COVID-19 Preliminary Patient Filtering based on Regular Blood Tests using Auto-Adaptive Artificial Intelligence Platform

Zoltan Czako
Technical University of Cluj-Napoca
Computer Science
Cluj-Napoca
Email: Zoltan.Czako@cs.utcluj.ro

Gheorghe Sebestyen
Technical University of Cluj-Napoca
Computer Science
Cluj-Napoca
Email: Gheorghe.Sebestyen@cs.utcluj.ro

Anca Hangan
Technical University of Cluj-Napoca
Computer Science
Cluj-Napoca
Email: Anca.Hangan@cs.utcluj.ro

***Abstract*—In order to control the spread of the COVID-19 it is very important to identify those who have been already infected by this new type of virus. The rRT-PCR (reverse transcription polymerase chain reaction) testing is the golden standard for COVID-19 detection, but it is time consuming, laborious manual process and it is very short in supply. In order to reduce the number of tests, in this article we will present a possible solution for COVID-19 preliminary patient filtering based on regular blood tests, using artificial intelligence (AI) models. The most appropriate AI model will be selected using our auto-adaptive AI platform, AutomaticAI. The hyperparameters of the selected algorithm will also be adjusted automatically by this platform to match the context of the problem.**

I. Introduction

The Coronavirus disease 2019 (COVID-19) was identified in December 2019 in Wuhan, China [3] resulting the 2019-2020 pandemic. This new SARS-CoV-2 virus is very aggressive, because it is transmitted much more easier between humans than its closest relation, SARS. Computer modelling and simulation techniques suggest that each new COVID-19 case infects on average 2.67 other people [4]. The most effective way to fight against this new virus is social distancing, fast and early detection and isolation. Early detection can be very hard, because the incubation period (between infection and symptoms appearing) can range from 2 to 14 days (5 days is most common according to the World Health Organization)[?]. In many cases the infection is asymptomatic (infected, but with no signs or symptoms) or patients develop flu-like symptoms, fever, cough or shortness of breath. More aggravated symptoms are chest pain or pressure. Sometimes individuals infected by SARS-CoV-2 can show symptoms like sneezing, runny nose, vomiting, diarrhoea or sore throat. Because the symptoms can be similar to other respiratory infections or Influenza, early detection is difficult; it is hard to differentiate COVID-19 and Influenza based only on symptoms. Another way of detecting Coronavirus is to use a procedure called polymerase chain reaction (PCR) [1] or real-time reverse transcription polymerase chain reaction (rRT-PCR) [2]. The problem with this kind of tests is that they are slow and require special equipment and chemicals which are in short supply.

Under these circumstances, doctors must carefully choose who to test. One solution is to create groups of people, and make one test per group (mix the samples together and then make only one PCR test). If the test is negative, then we managed to test the whole group using only one test. If the test is positive then slice the group in two parts and follow the binary search rules.

A more effective way of selecting people to test is to use other, less expensive and more common tests to evaluate the risk of being infected by SARS-CoV-2. In this article we propose a COVID-19 preliminary filtering algorithm based on regular blood tests.

This algorithm can help doctors to choose who to test, but it is not meant for precise detection. Our algorithm can be used as a preliminary test, before applying COVID-19 detection tests. If it shows that there is a very low or zero percent of chance of infection, the patient can be considered healthy and no further tests are required. Otherwise, further investigation is needed and testing for COVID-19 is required. So this algorithm can be seen as an automatic pre-filtering of persons who are potentially infected or not by a virus (including the SARS-CoV-2). The filter could be improved regarding COVID-19 sensitivity if the training dataset would contain more cases of persons infected by a another virus.

The rest of this paper is organized as follows: in Section 2 we will present some interesting solutions for COVID-19 detection and classification and compare them with our work, in Section 3 we will describe the algorithm used for COVID-19 preliminary filtering and for explaining its results, in Section 4 we present experimental results and Section 5 concludes our paper.

