

An automated machine learning model for diagnosing COVID-19 infection

Noor Maher, Suhad A. Yousif

Department of Computer Science, Al-Nahrain University, Baghdad, Iraq

Article Info

Article history:

Received Jul 25, 2022

Revised Sep 14, 2022

Accepted Jan 1, 2023

Keywords:

Automated machine learning

COVID-19

Genetic programming

Tree-based pipeline

optimization tool

ABSTRACT

The coronavirus disease 2019 (COVID-19) epidemic still impacts every facet of life and necessitates a fast and accurate diagnosis. The need for an effective, rapid, and precise way to reduce radiologists' workload in diagnosing suspected cases has emerged. This study used the tree-based pipeline optimization tool (TPOT) and many machine learning (ML) algorithms. TPOT is an open-source genetic programming-based AutoML system that optimizes a set of feature preprocessors and ML models to maximize classification accuracy on a supervised classification problem. A series of trials and comparisons with the results of ML and earlier studies discovered that most of the AutoML beat traditional ML in terms of accuracy. A blood test dataset that has 111 variables and 5644 cases were used. In TPOT, 450 pipelines were used, and the best pipeline selected consisted of radial basis function (RBF) Sampler preprocessing and Gradient boosting classifier as the best algorithm with a 99% accuracy rate.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Noor Maher

Department of Computer Science, Al-Nahrain University

Km. Al Jadriyah Bridge, Baghdad, Iraq

Email: noor.maher.cs2020@ced.nahrainuniv.edu.iq

1. INTRODUCTION

On March 11th, 2020, the world health organization (WHO) declared coronavirus disease 2019 (COVID-19), the sickness caused by the COVID-19 virus, as a pandemic. This pandemic has impacted all facets of life. Despite significant medical developments and a comprehensive vaccination campaign in some countries, the situation remains uncontrollable [1]. The biggest issue is that COVID-19, in its later stages, can cause the failure of the lungs, which might result in mortality. People nowadays use polymerase chain reaction (PCR) or computed tomography scan (CT) to determine whether they have COVID-19 or not. However, PCR and CT scans are more expensive and time-consuming than our suggested diagnostic technique, blood tests. Therefore, many attempts are tried to improve the accuracy of blood testing [2]. Previous studies have used machine learning (ML) methods to detect COVID-19 [3]. One of the most popular and essential tasks in ML is classification.

Moreover, choosing the best classification algorithm for a particular dataset is difficult because an algorithm's predicted performance is heavily influenced by the dataset's properties and hyperparameter settings [4]. Therefore, skilled ML practitioners know the importance of designing practical ML algorithms. Furthermore, creating ML pipelines is time-consuming and requires professional field knowledge [5]. Therefore, there will be an increasing need for more accessible, adaptable and scalable data science tools as

the field grows more and more popular. In response to this need, researchers working on AutoML have started developing systems that automate the creation and improvement of ML pipelines. Over the years, various AutoML strategies have been developed to address the above challenges [6]. For example, a tree-based pipeline optimization tool (TPOT) is one of the AutoML tools that automatically creates and optimizes ML pipelines for the problem domain [7]. In short, TPOT uses a variant of genetic programming (GP), a classification and clustering system based on evolutionary learning where encoding the chromosomes into trees is a highly flexible heuristic tool.

Furthermore, GP generates a classifier as a search and optimization algorithm [8]. The authors [9], [10] used a blood test dataset to detect COVID-19. Nevertheless, their studies have some limitations, such as low accuracy or small dataset. This paper proposes a model based on one of the AutoML tools (TPOT) developed to tackle the above limitations, tries to diagnose the COVID-19 virus based on a routine blood tests model, and finally tries to find answers to the following questions:

- a. Is it possible to reach a higher accuracy using AutoML?
- b. Could artificial AutoML replace data scientists?

The structure of this paper is divided as follows: The definitions and background of ML and AutoML are described in Sections 2. Related works are presented in Sections 3, while the methodology is described in detail in Sections 4, the results and discussion are presented in Sections 5. Finally, Sections 6 illustrates the conclusion.

2. BACKGROUND

2.1. Machine learning

"ML is a branch of artificial intelligence that focuses on developing systems that can learn from examples and improve without being explicitly programmed" [11]. ML is beneficial in various situations. A single ML method may frequently simplify code and improve performance for problems requiring long lists of rules or much hand-tuning [12]. These systems are classified according to the amount and type of supervision they receive during their training. The four major categories are supervised, unsupervised, semi-supervised, and reinforcement ML. Methods such as k-nearest neighbors (k-NN) [13], linear regression (LR) [14], logistic regression (LR) [15], support vector machines (SVM) [16], decision tree (DT) [17], random forest (RF) [18], Gradient boosting classifier [19], and neural network (NN) [20] are some of the most popular supervised learning algorithms. Unsupervised learning algorithms are also helpful in clustering methods like k-means [21], [22], hierarchical cluster analysis [23], and principal component analysis (PCA) [24].

