dataset and test GP with a separate test dataset. Finally, the fitness of the evolved model is measured with an appropriate metric.

Data collection and preparation

The data used for this experiment was taken from Menelik II hospital, Addis Ababa. The data was collected from TB suspects and hospitalized patients for analysis and prediction. The patients' real data contains 4241 instances and 25 attributes recorded for administrative purposes. All the needed data pre-processing steps such as data cleaning, transformation and feature selection activities were performed before analysis, and only 13 relevant attributes were selected for this experiment. The following attributes were excluded in the analysis: medical record number, unit TB number, name of patient, address of patient, name of contact person, address of contact person, laboratory number, drug, month, CPT started date, enrolled in HIV care, since they had no important effect in the prediction. The final selected attributes included in the analysis are shown in Table 1.

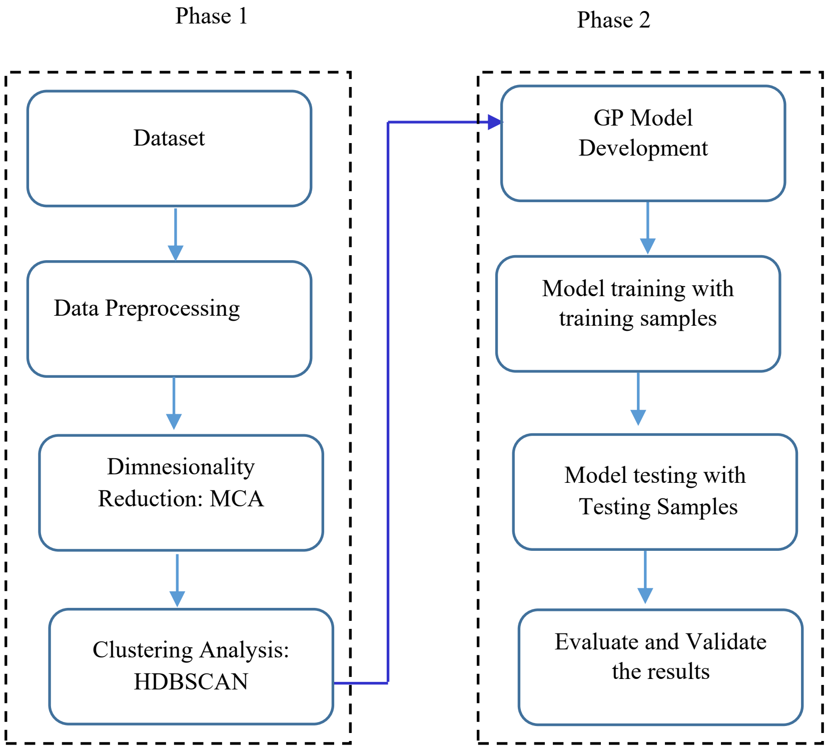
Phase 1
Phase 2
Dataset
Data Preprocessing
Dimnesionality Reduction: MCA
Clustering Analysis: HDBSCAN
GP Model Development
Model training with training samples
Model testing with Testing Samples
Evaluate and Validate the results

Figure 1. The proposed framework

Table 1. Final selected variables and their description

No.	Variable	Description
1	Sex	Patient’s sex
2	Age	Patient’s age in years
3	Weight	Weight of the patient
4	Exhaustion	A weakness of the patient
5	HIV performed	Patient tested for HIV
6	HIV test result	HIV test result of the patient
7	Headache	Whether the patient has a headache
8	Cough	Cough for about 2 weeks
9	Chest pain	Some pain around the chest
10	Bloody sputum	Sputum mixed with a blood
11	Fever	An expected increase in temperature
12	Weight loss	Whether the patient reduced in weight
13	Night sweats	Whether the patient has sweats

Clustering approach

The study aims to detect patterns in the patients' data and to classify individuals at risk of tuberculosis (TB) correctly. However, since the data doesn’t contain any identified number of groups, it was not feasible to apply classification algorithms directly on the dataset. Therefore, clustering analysis was first considered as a pre-processing step for classifying patients into different groups. We identified subgroups TB patients that have similar characteristics using the following three steps:

