2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) | 978-1-7281-9675-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDABI51230.2020.9325673

EXPLORING THE RELATION BETWEEN BLOOD TESTS AND COVID-19 USING MACHINE LEARNING

Mohamed Almansoor
Big Data Science & Analytics Program
University of Bahrain
mknsr@yahoo.com

Nabil M. Hewahi
Department of Computer Science
University of Bahrain
nhewahi@uob.edu.bh

***Abstract* – COVID-19 is a global pandemic that hit the world in 2019-2020 and caused massive losses. Every day, hundreds of thousands of tests are being done on possible infected cases. It usually takes several hours to get the results of virus test in advanced countries, whereas in other countries might take days. The aim of this study is to investigate whether normal blood medical tests help in detecting covid-19 using various machine learning approaches. If true, this would give an indication to people who should undergo the virus test. In this paper we independently use machine learning algorithms including support vector machines, adaptive boosting, random forest and *k*-nearest neighbors. These algorithms are then merged to form ensemble learning which leads to the classification. The results show that the ensemble learning is having the highest true positive rate of 30%. The obtained results show that normal blood tests do not help much in giving right indications about detecting COVID-19.**

Keywords – COVID-19, Machine Learning, Blood Tests, KNN, SVM, AdaBoost, Random Forest, Ensemble Learning

I. INTRODUCTION

The World Health Organization (WHO) listed COVID-19, caused by SARS-CoV-2, as a pandemic on March 11, 2020 [1]. Generally, the global healthcare sector is a field that demands long and continuous hours of working especially in the awake of a pandemic. During this pandemic hard time, data scientists are looking for ways to discover patterns in the medical data through data mining and machine learning algorithms, in order to support the analysis and prediction of classification, which helps with the diagnosis of diseases.

With the ongoing COVID-19 pandemic, it is a must to research ways to help improve efficiency when it comes to the diagnosis of suspected cases, in order to have a quick and accurate prediction especially when there is a large number of people who need to be examined.

Currently, it takes many hours, and sometimes a couple of days, to have a definite result which is also not very accurate. There have been cases where the results of COVID-19 tests were false positive and false negative.

The aim of this work is to explore whether there is a relation between normal blood tests and COVID-19 using classification techniques of machine learning.

The dataset, which will be discussed in detail in the next section was obtained recently from a hospital in Brazil with more than 5000 blood tests with their COVID-19 results [2]. The dataset includes different blood tests, but ultimately the goal is to draw a relation between the important independent variables and the COVID-19 binary classification (positive or negative). The models will be discussed in the upcoming sections with comparisons.

II. LITERATURE REVIEW

Machine Learning has been applied in a lot of research, especially considering the ongoing COVID-19 pandemic, to help detect patterns and insights leading or related to the infection. This section will address some of published papers relevant to the subject.

A recent research by Khanday et al. [3] applied machine learning to detect COVID-19. The paper addressed both classic and ensemble machine learning algorithms, and AI tools. The paper aimed to discover cases of COVID-19 based on clinical text data.

The project used Term Frequency/Inverse Document Frequency, Bag of Words, and report length for feature engineering. It was found that Logistic Regression and Naïve Bayes gained better results with 96.2% accuracy.

Authors applied machine learning to forecast COVID-19, presenting a model that relied on regression, Multilayer perceptron and Vector autoregression to predict the number of cases in India [4]. The paper was able to create a prediction model to measure the spread of the virus.

A study applied machine learning to detect COVID-19 in a fast and accurate manner through deep learning methods [5]. This research relied on X-ray and CT scan images based on data obtained from Iran. The researched obtained

84.67% accuracy from X-ray images and 98.78% accuracy in CT.

A study conducted in Hungary, researchers used hybrid machine learning to predict the number of infections and mortality [6]. They used adaptive network-based fuzzy inference system and multi-layered perceptron-imperialist competitive algorithm. The model accurately predicted the decline in by the end of May 2020 in Hungary, although the September 2020 outbreak was not part of the analysis.

III. Exploratory Data Analysis

A. Dataset Description

The dataset used in our research contains anonymized data for patients that had come to the Albert Einstein Hospital in São Paulo, Brazil. The patients had samples collected to perform the COVID-19 tests as well as other laboratory measures.

All data were anonymized with codes using best practices and recommendations. The clinical data were standardized with a mean of zero to create normal distribution.