2.2. Automated machine learning (AutoML)

AutoML automates the end-to-end process of applying ML to real-world problems. In a typical ML application, data preprocessing, feature extraction, feature selection, and other tasks must be made correctly so that the data fits the target ML task. Practitioners must also experiment with alternative learning models, tune their hyperparameters, and optimize their normal parameters to get good outcomes. Non-experts are generally unable to complete most of these steps; therefore, AutoML is presented as an artificial intelligence-based answer to the ever-increasing problem of ML applications. Automating the end-to-end process of applying ML provides more straightforward solutions, faster construction of those solutions, and models that frequently outperform models created manually. AutoML has the potential to dramatically lower the barrier to entry for non-experts when it comes to using ML techniques [25]. These approaches, which seek the best combination of classification algorithms and hyperparameter settings to optimize prediction performance in an input dataset, are an emergent approach to solving this challenge. There are several AutoML platforms to choose from [26].

2.3. TPOT tool operations

One of the earliest AutoML techniques and open-source software programs created for the data science community was TPOT. In the operations of TPOT, each GP primitive or ML pipeline operator corresponds to an ML tool. For example, scikitlearn is a python ML tool for which TPOT is a wrapper [5]. TPOT has numerous data processing operations, each of which has many operators. These operators are combined as GP primitives to form a GP tree that enables automatic ML. This paper discusses the following four operators [27].

- 1) Preprocessors. The TPOT tool's operator scales features using the sample mean and variance (standard scaler), scales features using sample median and interquartile range (robustscaler) and produces interactive features using a polynomial combination of numerical features (polynomial features).
- 2) Decomposition. Randomized PCA, a variant of PCA that uses randomized singular value decomposition (SVD), is used to decompose the dimensionality reduction.

- 3) Feature selection. Select KBest, select percentile, and variance threshold are feature selection operators in this tool. First, KBest was selected, meaning that the top K features were chosen; when the percentile was selected, the top K percentile of features was chosen. Finally, when the variance threshold was established, the threshold was set, and any feature that did not meet it was rejected. In addition, SelectKBest with chi-square tests and mutual information was used to pick features.
- 4) Models' selection. This tool includes a DT, RF, Gradient boosting classifier, SVM, LR, and a K-NN classifier for supervised learning. The signal data set serves as the tree's leaves in any tree-based pipeline. The tree nodes execute four different forms of processing: preprocessing, decomposition, feature selection, and model selection, before passing the input to the downstream node. In addition, a dataset combination operator can combine several copies of the dataset being processed into a single dataset. This node then processes the dataset; the GP approach automatically creates and optimizes these tree-based pipelines. Algorithm 1 demonstrates the steps of TPOT pipeline operation based on the GP approach Figure 1 shows the example of the genetic programming pipelines.

```

Algorithm 1. TPOT's AutoML pipeline
Data: input x; target y
Input: the number of generation g; population size p; mutation rate m; crossover rate c
Output: the best ML pipeline implemented as a TPOT operation
Start
Population ← initialize random tree (size=p);
a=1
For a= 1 to g do
  Train and score all trees in the population
  For b = 1 to p do
    Train pipeline (population [b], x, y);
    Pop fitness[b]← score pipeline (population[b], x, y);
  End
  Copy the most suitable tree to 10% of the new population
  For t = 1 to p ×0.1 do
    New population [t] ← population [arg max (pop fitness)];
  End
  Fill the remaining 90% with the best tournament top teams.
  For n = 1 to p ×0.9 do
    Run a tournament on random three tree
    Top tem ← run tree way tournament (population)
    Population [(p ×0.1) +n]← top tem
  End
  Perform one point crossover on 5% of the new population
  One point crossover (random choice (new population, 0.05));
  Mutate (random choice (new population, 0.9));
  Population←new population
End
return population [arg max (pop fitness)]
End
    
```