1. Multiple Correspondence Analysis (MCA), a data analysis method designed for categorical variables, was used to detect underlying structures in the data set. It is an extension of correspondence analysis (Greenacre, 2015) for multivariate datasets which projects a given dataset in a lower-dimensional subspace producing two major effects: It reduces the dimensionality of the dataset, and it projects the observations on continuous space. In our study, the transformed dataset contains two numerical dimensions derived from 13 categorical variables. Then, the resulting data has joined the continuous variables which were ready for the process of clustering analysis.
2. Min –Max Normalization**:** It is common practice to normalize the data before clustering in case that the range of features values varies widely, and the relationship between each feature is unknown. Many studies in the literature argued that large variations within the range of feature values could affect the quality of clusters (Visalakshi and Thangavel, 2009). In our case, after transforming the data into low-dimensional representation using MCA, the newly created artificial features were rescaled to constrain dataset values to a

standard range. The min-max normalization method was used, where each feature was rescaled to the [0, 1] interval. The values were transformed based on the formula shown below (Elbattah and Molloy, 2017):

$$Z = \frac{x - \min(x)}{[\max(x) - \min(x)]} \qquad \text{Eq. (1)}$$

3. Hierarchical Density-based Spatial Clustering of Applications with Noise (HDBSCAN): From the geometric space created in Multiple Correspondence Analysis (MCA), TB patients were grouped into clusters using HDBSCAN. HDBSCAN algorithm extends DBSCAN by converting it into a hierarchical clustering algorithm and then using a technique to extract a flat clustering based on the stability of clusters (McInnes and Healy, 2017). It is especially good at finding oddly shaped clusters or more dense regions of a dataset that are surrounded by other lower density regions, in which the partitioning clustering methods such as k-means might have difficulty in doing it. Figure 2 depicts the results of clustering using HDBSCAN via MCA, in which two well- separated clusters have been produced.

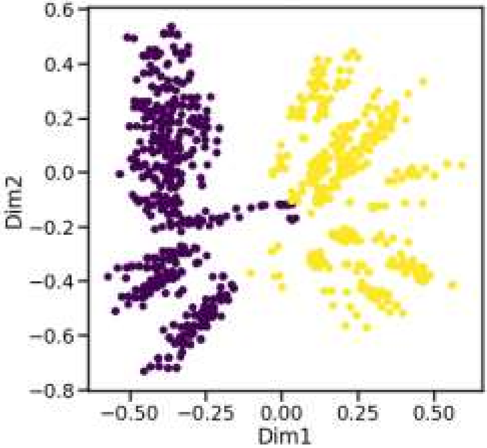

Figure 2. Two-dimensional visualization of clustering results using HDBSCAN

Using silhouette analysis, we determined the degree of separation between clusters as well as the compactness within each cluster. The cluster outcome showed a silhouette measure of cohesion and separation of 0.73, indicating that it is a substantial cluster solution.

Validation and stability analysis

Stability measure is one method of validation for a learning system (Johnson and Omland, 2004; Lange *et al.*, 2004). In the domain of machine learning, a learning system is stable if it can generate consistent results with respect to small perturbation of training samples, such as sub-sampling or the addition of noise (Pascual *et al.*, 2010). In this study, two major consecutive learning systems have been employed, i.e., clustering followed by the classification, and therefore, stability can be viewed from two perspectives: clustering stability and classification stability. Here, we used the validation/stability approach for these two learning paradigms that are in line with current practices in the field.

A) Validation of clustering results

Validation of clustering results is, generally, one of the most difficult problems that remain unsolved, as there is no ground truth to compare with (Kleinberg, 2003; Nascimento *et al.*, 2003). However, various methods have been suggested in the literature, including, external validity, internal validity, relative criteria (Xu, 2005), and stability approaches (Rakhlin and Caponnetto, 2007). The internal measure, which is the most commonly used approach in the literature and the stability approaches, have been adopted in this study. Silhouette analysis is one of the most popular and an effective internal measure which allows evaluating the appropriateness of the assignment of a data object to a cluster by measuring both intra-cluster cohesion and inter-cluster separation. Clusters within the range of 51% to 70% and 71% to 100%, respectively, indicate that a reasonable and a strong intra-cluster cohesion and inter-cluster separation are found (Lv *et al.*, 2016). The silhouette score can take values in the interval [-1, 1]. Negative silhouette values represent wrong data placements, while positive silhouette values better data assignments. Therefore, we want the scores to be as big as possible and close to 1 to have good clusters. In our experiments, the cluster outcome showed a silhouette measure of cohesion and separation of 0.73, indicating that it is a plausible cluster solution.