The dataset was uploaded to Kaggle by the hospital itself and covers the duration between the 28th of March and the 3rd of April 2020. It has 5644 instances and 111 variables (Table 1), including the dependent variable of the COVID-19 result (positive or negative).

Table I: Dataset Overview

Feature	Result
Number of observations	5644
Number of variables	111

Out of 5644 tests, 558 (about 10%) were COVID-19 positive, while 5086 (about 90%) were negative. The problem with this dataset is that it suffers from unbalancing where the number of positives cases is much lower than negative cases. Anyhow, this dataset represents the reality (the number of infected people is much less than the uninfected people).

B. Data Distribution & Correlation Analysis

Fig. 1 as example shows the distribution of four numerical attributes of the dataset; namely Monocytes, Patient's Age, Red Blood Cells, and Serum Glucose, where the x-axis is the value while the y-axis is the frequency.

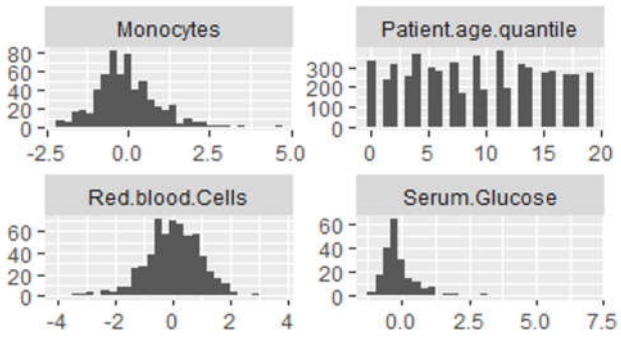

Fig. 1: Histogram Plots of Numerical Attributes

The dataset includes many other attributes including: Hematocrit, Hemoglobin, Leukocytes, Lymphocytes, Mean Platelet Volume, Creatinine, Calcium, Magnesium, Potassium, Sodium, Urea, Vitamin B12, Phosphor, among others.

Fig. 2 shows some of the attributes in boxplot against the COVID-19 outcome indicating how strong the relation of that attribute with the outcome. When cases are clearly separate between positive and negative outcomes, it can be estimated that this attribute strongly affects the outcome.

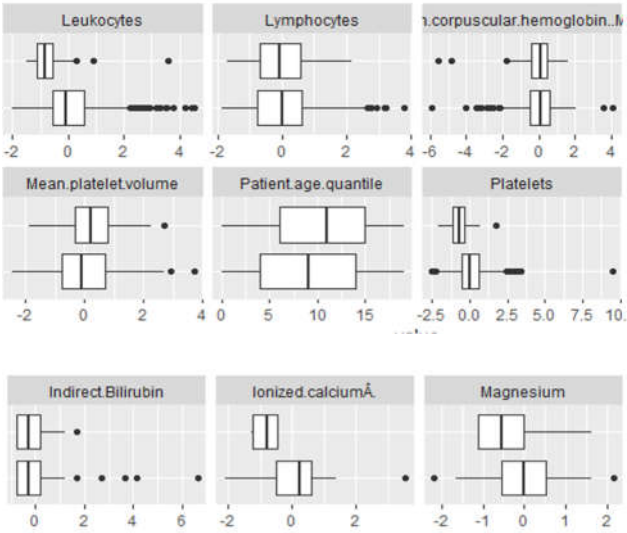

Fig. 2: Boxplots of Some of the Attributes against COVID-19 Result

From Fig. 2, a clear relation between some attributes and the COVID-19 outcome can be noticed, namely for the following attributes: Leukocytes, Platelets, Ionized Calcium, as well as, to a less extent, Magnesium.

To confirm this, a correlation list has been calculated for the most corelated attributes, including the non-numerical ones, to the COVID-19 dependent variable, as in Table 2 below.

Table II: Correlation Table (Top 10 Factors) against COVID-19

ATTRIBUTE	CORRELATION VALUE
Patient admitted to regular ward (1=yes, 0=no)	0.156813
Patient age quantile	0.085209
Monocytes	0.085082
pH (arterial blood gas analysis)	0.084699
Mean platelet volume	0.046766
Red blood Cells	0.046316
Hemoglobin	0.042441
Hematocrit	0.039716
Hb saturation (arterial blood gases)	0.037153
Urine - Granular cylinders	0.036816

In addition to what can be noticed through the histograms of the numerical attributes (Fig. 2), it shows that patient age is also very relevant to the COVID-19 test outcome, as older people are less immune to it.