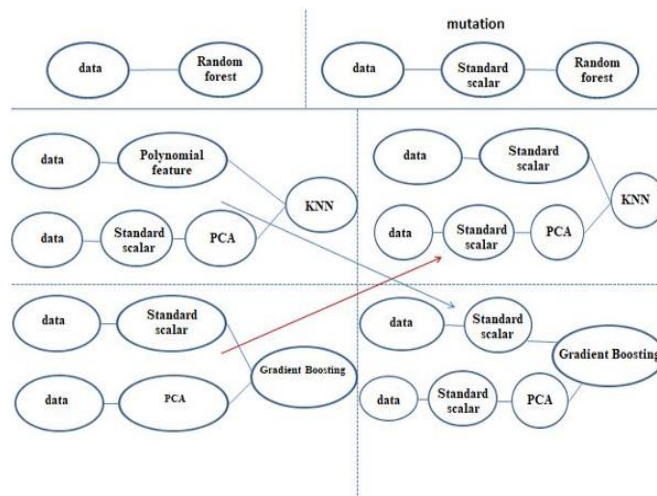


Figure 1. Genetic programming pipeline operations

3. RELATED WORK

Using blood tests to diagnose COVID-19 is uncommon; however, some studies have covered similar ideas using a dataset. In this paper, some of these studies will be mentioned in Slovenia, [28] employed deep neural network (DNN), RF, and extreme gradient boosting (XGBoost). These algorithms were used to construct models predicting COVID-19 diagnoses based on routine blood test results. The researchers analyzed data from 5333 patients hospitalized in the Department of Infectious Diseases at the University Medical Centre Ljubljana. One hundred sixty tested positive for HIV. The best results came from XGBoost, with an area under the ROC curve (AUC) of 97%, a sensitivity of 81%, and a specificity of 97%. In 2020, Almansoor and Hewahi [29] investigated the association between COVID-19 and blood tests using ML techniques. AdaBoost, library for support vector machines (LIBSVM), RF, k-NN, and ensemble learning were employed in this research. A real dataset from a Brazilian hospital was used to test the models. It includes 5644 occurrences and 111 variables. In terms of accuracy, all of the ML models performed well, including LIBSVM (70%), AdaBoost (85%), K-NN (78%), and RF (76%) accuracy percent. In comparison, the ensemble model achieved a 65% accuracy rate. Because all models had a low true positive rate (TP), this study may suggest that having a standard blood test does not help identify COVID-19. The TP rate may also be lowered if the data is unbalanced. A future research direction would be to use the other ML models on more and various real datasets. In 2021, Izdihar *et al.* [30], the AutoML method; study collected 70 chest x-ray (CXR) photos from a hospital in Kuala Lumpur to investigate the sensitivities of detection and evaluate the accuracy of the classifier performance. TPOT has an accuracy result of 0.83, and K-NN was chosen as the best pipeline. In 2019, Javel *et al.* [31] used a (TPOT) and proposed a model for epileptic seizure identification via electroencephalogram (EEG) inputs to TPOT. The dataset was obtained from the University of California at Irvine's ML repository. It features a 23.6-second recording of 500 people's brain activity. In a single grid search, there are 90 pipeline configurations for evaluation, with roughly 450 models fitted and assessed against the training data. The best pipeline has the highest cross-validation score in the run at 95.94%. [32] This study predicted COVID-19 via data analysis (DA) and ML models based on patient clinical data. The authors presented five models to be trained and evaluated according to well-defined evaluation criteria, aiming to select the best model. They established the "SVM" as the best model, then optimized it using "GridSearchCV" optimization. At the end of the optimization, the performance remained the same: an accuracy of 0.99, a recall of 0.93, and a perfect specificity of 1.0 on the first dataset collected from the Hospital Israelita Albert Einstein So Paulo Brasilia dataset. Then the authors applied for the same work with a different dataset collected from Raffaele Hospital Milan Italia. Once more, the SVM presented the best performance: 92.86%, 93.55%, and 90.91% for accuracy, sensitivity, and specificity, respectively. [33] This study developed an ML model using 27 standard laboratory tests and patient demographic features to predict an individual's COVID-19 infection status. Four popular classifiers were used, with laboratory testing results to train a gradient boosting decision tree (GBDT) model from 3,356 SARS-CoV-2 PCR-tested patients evaluated at a metropolitan hospital. The model achieved an AUC of 0.854 (95% CI: 0.829–0.878). [34] By applying the XGBoost algorithm, the authors propose a COVID-19 diagnosis model based on patient symptoms and routine test results. The XGBoost model achieved a sensitivity of 92.5% and a specificity of 97.9% in discriminating COVID-19 patients from influenza patients. Furthermore, the AUC is over 0.9 for all clusters. The high AUC for the cluster may not be accurate because it contains more missing data fields.

4. METHOD

TPOT and many ML algorithms have been used to determine the positive and negative cases of COVID-19. This paper first discusses the COVID-19 dataset and the steps needed to prepare the data for the model. The collected data contains some noise and needs to be cleaned up. Therefore, it cannot be processed directly. Figure 2, Depicts the workflow steps as they are divided into four steps. The four steps are (loading dataset, data preparation, classification phase, and evaluation phase) as described in sections (4.1), (4.2), and (4.3) respectively.

4.1. Description and loading of the dataset

Patients from Albert Einstein Hospital in So Paulo, Brazil, provided the anonymized data for this study [35]. Samples from patients were taken to complete COVID-19 tests and other lab work. The hospital submitted the data to Kaggle, covering the dates of March 28th and April 3rd, 2020. There are 5644 instances and 111 variables, with the COVID-19 dependent variable being one of them (positive or negative). COVID-19 was detected in 558 (~10%) of the 5644 tests, whereas 5086 (~90%) were negative, confirming that it is an unbalanced dataset [36].

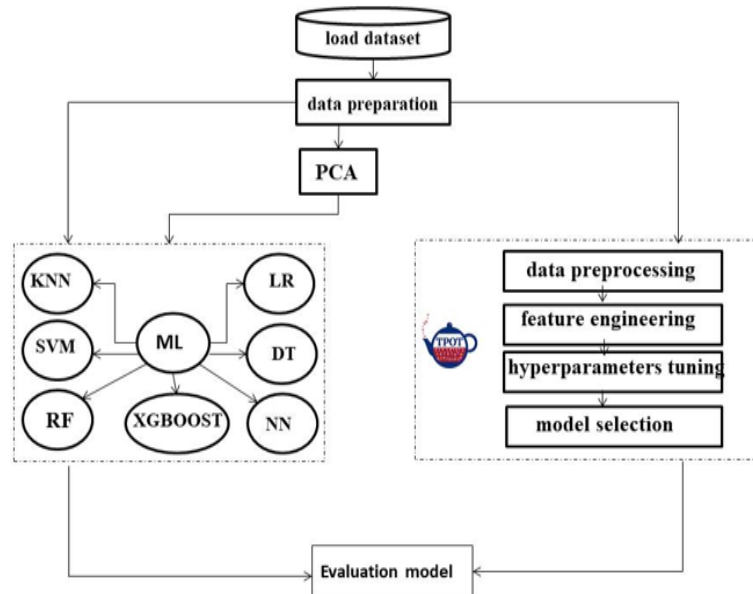


Figure 2. The workflow steps

4.2. Data preparation

This data needs to be cleaned up first because it contains many NULL values and some columns contain categories without numbers. The process of data cleaning includes the following:

- a. Dropping the target ['SARS'] from the dataset.
- b. Filling all the NULL values using the mean and deleting the empty columns.
- c. Label encoding involves replacing categorical values with numerical ones.
- d. Balancing the dataset using the random over sampler technique, altering the size to 10172 rows by 106 columns.