We also propose to measure the stability of the clustering results through statistical significance. Differences in characteristics between clusters were compared with respect to all the variables, using Pearson Chi-square tests. The significance level was set at $\alpha = 0.05$ and all tests were two-tailed. All input variables varied significantly between clusters (all p-values <0.05), except three variables, namely sex, weight, and bloody sputum with p-values 0.36, 0.61, and 0.36, respectively. Table 3 presents detailed information in the two clusters (clusters 1 and 2) and statistical test results between the two clusters with respect to all the input variables.

Table 2. Comparison of input variables between the two clusters.

Variable	Code	Cluster 1		Cluster 2		χ^2	DF	CV	P-value
		n	%	n	%				
Sex	0	534	43.70	478	45.61	0.83	1	3.84	0.36
	1	688	56.30	570	54.39				
Age	0	138	11.29	8	0.76	0.21	2	5.99	0.00
	1	909	74.39	937	89.41				
	2	175	14.32	103	9.83				
Weight	0	118	9.66	95	9.06	1.00	2	5.99	0.61
	1	743	60.80	624	59.54				
	2	361	29.54	329	31.39				
Year	0	90	7.37	97	9.26	9.82	3	7.81	0.02
	1	312	25.53	218	20.80				
	2	479	39.20	449	42.84				
	3	341	27.91	284	27.10				
HIV performed	0	14	1.15	45	4.29	22.09	1	3.84	0.00
	1	1208	98.85	1003	95.71				
HIV test result	0	1218	99.67			0.16	1	3.84	0.00
	1	4	0.33	44	4.20				
	2			1004	95.80				
Headache	0	624	51.06	479	45.71	6.48	1	3.84	0.01
	1	598	48.94	569	54.29				
Cough	0	939	76.84	763	72.81	4.90	1	3.84	0.03
	1	283	23.16	285	27.19				
Chest pain	0	902	73.81	723	68.99	6.46	1	3.84	0.01
	1	320	26.19	325	31.01				
Bloody sputum	0	666	54.50	551	52.58	0.84	1	3.84	0.36
	1	556	45.50	497	47.42				
Fever	0	681	55.73	456	43.51	33.68	1	3.84	0.00
	1	541	44.27	592	56.49				
Night sweating	0	348	28.48	239	22.81	9.47	1	3.84	0.00
	1	874	71.52	809	77.19				

*Comparison of the variables between clusters using Pearson's Chi-square test; χ^2: Chi-square statistic; DF: Degrees of freedom; CV: Critical Value; n: number of feature values in each cluster; %: percentage of values in each cluster

B) Validation of classification results

The primary purpose of a classification model is to increase our understanding of the current world (e.g., identify risk factors for infection) or make predictions about the future (e.g., predict who will become infected) (Adane Tarekegn *et al.*., 2020). Validation of such models is at the core of machine learning. There are different approaches to assessing the performance or stability of machine learning models, such as hypothesis stability, error stability, and leave-one-out cross-validation stability (Elisseeff and Pontil, 2003). The aim of all these different approaches is to estimate the generalization error.

Currently, the well adopted and commonly used measure of generalization error is the cross-validation approach. In this study, we adopted the k-fold cross-validation, where the training set is split into several subsamples. In particular, we employed 10-fold cross-validation aiming to tune the parameters of the model and reduce the variance of the resulting generalization estimate by averaging over ten different subsamples. This 10-fold cross-validation can deal with limitations of the holdout method, such as to reduce overfitting, and therefore is more reliable and provides better generalization performance on the test data (Adane Tarekegn *et al.*, 2020).