IV. Data Preprocessing

In this work, Python was used along with several data science libraries. In addition, R has been also added to benefit from the existing R libraries, through the IPython magic extension of R [7].

The code and processing have been hosted on Google Colab which is a cloud-based platform for running Python programs on Google's servers, which offer GPUs and TPUs (Tensor Processing Units) for faster processing [8].

Through the Pandas library, the dataset has been loaded and treated for null inconsistencies. Categorical columns have been converted from numbers to actual categories, which helps in the model fitting later [9]. The Scikit-Learn (sklearn) [10] was used as the main library for the model creation and prediction work.

Due to the imbalance of the dataset of which 90% entries are related to negative results, that portion of the dataset has been under-sampled by 33% resulting in a more balanced dataset. For that, the One-Sided Selection (OSS) technique has been applied. OSS combines Tomek Links and Condensed Nearest Neighbor [11].

The dataset has been split into 70% training and 30% test. Column 2 (i.e. the third column) is the one that has the COVID-19 result category, so it has been dropped from the x dataset (inputs). Furthermore, feature selection was applied here to make use of the top features of the dataset according to the correlation matrix and the chi-squared test result. Not all factors were used in the final models; only the top 15 were included to emphasize their correlation.

V. Learning Methods & Results

Various machine learning methods have been applied and tested with different parameters to achieve the best classification outcomes. This has been done mainly with the help of Scikit-Learn on Python as well as other R libraries that have been used through the IPython magic extension. In this section, the different applied methods and results will be discussed.

A. Support Vector Machine

LIBSVM, stands for a Library for Support Vector Machines, is a very popular machine learning library that uses Sequential Minimal Optimization (SMO) for Support Vector Machines with kernels. It was developed by the National Taiwan University [12].

After testing many parameters, the following have been selected for the final fit: Regularization = 1, Kernel = Sigmoid Function, Kernel coefficient = 1, Class Weight = giving twice the weight for the positive COVID-9 class.

The achieved accuracy varies in every run, but it was 69.79% on average of 10 runs. Fig. 3 presents the confusion matrix that shows 42 true positive COVID-19 cases have been accurately predicted out of 168 positive cases, and 776 true negative cases out of 1004 negative cases. However, the percentage of true positive out of all the positive cases is only 25% and the percentage of true negative out of all the negative cases is 77.29%. This model could work much better in cases of true negative cases, but it is really a big problem because this means 75% of infected people will not be detected.

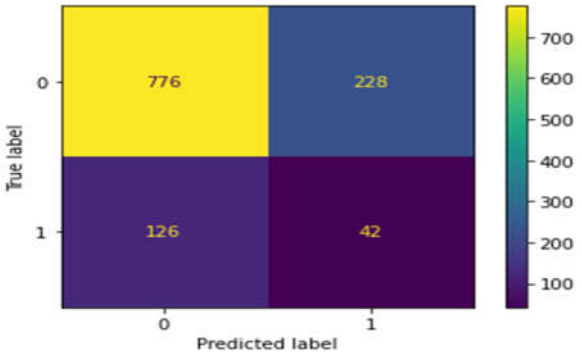

Fig. 3: SVM Classification Model Fit & Confusion Matrix Using sklearn

B. Adaptive Boosting

Adaptive Boosting (AdaBoost) is an ensemble machine learning algorithm that creates a very good classifier from several weak ones [13]. After that, a second model is created that attempts to correct errors from the first one. More models would be created sequentially until a maximum number is reached. Fig. 4 shows an example of how AdaBoost works.