4.3. Classification model

After preparing the dataset, two classification experiments are proposed; the first experiment applies the TPOT library for auto-classification of the data without any feature engineering because it does this process automatically, and the second experiment includes entering the data into ML algorithms with a default parameter without any feature engineering or dimensionality reduction. Then we will apply the dimensionality reduction using PCA before we enter it into the ML algorithms. More detail about these experiments will be discussed in the classification model sections.

4.3.1. AutoMI (TPOT) classification model

This part discusses the many TPOT experiments which were performed. Three different configurations were used, and the dataset was divided into 75% for training the model and 25% for testing the model and score pipelines based on classification accuracy with 5-fold cross-validation. To study the effects across all experiments, the TPOT gives different results in each experiment depending on the various settings of hyperparameters. TPOT provides different accuracy results, feature selectors, feature preprocessing, and algorithms. The implemented experiments are discussed following:

- a. Experiment A. A TPOT with 450 pipeline configurations, including eight generations and 50 population sizes, was employed.
- b. Experiment B. A TPOT with 180 pipeline configurations, including eight generations and 20 population sizes, was used.
- c. Experiment C. A TPOT with 80 pipeline configurations, including three generations and 20 population sizes, was applied.

This work has already discussed the details of TPOT at the command line. In which TPOT accepts several arguments. Equation (1) is used to calculate the number of pipelines used inside the TPOT. We used several of these command lines where the population size is the number of individuals retained in the GP population that is different for each generation; generation Iterations are required to complete the pipeline optimization procedure, and offspring size is the number of offspring to be produced in each generation of GP.

$$\text{TPOT pipelines} = \text{POPULATION_SIZE} + \text{GENERATIONS} \times \text{OFFSPRING_SIZE} \quad (1)$$

4.3.2. Machine learning algorithms

Another experiment used only the multiple ML algorithms defined in TPOT, such as LR, DT, RF, SVM, K-NN, XGBoost, Gradient boosting classifier, and NN. These algorithms used in two experiments:

- a. The ML algorithms with default parameters.
- b. The ML techniques used along with the PCA for dimensionality reduction. Before performing PCA, outliers must be removed from the dataset, features must be scaled to comparable dynamic ranges (normalization), and missing data must be addressed.

4.4. Evaluation metric

Accuracy is the recommended evaluation metric, which is based on the values: true negative (TN) and true positive (TP): The model correctly predicts the class (negative or positive). False negative (FN) and false positive (FP): The model wrongly predicts the class (switch negative and positive classes). As described in (2).

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{All samples}} \quad (2)$$

5. RESULTS AND DISCUSSION

This study uses AutoML tools (TPOT) to detect COVID-19 and compares the results of the TPOT with several ML algorithms. After making some adjustments as mentioned in section (4). When evaluating the models we note the TPOT performed exceptionally well in classification and even outperformed the ML algorithms.

5.1. Comparing the outputs of TPOT experiments

Based on the trial-and-error method for determining the number of generations and population size, several TPOT experiments were conducted on the dataset. First, the classification accuracy of the pipelines was evaluated using the confusion matrix and 5-fold cross-validation. Figure 3 shows the confusion matrix of the TPOT experiments. The most accurate experiment was experiment A, with results where accuracy equals 99%. TPOT used a predictive model for this experiment that included a Gradient boosting classifier and radial basis function (RBF) sampler preprocessing.

Figure 3(a) illustrates the confusion matrix of experiment A, where 1,248 true positives have been accurately predicted, and the false positive number is zero. Hence, there is no positive case in which the model predicts a mistake, and of the 1,288 true negative cases, only 7 cases were false negatives. In experiment B, TPOT achieved 98% accuracy. The best algorithm for the predictive model consists of RBF sampler preprocessing and an XGBoost classifier. Figure 3(b), Shows the confusion matrix of this experiment where 1,248 true positive COVID-19 cases have been accurately predicted, and the false positive number is zero. These results are identical to previous findings, with 1,254 true negative cases and 41 false negatives. Finally, experiment C achieved the least accuracy at 94%, with the extra trees classifier as the estimator and XGBoost as the classifier model. Figure 3(c) shows the confusion matrix that contained 1,241 true positive COVID-19 cases that have been accurately predicted out of 1,248 positive cases and 1,147 true negative cases out of 1,295 negative cases.

5.2. Comparing the outputs of ML algorithms

This paper uses default parameters and the most known traditional ML algorithms defined in the TPOT library. These algorithms produced several different results than TPOT results. To check the effect of applying AutoML classification and the standard ML algorithms. First, we applied the same algorithm that gives the highest accuracy (Gradient boost classifier) with its hyperparameters and preprocessing steps produced from TPOT with the same ML algorithm (Gradient boost classifier). Similar results were obtained in both cases as a result equals 99%, while the result equals 67% when using (the Gradient boost classifier) with the default parameters defined in algorithm hyperparameters in Python.