In genetic programming (GP), the classification model that we adopted here, validation is just more than that of the 10-fold cross-validation. Because GP is a probabilistic stochastic search algorithm, that is, even if all the parameters are the same, the results are not the same for each operation of each subsample. Stochastic characteristic enables GP to explore the solutions space of optimization problem and realize the diversification and exploration. Without the randomness, GP usually will converge to a local optimum very quickly (Pétrowski and Ben Hamida, 2017). Therefore, to measure the stability of the GP model across its stochastic nature, we did 30 runs and take the best and median of the solutions. The details of the whole validation process for the GP model are presented in subsequent sections (sections 3).

Experimental settings

After having identified the target variable through the process of clustering analysis, the dataset was randomly split into training and testing. The GP model is trained using the proportions of 75% the training and 25% for testing. First, the 10-folds cross-validation approach is applied to the training set (i.e., on the 75%). The training set is used to train and optimize the model and includes both input data and the corresponding expected output. The testing set, on the other hand, includes only input data, not the corresponding expected output. The testing data is used to assess how well the algorithm was trained and to estimate model properties. The experiment includes learning a binary classification of data to TB positive and TB negative by considering the profiles of each individual patient.

GP parameter setup

In GP, setting the control parameters is an important first step to manipulate data and to obtain good results. In our problem, we tried several experiments for classification tasks by using the control parameters, such as population size, selection method, number of elite individuals, initialization method, number of generations, crossover probability rates, and mutation probability rates. Due to the stochastic nature of GP, 30 runs were performed in all problems, each with a different random number generator seed. The selection mechanism has been the tournament selection and the maximum tree depth set to the default value. GP requires that further control parameters be specified. The common parameter settings with their values that were used for this experiment are listed in Table 3.

Table 3. GP control parameter settings

Parameter	Value
Population size	1000
Maximum number of generations	100
Crossover probability	0.90
Mutation probability	0.15
Selection method	Tournament selection
Termination Condition	Max generation
Tree initialization	Ramped half and half
Genetic operators	Crossover, Mutation
Elites	1

Population size: is the actual number of individuals in a population.

Maximum number of generations: are fixed based on some trial runs. However, it depends upon the population size, preciseness of definition of objective function and constraints, use of real valued or binary valued chromosomes, method of selecting chromosomes for reproduction, crossover type (single point/multipoint), crossover rate, mutation rate, etc.

Crossover probability: is the probability that crossover will occur at a particular mating; that is, not all mating must reproduce by crossover, but one could choose Pc=1.0.

Mutation probability: After the offspring are generated from the selection and crossover, the offspring chromosomes may be mutated. Like crossover, there is a mutation probability. If a randomly selected floating-point value is less than the mutation probability, mutation is performed on the offspring; otherwise, no mutation occurs.

Selection method: Usually with a larger population, not all units of observation can be analysed. Therefore, a smaller sample is drawn from the population, whereby the survey results are representative of the entire population.

Termination condition: is an expression or a mathematical equation consisting of variables, constants, operators, and common functions that limit or define movement.

Tree initialization: A tree is a nonlinear data structure, compared to arrays, linked lists, stacks and queues which are linear data structures. A tree can be empty with no nodes or a tree is a structure consisting of one node called the root and zero or one or more sub trees.

Genetic operators: is an operator used in genetic algorithms to guide the algorithm towards a solution to a given problem. There are three main types of operators (mutation, crossover and selection), which must work in conjunction with one another in order for the algorithm to be successful.

Elites: It is the choice or best of anything considered collectively, as of a group or class.