II. Related Work

The ongoing pandemic situation generated a large amount of research in the area of COVID-19 classification and detection. Many researchers try to use traditional evaluation techniques (analyses), such as x-ray, CT or blood tests, in order to identify COVID-19 infection. One of the most popular and highly debated problem is detecting SARS-CoV-2 infections

based on CT or X-Ray lung images. In [5] we can see two different approaches to classify COVID-19 based on CT images. Firstly, the authors tried to classify the images without feature extraction, the images were transformed to vectors and the classification was made using the SVM algorithm [16]. Because the results were not so promising, they have tried a different approach, in which, before applying SVM, they used five different feature extraction algorithms, namely Grey Level Co-occurrence Matrix [17], Local Directional Pattern [18], Grey Level Run Length Matrix [19], Grey Level Size Zone Matrix (GLSZM) [20] and Discrete Wavelet Transform [21]. After running the experiments with each feature extraction algorithm, one-by-one, the best result was given by using GLSZM as feature extraction algorithm, which generates a vector of 13 features. The results of the GLSZM was used with SVM in order to classify the images. This way they managed to obtain an f1-score of 96.46+-3.7%.

In [6] we can see a very similar approach, applied to X-Rey images. The difference is that, before using the five feature extraction algorithm mentioned in [5], they've used the SMOTE algorithm [22] for over-sampling (because there are only a few x-Ray images for COVID-19 and a lot of examples for pneumonia and healthy patients) and after applying the feature extractors, they applied PCA [23] to reduce the dimensionality of the obtained vectors from 78 features to 20 most relevant features. On the obtained feature vectors SVM was applied. This way they've managed to obtain an f1-score of 84.5 percent, which is one of the best results obtained so far without using Convolutional Neural Networks (CNNs).

In [7] the authors solved the x-Ray image classification problem by using an ensemble of different type of CNNs. During the training process, they've applied model pruning, this way decreasing computational complexity and overfitting. Besides pruning, they've also preprocessed the images by cropping out only the lungs from the images, using lung masks generated by U-Net [8]. With this approach the authors of the article managed to obtain an f1-score of 97.82 percent.

In other papers, like in [10] we can find newly created Convolutional Neural Networks with custom layers, like projection, expansion and depth-wise representation. Using the newly created COVID-Net architecture, the authors obtained a Positive Predictive Value of 80% and a Sensitivity of almost 100%. In article [12] three different CNN architectures were tested using transfer learning. In this case, the best result was obtained using ResNet50, but the authors also tested the InceptionV3 [13] and Inception-ResNetV2 [14] architecture, which gave an accuracy of 87 respectively 97 percent.

In other articles transfer learning was used in order to classify images into healthy or infected categories.The authors of [9] used a ResNet50 [11] CNN to classify normal, and COVID-19 viral manifestations in x-ray images and achieved an f1-score of 98.19%.

In all of the articles mentioned above we can find solutions for COVID-19 classification based on different types of images (CT or x-Ray). But in the first, asymptomatic stage of the COVID-19 illness, when the probability of transmitting the virus is high, the effects of the virus on lungs (seen through CT or x-ray) is rather small. Therefor a lung image analysis would be a too-late test.

Therefore, in this article we will approach the problem of preliminary patients filtering from a different angle; we will use regular blood tests results to estimate the probability for a patient to be infected. The algorithm can be used as a prefilter, to choose who to test, this way reducing the number of required specific COVID-19 tests and also helping doctors to automatically separate patients with high risk of infection.

In the following section we will present the steps that we took in order to preprocess the data, to find the best AI model in the context of COVID-19 prefiltering and tune its hyperparameters.

III. COVID-19 prefiltering using AutomaticAI

A. Automatic preliminary patient filtering

The main objective of this work is to find an artificial intelligence algorithm, which can help the doctors to automatically filter out healthy patients and choose those patients for which further investigation or COVID-19 tests are necessary. Our idea is that we can use regular blood test results in order to filter out healthy patients and also to estimate the probability of a patient to be infected by a virus. If there is an infection, then further investigation is required to be sure that it is SARS-CoV-2 and not other types of viruses. So our initial goal is to classify healthy patients and patients susceptible of a viral infection, including COVID-19. As the publicly available dataset will include records of patients with other virus infections as well, our AI model and classifier may be trained for a better discrimination between COVID-19 and other viral illnesses. Therefore, in this stage of the research the present method won't replace the necessity for more accurate COVID-19 tests, it is rather meant as a preliminary, much cheaper and faster classification method.