Furthermore, the result shows that The RF had the highest accuracy of the ML algorithms tested with default parameters with a score of 95%, while the NN achieved the lowest accuracy with a score of 49% using the standard NN algorithm. Figure 4. Shows the accuracy of the algorithms used. From these results, we can see the effect of setting hyperparameters and choosing the best preprocessing and feature extraction and selection using the concept of GP in the TPOT library rather than using the default parameters or preprocessing and feature extraction and selection.

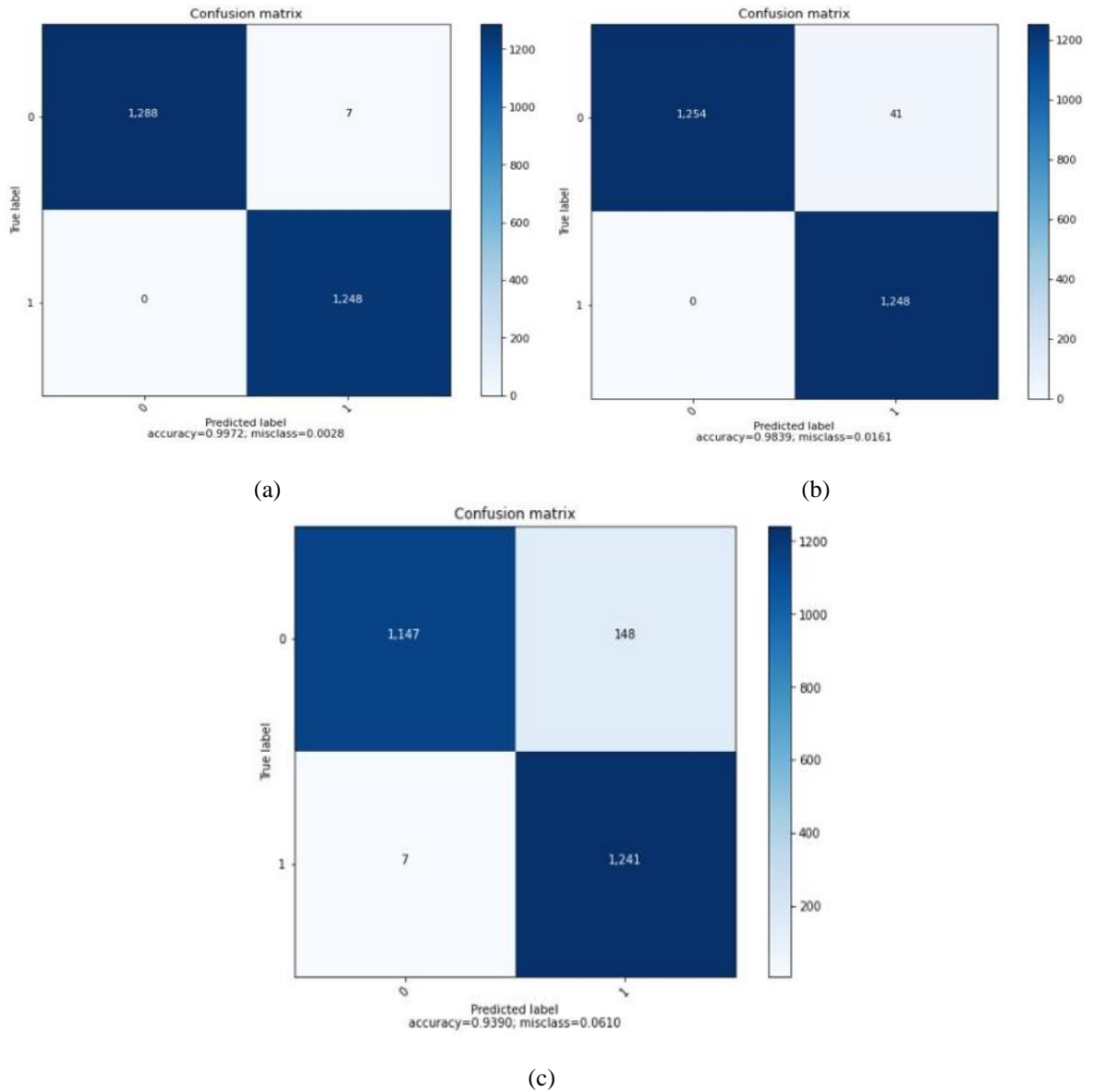


Figure 3. Shows the confusion matrix of many TPOT experiments in figures (a) Show the result of TPOT-450 pipeline, (b) Show the result of TPOT-180 pipeline, and (c) Show the result of TPOT-80 pipeline.