Evaluation metrics

In GP, fitness function defines a measure to calculate the accuracy of a solution by comparing the predicted class labels with the actual class labels. In the two-class classification problem, the outcome of classification performance can be represented by a confusion matrix shown below:

Outcomes of a two-class classification problem

Actual positive class	True positive (TP)	False negative (FN)
Actual negative class	False positive (FP)	True negative (TN)

Then, the following performance metrics are obtained from the confusion matrix, as shown in equations 2-4. The prediction model obtained from GP was evaluated in terms of overall accuracy, AUC, sensitivity, and specificity. In the context of this study, sensitivity measures the percentage of subjects who are correctly identified as having the event, i.e., TB positive, while specificity refers to the percentage of subjects who are correctly identified as not having the event, i.e., TB negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \qquad Eq.(2)$$

$$\text{Sensitivity } = \frac{TP}{TP + FN} \qquad Eq.(3)$$

$$\text{Specificty } = \frac{TN}{TN + FP} \qquad Eq.(4)$$

The major limitation of accuracy and other measures is that they represent the performance of a solution when it is evaluated using a single class threshold. In contrast, the area under the ROC curve (or AUC) measures the classification performance at multiple class thresholds. The AUC measures the overall quality of a classifier when the threshold parameter biasing the final classification decision is varied (Urvesh Bhowan et al., 2002; Kumar and Indrayan, 2011).

$$AUC = \sum_{i=1}^{N-1} \frac{1}{2}(FP_{i+1} - FP_i)(TP_{i+1} + TP_i) \qquad \text{Eq. (5)}$$

where N is the number of thresholds, and TP_i / FP_i represents the performance of the solution at class threshold i. The equation sums the area of the individual trapezoids fitted under the ROC points. The AUC corresponds to the probability that a minority class example is correctly predicted across different class thresholds (Hajian-Tilaki, 2013). The AUC is a particularly useful and common measure of performance in classification tasks with unbalanced data as it represents how well a learned classifier approximates the trade-off between the minority and the majority classes across multiple classification thresholds.

True positive is an outcome where the model correctly predicts the positive class.
True negative is an outcome where the model correctly predicts the negative class.
False positive is an outcome where the model incorrectly predicts the positive class.
False negative is an error in which a test result improperly indicates no presence of a condition (the result is negative), when in reality it is present.

RESULTS AND DISCUSSION

GP model selection

In the GP process, the first task is to evaluate the quality of the generated model. This quality is called the fitness of a solution candidate. In GP based classifier, there are multiple possible ways to compute the quality of a model. In this paper, the mean squared error function (MSE) is considered, which calculates the average value of the squared residuals of the estimated values and original values. The population dynamics across generations are also evaluated based on this mean squared error. With the results stored from 30 runs of GP, we calculated the average fitness of the best solution per generation. Figure 3 and 4 show the best and median MSE on the test data at each generation over the 30 runs. Best fitness refers to the fitness of the best individual in the current population and the average fitness is simply the mean of the fitness values across the entire population. The evolution of the error in both average and best fitness reveals the ability of GP in learning the relationship between variables. There is a constant reduction in the test error across generations, indicating that no overfitting is occurring.

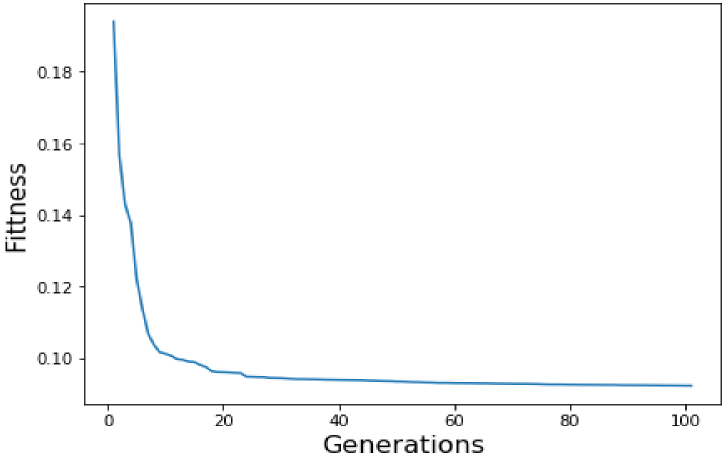

Figure 3. GP evolution plot. The line represents the best MSE on the test set.