In order to find the best model and the best hyperparameter configuration based on the input dataset, we used the pipeline presented in Figure 1.

As it is presented in Figure 1 in the first three steps we handled the problem of incomplete, missing or irrelevant data as follows:

1) We deleted the rows were the blood tests were completely missing;
2) We removed the columns which were irrelevant for the classification problem based on regular blood tests (so we removed every column which contained data that cannot be measured with only a regular blood test) ;
3) In columns with only a few missing data, we filled those in using the mean value of the respective column.

In the next steps we transformed (normalized) the data, to have the same magnitude for each feature (column). Than we handled the problem of imbalanced data using oversampling with a factor of 0.4 (the value was chosen experimentally, through a trial and error method).

Than in the most complex step of the processing flow, the cleaned, transformed and rebalanced data was fed into our

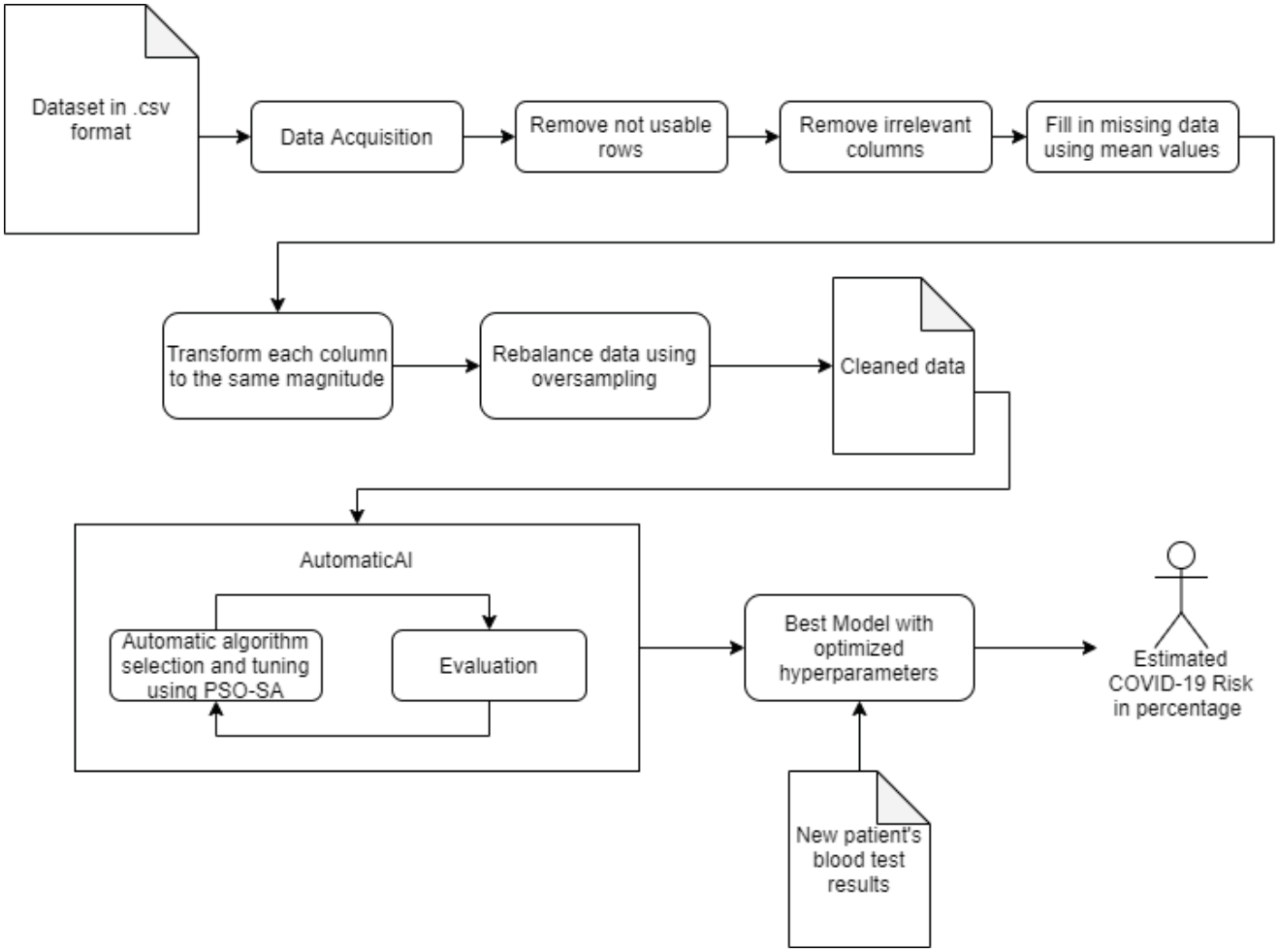

Fig. 1: Solution pipeline

AutomaticAI platform [25] here the PSO-SA algorithm was run in order to select the best classification method and its best performing setup. The algorithm is testing a number of AI classification models with different parameters setups until the best performing ones with the best hyperparameter setups are found. The method with the highest quality metrics results can than be used as a COVID-19 preliminary test for new potentially infected patients.

B. Dataset

The dataset [15] used in our experiments contains anonymous medical records of potentially COVID-19 infected patients seen at the Hospital Israelita Albert Einstein, at São Paulo, Brazil. All data were anonymized following the best international practices and recommendations. All clinical data were standardized to have a mean of zero and a unit standard deviation. Dates have been omitted and the information on the patient's sex has been coded. This dataset contains data about more than 5000 patients and it has over 100 medical features for each patient. Between features it contains the results of a regular blood test, with attributes like Hematocrit, Hemoglobin, Platelets, Mean platelet volume, Red blood cells, Lymphocytes, MCHC (Mean corpuscular hemoglobin concentration), Leukocytes, Basophils, MCH (Mean corpuscular hemoglobin), etc. Besides the regular blood test values, it also contains the results of different tests for viruses (different than Coronavirus) like Respiratory Syncytial Virus, Influenza A, Influenza B, Rhinovirus/Enterovirus, Parainfluenza 1, 2, 3, Adenovirus, etc. The