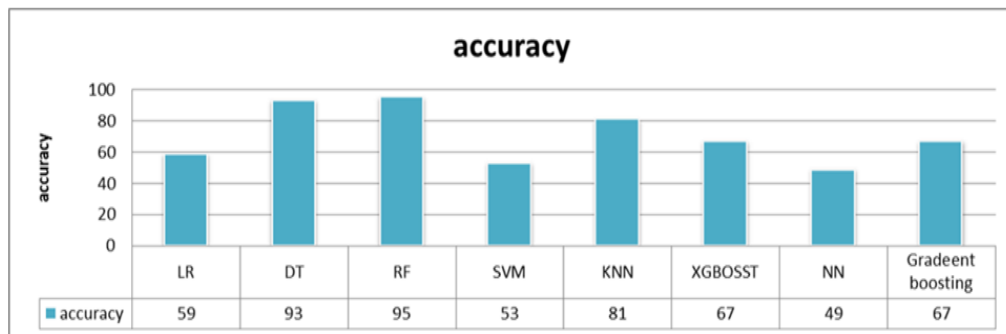


Figure 4. Accuracy of multiple ML with default parameters

5.3. Comparing the outputs of ML algorithms with PCA

This paper applied dimensionality reduction. Sometimes the number of features should be maintained to a minimum to reduce computational complexity. This work uses PCA with the same ML algorithms applied above. The DT and RF achieved the highest accuracy of the algorithms tested, with a score of 94%. In contrast, the LR achieved the lowest accuracy at 54%. All the features used are essential since we got less accuracy when reducing the dimensionality of the features, but it is an acceptable accuracy rate with a computational time less than applying ML algorithms alone. After that, this paper compares these results with those in previous works that used the same dataset [29] shown in Table 1.

Interestingly, AutoML outperforms better than traditional ML. Figure 5. Shows the accuracy of these experiments. While in Figure 6, the accuracy of ML and TPOT was shown. Finally, Figure 7 shows the accuracy of ML with PCA and TPOT.

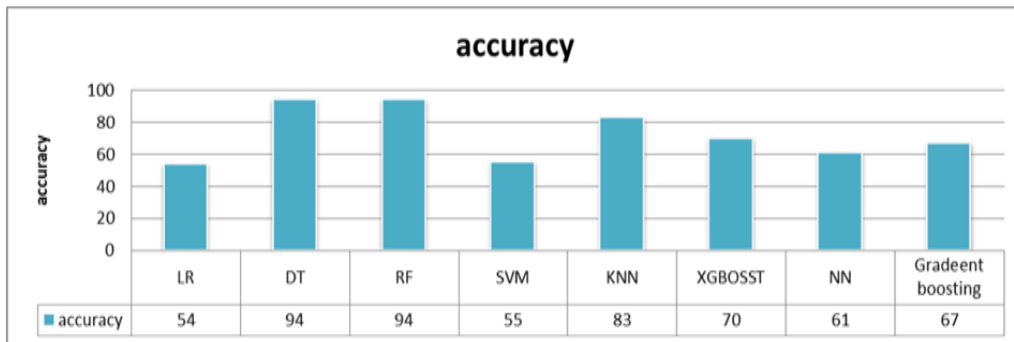


Figure 5. Accuracy of ML learning with PCA and default parameters

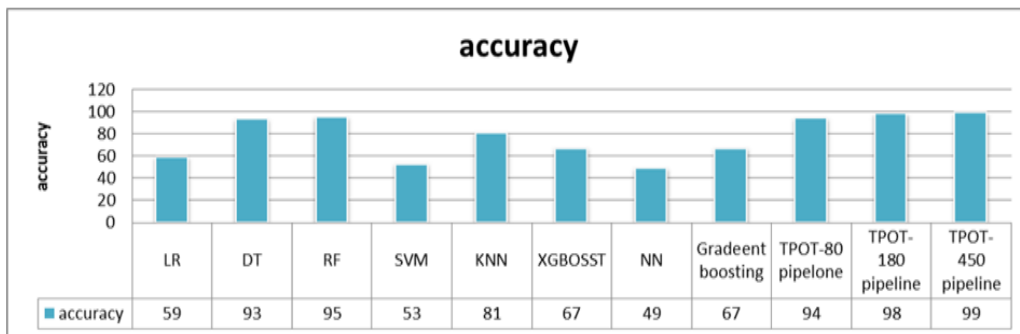


Figure 6. Shows the accuracy of ML and TPOT

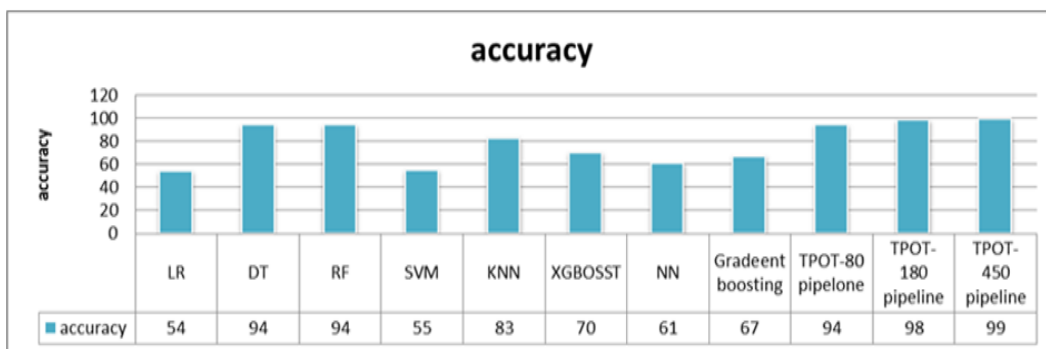


Figure 7. Shows the accuracy of ML with PCA and TPOT

Table 1. Comparison with state-of-the-art methods

Study	Method	Accuracy
[29]	AdaBoost	85%
	LIBSVM	70%
	RF	76%
	K-NN	78%
	ensemble learning	65%
Propose method	TPOT-450 pipeline	99%