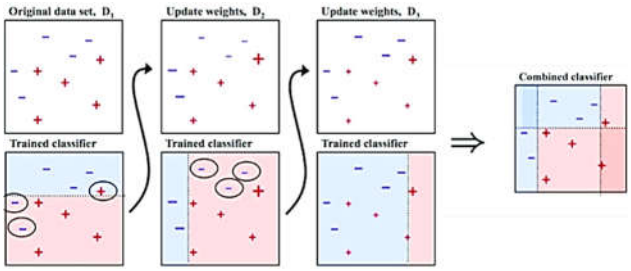

Fig. 4: The AdaBoost Algorithm

AdaBoost is one of the best algorithms that were built upon the concept of boosting and it was worth applying to the dataset that is being studied in this work. Different parameters were also applied, although the default ones showed the best results, which are: Max estimator number at which boosting is terminated = 50, Learning rate = 1, Algorithm = Real Boosting (SAMME.R)

The achieved accuracy rate was 85%, more than what has been achieved with SVM. However, the model failed to predict the positive cases for the most part. Fig. 5 shows the confusion matrix of the applied algorithm. The problem here is worse than SVM because the true positive rate is 0.0059 (0.59%).

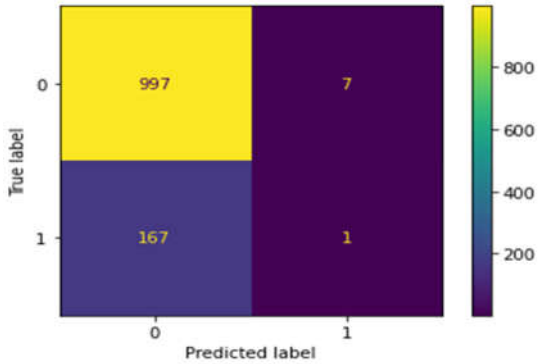

Fig. 5: The AdaBoost Model Code & Confusion Matrix

C. Random Forest

Random Forest (Fig. 6) is another ensemble learning algorithm that is based on the decision tree algorithm [14]. But instead of applying a single instance, elements from the dataset are selected randomly to generate a new dataset on which the decision tree algorithm will be conducted, and this is repeated for the n number of instances. In the end, the majority voting will decide which class elements shall belong to which class. However, unlike AdaBoost, Random Forest is not a boosting technique, but a bagging one. That means each instance does not benefit from the learning outcomes of the other ones but works on its own.

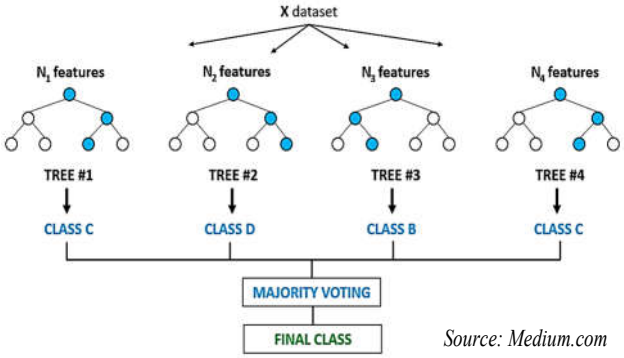

Fig. 6: The Random Forest Algorithm

Using sklearn, the model has been fitted as previously done with SVM and AdaBoost. Most default parameters gave very good results. It was noticed that adjusting the maximum number of features to consider when looking for the best split to half, the accuracy increased by about 2%.

The applied parameter values are: Number of trees = 100, Split quality criterion = Gini impurity, Min split sample number = 2, Min sample number at leaf node = 1, Max features = 1, Using bootstrap = true

The highest achieved accuracy rate was 78%, only slightly less than what has been achieved with AdaBoost. However, the prediction of the positive COVID-19 cases is significantly higher here. Fig. 7 shows the confusion matrix of this model. The true positive rate is 0.13 (13.17%) and true negative rate is 0.887 (88.75%). The problem is still as in the previous models is the true positive rate.

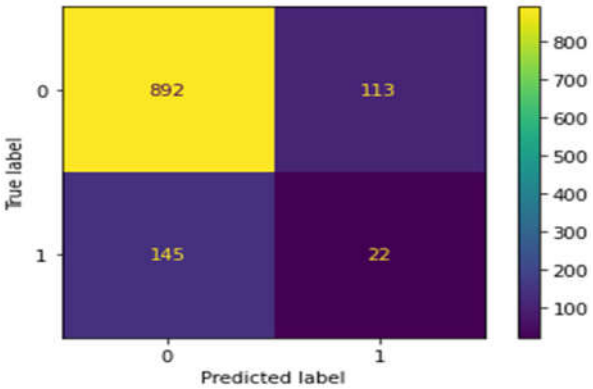

Fig. 7: Confusion Matrix of the Random Forest Model

D. K-Nearest Neighbors

k-Nearest Neighbors (*k*-NN) is a very simple supervised machine learning algorithm that can perform classification tasks. In *k*-NN classification, an observation would be classified by a vote of its neighbors [15]. Therefore, said observation would belong to the class most common among its *k* nearest neighbors.