ground truth column is given by the results of the rRT-PCR test; it is a binary column, having "positive" for patient infected by COVID-19 and "negative" for healthy patients. Our goal was to create an algorithm to estimate the possibility of being infected by SARS-CoV-2 based on the results of regular blood test. As a shortcoming of this dataset we can mention that there are few (not enough) examples of patients with viral infection other than covid-19. Because of this drawback our trained classification model is not a good discriminant between covir and other viral infections; as more complete datasets will be publicly available we will use them to improve our classifier.

C. Data preprocessing details

The most important issue with the dataset described above, is that it has lots of missing data, so preprocessing is absolutely necessary. Another problem is that the data is highly imbalanced, there are only 557 positive cases compared to 5016 negative cases.

The first step in the preprocessing phase is to remove the rows where the regular blood test results are completely missing. After this step we obtained a dataset of more than 600 patients with almost complete blood test attributes. In the next step we removed columns that are not relevant for our goal, so we removed the columns containing calculated attributes, attributes which were obtained from different sources than regular blood test results or attributes with more than 50% of missing data. This way we obtained a total of 27 relevant features. In order to be sure that we treated all the missing data, we ran trough the dataset again and filled out all the empty columns with the mean value of the respective feature.

All the columns were already normalized by the author of the dataset and each feature has very small values, usually in the interval of [-3, 3]. The only outlier is the column called "Patient age quantile", which contains values between 1 and 19. Because there are some AI algorithms in which the magnitude of the features is very important (can influence the final result), we transformed the age quantile column to have values between [-3, 3], as in the case of other columns, this way avoiding the negative impact of columns with different scales.

In the last step, we attenuated the problem of imbalanced data using oversampling, with a rate of 0.4 (40%). We chose this rate heuristically, by trying out multiple values in the range of (0,1] and paying attention on overfitting. Using 40% for oversampling, we obtained a dataset with 507 positive and 232 negative cases, this way reducing the difference between the two classes. The results of the oversampling can be seen in Figure 2.