6. CONCLUSION

This study developed a model for COVID-19 diagnosis by using (TPOT). The blood tests dataset is used and fed to TPOT after balancing it. Additionally, many experiments were run using TPOT with different parameters and compared the results were too many ML algorithms. Also, apply ML with PCA. It was discovered that feature preprocessing and algorithms with the parameters obtained via TPOT resulted in high-performance accuracy for COVID-19 detection. Depending on the parameters we choose, TPOT produces different results. The more pipelines are employed, the better the results and accuracy. Furthermore, we answered most of the questions posed in this paper and concluded that AutoML is superior to traditional ML. It shortens most programming steps, making it a valuable tool for beginners and those without sufficient programming experience. However, AutoML will not replace data scientists anytime soon. It is a tool to help data scientists and is a great way to clarify this complex field for non-experts so they can learn from the ML experience. Also, in some situations, there is no need for an expert data scientist to handle the hyperparameters tuning and choosing the best preprocessing steps and feature engineering to achieve higher accuracy.




REFERENCES

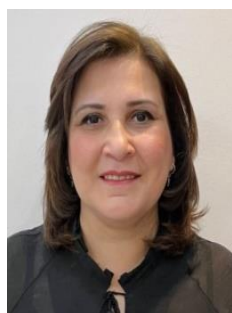
- [1] M. A. Alves *et al.*, "Explaining machine learning based diagnosis of COVID-19 from routine blood tests with decision trees and criteria graphs," *Computers in Biology and Medicine*, vol. 132, p. 104335, May 2021, doi: 10.1016/j.combiomed.2021.104335.
- [2] N. Hany, N. Atef, N. Mostafa, S. Mohamed, M. ElSahhar, and A. AbdelRaouf, "Detection COVID-19 using machine learning from blood tests," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, May 2021, pp. 229–234. doi: 10.1109/MIUCC52538.2021.9447639.
- [3] N. Alballa and I. Al-Turaiki, "Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: A review," *Informatics in Medicine Unlocked*, vol. 24, p. 100564, 2021, doi: 10.1016/j.imu.2021.100564.
- [4] J. C. Xavier-Junior, A. A. Freitas, A. Feitosa-Neto, and T. B. Ludermir, "A novel evolutionary algorithm for automated machine learning focusing on classifier ensembles," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct. 2018, pp. 462–467. doi: 10.1109/BRACIS.2018.00086.
- [5] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., *Automated machine learning*. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-05318-5.
- [6] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, Nov. 2015, doi: 10.1007/s13218-015-0381-0.
- [7] G. Squillero and P. Burelli, Eds., *Applications of evolutionary computation*, vol. 9598. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-31153-1.
- [8] M. Khamar and M. Eftekhari, "Generating kernel matrix for rotation forest through genetic programming," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Feb. 2018, pp. 98–101. doi: 10.1109/CFIS.2018.8336642.
- [9] V. Bayat *et al.*, "A severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) prediction model from standard laboratory tests," *Clinical Infectious Diseases*, vol. 73, no. 9, pp. e2901–e2907, Nov. 2021, doi: 10.1093/cid/ciaa1175.
- [10] L. T *et al.*, "Use of machine learning to rapidly predict positivity to severe acute respiratory syndrome coronavirus 2(SARS-COV-2) using basic clinical data," 2020, doi: 10.21203/rs.3.rs-38576/v1.
- [11] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, Sep. 2020, doi: 10.1007/s11831-019-09344-w.
- [12] A. Géron, "O'Reilly media," in *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, 2019.
- [13] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," *Building and Environment*, vol. 202, p. 108026, Sep. 2021, doi: 10.1016/j.buildenv.2021.108026.
- [14] C. Yu and W. Yao, "Robust linear regression: A review and comparison," *Communications in Statistics - Simulation and Computation*, vol. 46, no. 8, pp. 6261–6282, Sep. 2017, doi: 10.1080/03610918.2016.1202271.
- [15] E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, Sep. 2019, doi: 10.1007/s12652-018-1093-8.
- [16] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, Elsevier, 2020, pp. 101–121. doi: 10.1016/B978-0-12-815739-8.00006-7.
- [17] L. Li and J. Wang, "Research on feature importance evaluation of wireless signal recognition based on decision tree algorithm in cognitive computing," *Cognitive Systems Research*, vol. 52, pp. 882–890, Dec. 2018, doi: 10.1016/j.cogsys.2018.09.007.
- [18] R. Jehad and S. A. Yousif, "Fake news classification using random forest and decision tree (J48)," *Al-Nahrain Journal of Science*, vol. 23, no. 4, pp. 49–55, Dec. 2020, doi: 10.22401/ANJS.23.4.09.
- [19] M. S. Islam Khan, N. Islam, J. Uddin, S. Islam, and M. K. Nasir, "Water quality prediction and classification based on principal