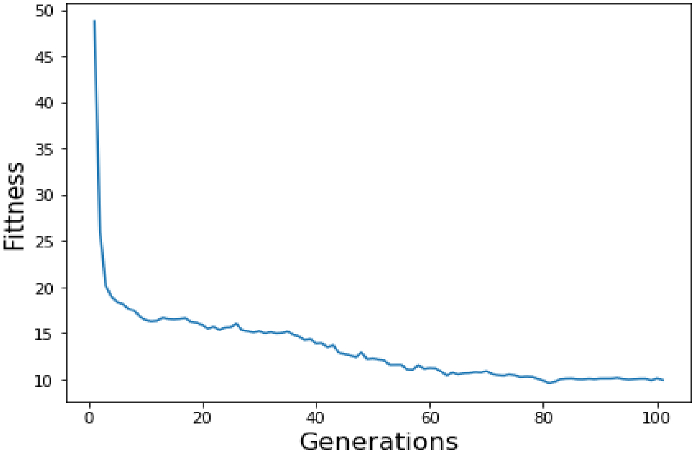

Figure 4. GP evolution plot. The line represents the average MSE on the test set.

Among the 30 best solutions, we selected a prediction model that produced the lowest MSE on the test set. The selected solution obtained the best value of 0.092 and its average is 9.96, as shown in Figures 3 and 4. The final model produced by the GP model included 7 predictors: bloody-sputum, night-sweats, chest-pain, and cough, HIV-test-result, night-sweats and headache. These variables were the most frequent variables which were the most relevant for the prediction of pulmonary TB. The final prediction model generated by GP was represented using the following equation:

$$\begin{aligned} Y = if(&\mathrm{c0.headache} > (\mathrm{if}((\mathrm{c1.fever} < \mathrm{c2}), \mathrm{c3.hiv} - \mathrm{test} - \mathrm{result}, \\ &\mathrm{c4.headache}) - \mathrm{c5.bloody} - \mathrm{sputum})), \\ &\mathrm{c6.hiv} - \mathrm{test} - \mathrm{result}, \\ &\mathrm{if}((\mathrm{c7.fever} < \mathrm{c8.hiv} - \mathrm{test} - \mathrm{result}), \\ &(\mathrm{c9} - \mathrm{c10.weight}), \\ &\mathrm{c11.headache})/(\mathrm{c12} - \mathrm{c13.weight})) \end{aligned}$$

Where c0=2.42, c1=0.62, c2=0.71, c3= .88, c4=0.89, c5=2.42, c6=0.62, c7=0.90, c8=0.94, c9=-0.12, c10=1.2, c11=0.31, c12=0.77, c13=0.81, c14=-0.13, c15=0.15, c16=1.15, c17=1.08

As shown in the equation, some variables were missing due to that GP performs an implicit features selection. The fact that GP required fewer predictors to achieve the required performance may have an advantage in the practical application of the developed TB prediction model.

GP model performance

In analysing GP for classification, the most fundamental aspect is to know the number of samples that are classified correctly and those, which are classified incorrectly. The results averaged from 30 runs of GP experiments are presented in Table 4 on the training set and Table 5 on the testing set. The classification performance is measured using sensitivity, specificity, overall accuracy and AUC.

For a fair comparison of GP with other machine learning methods, we used the Wilcoxon signed-rank test. From the results, we understood that for α =0.01 significance level, GP showed competitive results in performance compared to support vector machine, neural networks, and random forest. Finally, the performance of the selected GP model on the training together with other machine learning methods are shown in Table 4 and Table 5 in terms of accuracy, sensitivity, specificity, and AUC.

Table 4. Results of GP and other classifiers on training data

Classifier	Sensitivity	Specificity	Accuracy	AUC
Support vector machine	74.69	96.27	89.62	89.71
Random forest	87.34	96.68	93.80	89.65
Artificial neural network	74.93	96.59	89.84	89.01
GP (Max)	77.19	96.98	91.09	90.00
GP(Median)	76.59	96.02	89.94	89.58

Table 5. Results of GP and other classifiers on test data

Classifier	Sensitivity	Specificity	Accuracy	AUC
Support vector machine	76.94	95.73	89.82	90.03
Random forest	75.14	91.17	86.52	88.93
Artificial neural network	76.94	95.87	89.91	90.05
GP (Max)	73.92	96.99	89.99	89.92
GP (Median)	72.84	96.23	89.12	88.82

CONCLUSION

The combined clustering /genetic programming approach implemented in the study can draw more attention to the significance of patient clustering while dealing with prediction-related problems. The clustering-aided approach produced further concerns that can contribute to improving the accuracy of predicting patient outcomes.