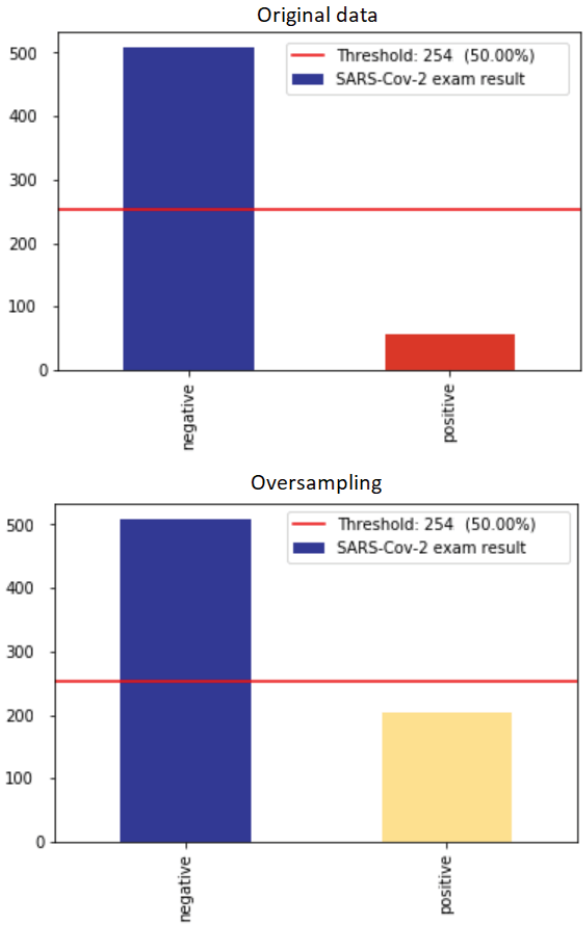

Fig. 2: Results of oversampling

D. The AutomaticAI platform

The platform developed by our research team, called AutomaticAI is a general purpose tool containing all the necessary functionality in order to conduct researches in the domain of artificial intelligence and more specifically in the domain of anomaly detection. The goal of the platform is to automate all the repetitive work that a researcher must do in order to clean the data, preprocess it, to search for the most appropriate AI algorithm, tune its parameters, evaluate and test it and finally to create a service or deploy it. The architecture of the platform was created with scalability, extainability and maintainability in mind, so it has a modular structure, each functionality is a separate module, in each module we have multiple components and any component or module can be easily replaced or new modules can be added this way extending the platform without the need of modifying existing code. In this platform we included all the functionalities that we considered necessary or useful for researchers in their daily work. These functionalities are the following:

1) Data acquisition - currently the platform supports three file formats, (csv, excel and arff), but it can be easily extended to read any kind of files
2) Data cleaninig and preprocessing - the platform contains algorithms to handle missing data, imbalance data using over or under sampling, handle anomalies based on statistical properties, different types of filters, like low-, high- or band-pass filters, Kalman filters, etc.
3) Data transformation - this module contains functionalities to normalize or scale the input data, to transform the data using different transformation techniques (for example Fourier), to find and select the most important attributes from the data and for dimensionality reduction (like Principal component analysis, Linear discriminant analysis, Non-negative matrix factorization or even Autoencoders)
4) Artificial Intelligence algorithms - contains most of the popular AI algorithms for classification, clustering and regression. Here we can find algorithms like kNN, SVM, Decision Tree, Random Forest, but also different type of ensemble methods like bagging, voting or boosting
5) Automatic algorithm selection and tuning - in this mod-

ule we can find our hybrid PSO-SA (Particle Swarm Optimization - Simulated Annealing) algorithm which was developed to automatically find the best model based on the input data and automatically optimize its hyperparameters. Besides this algorithm, here we can find basic model selection and tuning algorithms like grid search and it also contains third-party libraries like auto-sklearn and hyperopt-sklearn

6) Visualization - using the platform we can visualize the results of the models, we can generate different kind of statistics, like training curve, AUC, ROC, Confusion Matrix, Correlation Matrix, Box and Whisker plots and we can visualize the input data using different types of diagrams (like scatter plot, bar charts, line charts, etc.)
7) Generators - in this module we have some algorithms to enhance and enrich the input data, to generate synthetic data

Besides the functionalities presented above, in the future we would like to add the following modules:

- Explainability - this module will contain algorithms to explain the results of black-box models
- Automatic drift detection - this will contain algorithms to detect concept drifts in the input data and automatically attenuate their negative effects
- Automatic deployment - this module will be used to easily deploy AI services and automatically create secure APIs using different cloud providers (hybrid cloud) like Amazon, Google or Azure cloud.