- component regression and gradient boosting classifier approach,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 4773–4781, Sep. 2022, doi: 10.1016/j.jksuci.2021.06.003.
- [20] R. Jehad and S. A. Yousif, “Classification of fake news using multi-layer perceptron,” in *AIP Conference Proceedings*, 2021, p. 070004. doi: 10.1063/5.0042264.
- [21] H. H. Maala and S. A. Yousif, “Cluster trace analysis for performance enhancement in cloud computing environments,” *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 7, pp. 2076–2091, 2019.
- [22] N. S. Sagheer and S. A. Yousif, “Canopy with k-means clustering algorithm for big data analytics,” in *AIP Conference Proceedings*, 2021, p. 070006. doi: 10.1063/5.0042398.
- [23] N. Subba Rao and M. Chaudhary, “Hydrogeochemical processes regulating the spatial distribution of groundwater contamination, using pollution index of groundwater (PIG) and hierarchical cluster analysis (HCA): A case study,” *Groundwater for Sustainable Development*, vol. 9, p. 100238, Oct. 2019, doi: 10.1016/j.gsd.2019.100238.
- [24] G. Dougherty, *Pattern recognition and classification*. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-5323-9.
- [25] Z.-C. Lin, “How can machine learning and optimization help each other better?,” *Journal of the Operations Research Society of China*, vol. 8, no. 2, pp. 341–351, Jun. 2020, doi: 10.1007/s40305-019-00285-6.
- [26] A. Mustafa and M. Rahimi Azghadi, “Automated machine learning for healthcare and clinical notes analysis,” *Computers*, vol. 10, no. 2, p. 24, Feb. 2021, doi: 10.3390/computers10020024.
- [27] W. Zhang, P. Ge, W. Jin, and J. Guo, “Radar signal recognition based on TPOT and LIME,” in *2018 37th Chinese Control Conference (CCC)*, Jul. 2018, pp. 4158–4163. doi: 10.23919/ChiCC.2018.8483165.
- [28] M. Kukar *et al.*, “COVID-19 diagnosis by routine blood tests using machine learning,” *Scientific Reports*, vol. 11, no. 1, p. 10738, May 2021, doi: 10.1038/s41598-021-90265-9.
- [29] M. Almansoor and N. M. Hewahi, “Exploring the relation between blood tests and COVID-19 using machine learning,” in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, Oct. 2020, pp. 1–6. doi: 10.1109/ICDABI51230.2020.9325673.
- [30] K. Izdihar *et al.*, “Detection of novel coronavirus from chest x-ray radiograph images via automated machine learning and CAD4COVID,” in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, Jul. 2021, pp. 1–4. doi: 10.1109/ICOTEN52080.2021.9493542.
- [31] I. M. Javel, R. C. Salvador, E. Dadios, R. R. P. Vicerra, and A. T. Teologo, “Epileptic seizure detection via EEG using tree-based pipeline optimization tool,” in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Nov. 2019, pp. 1–5. doi: 10.1109/HNICEM48295.2019.9073465.
- [32] A. Tchagna Kouanou *et al.*, “An overview of supervised machine learning methods and data analysis for COVID-19 detection,” *Journal of Healthcare Engineering*, vol. 2021, pp. 1–18, Nov. 2021, doi: 10.1155/2021/4733167.
- [33] H. S. Yang *et al.*, “Routine laboratory blood tests predict SARS-CoV-2 infection using machine learning,” *Clinical Chemistry*, vol. 66, no. 11, pp. 1396–1404, Nov. 2020, doi: 10.1093/clinchem/hvaa200.
- [34] W. T. Li *et al.*, “Using machine learning of clinical data to diagnose COVID-19: a systematic review and meta-analysis,” *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, p. 247, Dec. 2020, doi: 10.1186/s12911-020-01266-z.
- [35] Kaggle. Diagnosis of covid-19 and its clinical spectrum | kaggle. 2020. <https://www.kaggle.com/einsteindata4u/covid19>. Accessed on 07/18/2020.
- [36] J. Wu, J. Shen, M. Xu, and M. Shao, “A novel combined dynamic ensemble selection model for imbalanced data to detect COVID-19 from complete blood count,” *Computer Methods and Programs in Biomedicine*, vol. 211, p. 106444, Nov. 2021, doi: 10.1016/j.cmpb.2021.106444.

BIOGRAPHIES OF AUTHORS



Noor Maher    is a Master's Student at Al-Nahrain University/College of Science/Computer science department. She received his BSc from Al-Nahrain University in 2012. Her special area is in Machine learning she can be contacted at email: noor.maher.cs2020@ced.nahrainuniv.edu.iq



Suhad A. Yousif    She is an assistant professor at Al-Nahrain University/College of science. She is head of the computer science department. She received her BSc from Al-Nahrain University in 1994, the M. Sc in Computer Science department/Baghdad university in 2005, and Ph.D. degrees from Mathematics and Computer Science Department in Beirut Arab University/Lebanon in 2015. Dr. Suhad supervises M.Sc. theses concerning cloud computing, big data analysis, text classification (natural language processing), and classification of Ensemble machine learning, Automated machine learning, and forecasting prediction. She also leads and teaches different subjects at both B.Sc. and M.Sc. Levels in computer science. In addition, she is on a scientific committee at some conferences and a reviewer in several conferences and Journals. Her particular area of research is big data Data Science, Machine learning, and deep learning. She can be contacted at email: suhad.a.yousif@nahrainuniv.edu.iq