The default parameters worked very well. However, one parameter was manually changed which lead the accuracy to jump about 1%. The parameter is called *p,* which is the power parameter in the Minkowski space, changed to 3.

The parameters used for fitting the *k*-NN model are: Number of neighbors = 1, Weight prediction = weighted equally, Leaf size = 30, Minkowski power = 3

With the above parameters, a prediction accuracy rate of 76% was achieved. It can be noticed from the confusion matrix (Fig. 8) that the prediction of the positive cases is higher than AdaBoost and Random Forest, but less than SVM. In this experiment, the true positive rate is 0.18 (18.56%) whereas the true negative rate is 0.85 (85.17%).

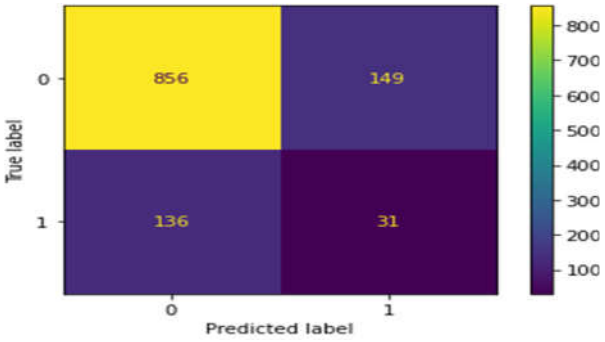

Fig. 8: k-NN Model Result Using sklearn

E. The Ensemble Machine Learning

Because different algorithms showed different results, especially related to the positive cases of COVID-19 which is the main purpose of this finding, an ensemble machine learning could be created to take all four models and cast voting to decide to which class the data belongs to [16]. This will reduce the reliance on a specific algorithm to help achieve more accurate results. Fig. 9 shows the basic principle of ensembles.

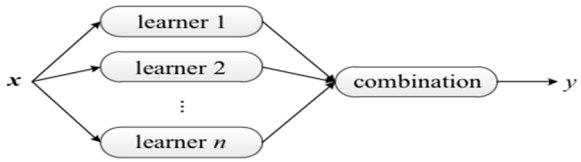

Fig. 9: Ensemble Learning

The four models created previously have been fed into the ensemble, while giving more weight to the SVM method due to achieving better ensemble results. 65% was the accuracy of the ensemble based on the voting of all four methods and the accuracy of the prediction of positive cases was the best of all. Fig. 10 shows confusion matrix. The true positive rate is 0.29 (29.94%) which is the highest in all the models, but the true negative rate decreased and became 0.70 (70.94%). We consider this result as the best result comparing it with other machine learning models despite that the overall accuracy is not the highest, but because its true positive is the highest.

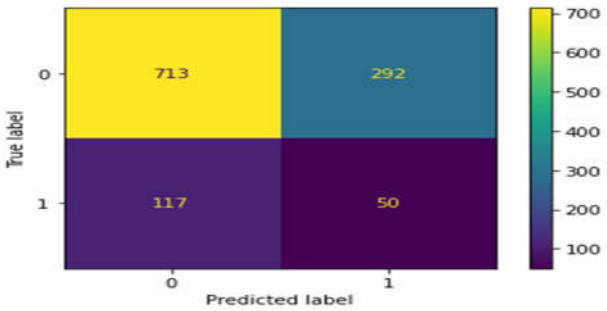

Fig. 10: The Ensemble Model's Confusion Matrix

V. Results & Discussions

As accuracy considered, all the learning models have achieved good results, LIBSVM (70%), AdaBoost (85%), *k*-Nearest Neighbors, (78%) and Random Forest (76%). Meanwhile, The Ensemble Model that combines all and classifies through weighted voting achieved 65%. The accuracy itself does not reflect the real performance of the model, for example, the Ensemble model achieved the best prediction for the positive cases of COVID-19 which is a major issue in this classification. However, its F measure is still the lowest. The worst is AdaBoost where its true positive rate is 0.0059 despite that the accuracy is the highest. Table 3 and Fig. 11 show a summary of the accuracy for all tested classification methods.