GP is used as a potential tool for developing a prediction model for the tuberculosis dataset. The performance of the model obtained by GP is evaluated using sensitivity, specificity, accuracy and AUC. From the results obtained, it is evident that GP algorithms perform well in separating the positive cases from the negative cases of the TB disease. The overall classification accuracy for both training and testing is comparable with the well-accepted existing machine learning techniques like artificial neural network and support vector machines with considerable additional advantages. However, the computationally intensive nature of genetic programming makes it difficult to apply to the real-world problems with large amounts of datasets. So, further research is recommended to accelerate the time-consuming fitness evaluation step.

ACKNOWLEDGMENTS

The author would like to thank Dr Minichil Ayalew for his support and technical expertise with the data collection and understanding. The authors would like to thank the reviewers of this paper for their supportive comments.

Availability of data: The whole data set may be available on-demand from the corresponding author for only research purpose.

Sources of funding: No funding was received for this study.

Conflict of Interest: The authors declare that they have no competing interests

REFERENCES

Adane Tarekegn, Ricceri, F., Costa, G., Ferracin, E and Giacobini, M. (2020). Predictive modeling for frailty conditions in elderly people: Machine learning approaches. *JMIR Medical Informatics* **8**(6): e16678.

Aguiar, F.S., Almeida, L.L., Ruffino-Netto, A., Kritski, A.L., Mello, F.C.Q and Werneck, G.L. (2012). Classification and regression tree (CART) model to predict pulmonary tuberculosis in hospitalized patients. *BMC Pulmonary Medicine* **12**(1): 40. https://doi.org/10.1186/1471-2466-12-40.

Angeline, P.J. (1994). Genetic programming: On the programming of computers by means of natural selection. *Biosystems* **33**(1): 69–73. https://doi.org/10.1016/0303-2647(94)90062-0

Bannister, C.A., Halcox, J.P., Currie, C.J., Preece, A and Spasić, I. (2018). A genetic programming approach to development of clinical prediction models: A case study in symptomatic cardiovascular disease. *PLoS ONE* **13**(9): e0202685. https://doi.org/10.1371/journal.pone.0202685

Bobak, C.A., Titus, A.J and Hill, J.E. (2019). Comparison of common machine learning models for classification of tuberculosis using transcriptional biomarkers from integrated datasets. *Applied Soft Computing* **74**: 264–273. https://doi.org/10.1016/j.asoc.2018.10.005

Elbattah, M and Molloy, O. (2017). Clustering-aided approach for predicting patient outcomes with application to elderly healthcare in Ireland. *AAAI Workshop - Technical Report*. National University of Ireland Galway.

Elisseeff, A and Pontil, M. (2003). Leave-one-out error and stability of learning algorithms with applications. *NATO Science Series Sub Series III Computer and Systems Sciences* **190**: 111-130.

Gandhi, N.R., Nunn, P., Dheda, K., Schaaf, H.S., Zignol, M., van Soolingen, D and Bayona, J. (2010). Multidrug-resistant and extensively drug-resistant tuberculosis: a threat to global control of tuberculosis. *The Lancet* **375**(9728): 1830-1843. https://doi.org/10.1016/S0140-6736(10)60410-2

Greenacre, M. (2015). Correspondence analysis. In: *International encyclopedia of the* *social and behavioral sciences: Second edition*. https://doi.org/10.1016/B978-0-08-097086-8.42005-2

Hajian-Tilaki, K. (2013). Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian Journal of Internal Medicine* **4**(2): 627.

Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E.W.T and Liu, M. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing Journal* **36**: 534-551. https://doi.org/10.1016/j.asoc.2015.07.008

Johnson, J.B and Omland, K.S. (2004). Model selection in ecology and evolution. https://doi.org/10.1016/j.tree.2003.10.013.