In the following section we will briefly describe the PSO-SA algorithm used in our platform for automatic model selection and hyperparameter tuning.

E. Algorithm used for AI model selection and tuning

In order to search for the best artificial intelligence classification model and to tune the hyperparameters we used our hybrid PSO-SA (Particle Swarm Optimization - Simulated Annealing) algorithm included in the AutomaticAI platform. This is a general purpose algorithm, which can be used in any kind of classification or prediction problem and it will automatically adjust itself and select a matching algorithm based on the current context, and on the input data. By combining this algorithm with preprocessing and feature extraction steps, we can solve almost any kind of classification (or regression) problem with fairly high scores (accuracy, precision, recall, etc.). The advantage of this algorithm is that we don't have to manually search for AI models and evaluate them one-by-one, we don't have to manually search for hyperparameter values, which can be a very time consuming labor and sometimes impossible, taking in consideration that sometimes the number of hyperparameter combinations is very high (even infinite in the continuous multidimensional space). In the case of the PSO-SA algorithm included in our platform the number of particles per algorithm type can be configured manually and it also uses Simulated Annealing-like heuristics to avoid local minimum or maximum values (depending if it is an error minimization or score maximization problem).

In this algorithm, each particle represents an algorithm type, and there are multiple swarms/groups of particles (one swarm per algorithm type). Each of the groups has a leader, the particle with the highest evaluation score and there is only one global leader, the particle with the overall (not just per group) best metric. The movement of the individual particles is affected by the local leader, the global leader and the personal best position. In the next position acceptance criteria we also used a Simulated Annealing like criteria: at first the particles make bigger moves (some in a wrong direction), trying to avoid local minimums or maximums; as the number of iterations grow the dimension of the step is progressively reduced, in order to better approximate the optimal solution.

In each epoch the particle with the lowest score will be removed and the same number of new particles will be added to the group were the global best particle can be found. This way we solved the problem of model selection, because after some epochs, each particle will search the hyperparameter space of the same algorithm type. The hyperparameter tuning was solved implicitly, because each element of the particle's position vector is a hyperparameter of that specific algorithm type. For example, in the case of the SVM algorithm the position vector will have three elements, the C parameter or the regularization parameter, which is a continuous value, the gamma parameter which defines how far the influence of a single training example reaches (this is also a continuous parameter) and the last parameter is the kernel, which is a discrete parameter and can get four values, linear, poly (for polynomial kernel), rbf (radial basis function) or sigmoid. With this representation and abstraction we resolves the parameter optimization problem, because each position vector represents a different kind of parameter configuration, as well as the model selection problem, because at the end of the algorithm only one particle will be the global best, so only one algorithm type will be accepted.

IV. Experimental Results

A. Metrics used for model evaluation

In order to have a comprehensive evaluation of our solution we used multiple evaluation metrics:

1) *Accuracy:* The number of correct classification compared to the total number of examples;
2) *Precision:* The ratio of the correctly classified positives to the total number of positive classifications;
3) *Recall:* The ratio of the correctly classified positives to the total number of positives from the dataset;
4) *F1-Score:* The harmonic mean of Precision and Recall;
5) *Confusion Matrix:* A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. The confusion matrix shows the ways in which your classification model is confused when it makes predictions;
6) *Area Under the Curve (AUC):* Degree or measure of separability, it tells how much the model is capable of

distinguishing between classes. By analogy, the higher the AUC, better the model is at distinguishing between patients with disease or no disease.

B. Results of the best model

Fig. 3: Confusion Matrix for the test data

The PSO-SA algorithm is an optimization algorithm using some heuristics, so each run will give slightly different results. To demonstrate that the algorithm returns very similar results after each run, we ran it multiple times and each time we got an f1-score of 97+-2.0% which is a fairly stable result. The top three models with the hyperparamters and scores can be seen in Table I. In each case we ran our algorithm with the same number of particles and on the same hardware configuration.