Table III: Classification Model Accuracy Comparison

CLASSIFIER	ACCURACY	F-MEASURE
SVM	0.69795	0.81427
AdaBoost	0.85154	0.91974
Random Forest	0.77986	0.87365
k-Nearest Neighbors	0.75683	0.85729
Ensemble	0.65102	0.77711

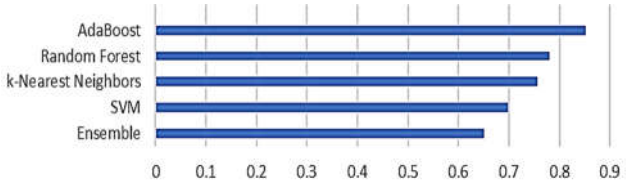

Fig. 11: Classification Model Accuracy Comparison Chart

VI. Conclusion

In this paper we investigated various machine learning algorithms to explore the relation of blood tests with COVID-19. We applied SVM, AdaBoost, random forest, K nearest neighbor and ensemble learning. A real dataset obtained from a Brazilian hospital has been used to test the models. The dataset was originally suffering from unbalancing, this forced us to balance the data using under-sampling. Scikit-Learn and other data science Python/R libraries have been used. As accuracy and F measure are considered, the AdaBoost gives the highest, whereas the ensemble learning gives the highest positive rate despite that the overall accuracy is the lowest. When compared to AdaBoost, Random Forest was less accurate by 9%, k-NN by 13%, and SVM by 21%. This research can primarily conclude that having normal blood tests do not help much in detecting COVID-19 since the true positive rate is very low in all the models. Also, the unbalance of data may also reduce the true positive rate. Future research direction would include applying the used and other machine learning models on more and various real datasets.

References

[1] A. Remuzzi and G. Remuzzi, "COVID-19 and Italy: what next?," *The Lancet,* 2020.

[2] N. Crokidakis, "Data analysis and modeling of the evolution of COVID-19 in Brazil," *arXiv preprint arXiv:2003.12150,* 2020.

[3] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, N. Rouf and M. M. U. Din, "Machine learning based approaches for detecting COVID-19 using clinical text data," *International Journal of Information Technology,* vol. 12, p. 731–739, 2020.

[4] R. Sujath, J. M. Chatterjee and A. E. Hassanien, "A machine learning forecasting model for COVID-19 pandemic in India," *Stochastic Environmental Research and Risk Assessment,* p. 1, 2020.

[5] M. Z. Alom, M. M. Rahman, M. S. Nasrin, T. M. Taha and V. K. Asari, "COVID_{M}{T}{N}et: COVID-19 Detection with Multi-Task Deep Learning Approaches," *arXiv preprint arXiv:2004.03747,* 2020.

[6] G. Pinter, I. Felde, A. Mosavi, P. Ghamisi and R. Gloaguen, "COVID-19 Pandemic Prediction for Hungary; a Hybrid Machine Learning Approach," *Mathematics,* vol. 8, p. 890, 2020.

[7] W. McKinney, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython, " O'Reilly Media, Inc.", 2012.

[8] E. Bisong, Building Machine Learning and Deep Learning Models on Google Cloud Platform, Springer, 2019.

[9] W. McKinney and others, "pandas: a foundational Python library for data analysis and statistics," *Python for High Performance and Scientific Computing,* vol. 14, 2011.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and others, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research,* vol. 12, p. 2825–2830, 2011.

[11] G. E. A. P. A. Batista, A. C. P. L. F. Carvalho and M. C. Monard, "Applying one-sided selection to unbalanced datasets," in *Mexican International Conference on Artificial Intelligence*, 2000.

[12] C.-c. Chang and C.-j. Lin, "Libsvm: A library for support vector machines. software available at http://www. csie. ntu. edu. tw/cjlin/libsvm," 2001.

[13] R. E. Schapire, "Explaining adaboost," in *Empirical inference*, Springer, 2013, p. 37–52.

[14] A. Liaw, M. Wiener and others, "Classification and regression by randomForest," *R news,* vol. 2, p. 18–22, 2002.

[15] L. E. Peterson, "K-nearest neighbor," *Scholarpedia,* vol. 4, p. 1883, 2009.

[16] R. Polikar, "Ensemble learning," in *Ensemble machine learning*, Springer, 2012, p. 1–34.