Khan, M.T., Kaushik, A.C., Ji, L., Malik, S.I., Ali, S and Wei, D.Q. (2019). Artificial Neural Networks for Prediction of Tuberculosis Disease. *Frontiers in Microbiology* **10:** 395 https://doi.org/10.3389/fmicb.2019.00395

Kleinberg, J. (2003). An impossibility theorem for clustering. In: *Advances in Neural Information Processing Systems.* (P.463–470),2002,ACM, NIPS.

Kumar, R and Indrayan, A. (2011). Receiver operating characteristic (ROC) curve for medical researchers. *Indian Pediatrics* **48**(4): 277-287. https://doi.org/10.1007/s13312-011-0055-4

Lange, T., Roth, V., Braun, M.L and Buhmann, J.M. (2004). Stability-based validation of clustering solutions. *Neural Computation* **16**(6): 1299-1323.

Pascual, D., Pla, F and Sánchez, J. S. (2010). Cluster validation using information stability measures. *Pattern Recognition Letters* **31**(6): 454-461.

Lv, Y., Ma, T., Tang, M., Cao, J., Tian, Y., Al-Dhelaan, A and Al-Rodhaan, M. (2016). An efficient and scalable density-based clustering algorithm for datasets with complex structures. *Neurocomputing* **171**: 9-22.

McInnes, L and Healy, J. (2017). Accelerated hierarchical density based clustering. *IEEE international conference on data mining workshops*, *ICDMW*. pp. 33-42. https://doi.org/10.1109/ICDMW.2017.12

Mello, F.C. de Queiroz Mello., Bastos, L.G. do Valle Bastos., Soares, S.L.M., Rezende, V.M.C., Conde, M.B., Chaisson, R.E and Werneck, G.L. (2006). Predicting smear negative pulmonary tuberculosis with classification trees and logistic regression: A cross-sectional study. *BMC Public Health* **6**(1): 43. https://doi.org/10.1186/1471-2458-6-43.

Nascimento, S., Mirkin, B and Moura-Pires, F. (2003). Modeling proportional membership in fuzzy clustering. *IEEE Transactions on fuzzy systems* **11**(2): 173-186.

Pascual, D., Pla, F and Sánchez, J.S. (2010). Cluster validation using information stability measures. *Pattern Recognition Letters* **31**(6): 454-461.

Pétrowski, A and Ben Hamida, S. (2017). Genetic programming for machine learning. In: *Evolutionary algorithms*. pp. 183–216 John Wiley & Sons, Inc., Hoboken, NJ, USA. https://doi.org/10.1002/9781119136378.ch6.

Rakhlin, A., Caponnetto, A. (2007). Stability of K-means clustering. In: *Advances in neural information processing systems*. https://doi.org/10.1007/978-3-540-72927-3-4.

Smith, S. L. (2018). Medical applications of evolutionary computation. *Proceedings of the genetic and evolutionary computation conference companion on - GECCO* **18**: 1141–1169. https://doi.org/10.1145/3205651.3207873

Urvesh Bhowan,Mengjie Zhang, Mark Johnston (2002). Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence and Robotics* **85**:173.

Visalakshi, N.K and Thangavel, K. (2009). Impact of normalization in distributed K-means clustering. *International Journal of Soft Computing* **4(**4): 168-172.

Wang, C.S., Juan, C.J., Yeh, C.C., Lin, T.Y and Chiang, S.Y. (2017). Prediction model of cervical spine disease established by genetic programming. In: *Proceedings of the 4th multidisciplinary international social networks conference* (p. 38). ACM https://doi.org/10.1145/3092090.3092097

Wang, S. (2019). Development of a predictive model of tuberculosis transmission among household contacts. *Canadian Journal of Infectious Diseases and Medical Microbiology.* https://doi.org/10.1155/2019/5214124

WHO (World Health Organisation). (2018). Global health TB report. https://doi.org/ISBN 978-92-4-156564-6

WHO (World Health Organization). (2018). Global tuberculosis report 2018- Executive summary. Geneva: World Health Organization.

Xu, W. (2005). Survey of clustering algorithms. IEEE. *Transactions on Neural Networks* **16**(3): 645.