As we can see in Table I, we obtained really high scores, the best f1-score was 99.8% in the case of the Extra Tree Classifier using 199 estimators.

The confusion matrix of the best model can be seen in Figure 3.

C. Explaining the results

The medical field, requires high level of accountability, and thus transparency, which means we need to be able to explain machine decisions, predictions and justify their reliability. In order to explain the results of the best model, (which in the current context is the Extra Trees Classifier), we evaluated the importance of the different feature in the final classification. Feature importance is available for more than just linear models. Most Random Forest (RF) implementations also provide measures of feature importance. If the best model would be different than Extra Tree or Random Forest, which implicitly calculates the feature importance, there is another technique to measure it, called Permutation importance [24]. Permutation importance is a common, reasonably efficient, and very reliable technique. It directly measures feature importance by observing the effect on model accuracy of randomly shuffling each predictor feature. The feature importance diagram can be seen in Figure 4.

The Extra Tree classifier is composed of 199 decision trees and the importance is calculated for a single decision tree by the amount that each attribute split point improves the performance measure, weighted by the number of observations the node is responsible for. The final feature importance are then averaged across all of the the decision trees within the model.

As we expected the most important feature is the Leukocytes or white blood cells (WBC) count. Having a higher or lower number of WBCs than normal may indicate an underlying condition. A WBC count can detect hidden infections within your body and alert doctors to undiagnosed medical conditions. There are five major types of white blood cells: neutrophils, lymphocytes, eosinophils, onocytes and basophils.

Because high basophilis is not related to virus, bacteria or other type of infections, but it is a usual sign of Hypothyroidism, in the feature importance diagram it appears very low. A high number of neutrophils, lymphocytes, monocytes or eosinophils it's a usual sign of viral infections, so, as we may expect, they have a much higher importance value in the diagram.

The second most important feature is the Platelets. Platelets are the cells that circulate within our blood and bind together when they recognize damaged blood vessels. Excess platelets are due to an infection or other condition. Usually the platelets count is evaluated in combination with the mean platelet volume (MPV), which is also an important feature as shown in the picture. A low platelet count along with high MPV occurs when platelets are destroyed, usually by antibodies, an infection, or toxins.

Patient age is also a very important factor, which is normal, because older people have a much higher risk of being infected.

Even if the patient has symptoms like shortness of breath, fatigue, high blood pressure, confusion or vomiting, if the level of Creatine is high, it's the sign of kidney failure or diabetes, so it is not caused by Coronavirus. This is the reason why Creatine has also a high score in the feature importance diagram (it is used to exclude Coronavirus).

High level of Alanine Transaminase can show liver problems. So if the patient has symptoms like weakness or fatigue, but the level of Alanine Transaminase is high, it is very likely that it is caused by liver problems and not by Coronavirus. So it can be used to exclude COVID-19 even if there are some similar symptoms.

The other features like Direct of Indirect Bilirubin, Potassium, etc. are also used by the model to exclude COVID-19 infections, so these features will decrease the probability of Coronavirus infections.

V. Conclusion

In this article we presented a possible solution for COVID-19 preliminary patient filtering using Artificial Intelligence. In the first part of the article we described the steps that are required to clean the input dataset, to handle missing data and to attenuate the effects of imbalanced training data. In the second part we presented our hybrid PSO-SA algorithm, which is a generic solution for automatic artificial intelligence model selection and hyperparameter tuning. We used this algorithm

Nr.	Accuracy	F1-score	Precision	Recall	AUC	Resulted Algorithm
1	0.981	0.964	0.942	0.987	0.978	RandomForestClassifier(criterion='gini', min_samples_split=8, n_estimators=155)
2	0.975	0.967	0.956	0.978	0.976	ExtraTreesClassifier(criterion='gini', min_samples_split=2, n_estimators=186)
3	0.987	0.982	0.998	0.968	0.983	ExtraTreesClassifier(criterion='gini', min_samples_split=2, n_estimators=199)

TABLE I: The results of the algorithm for the regular blood tests dataset

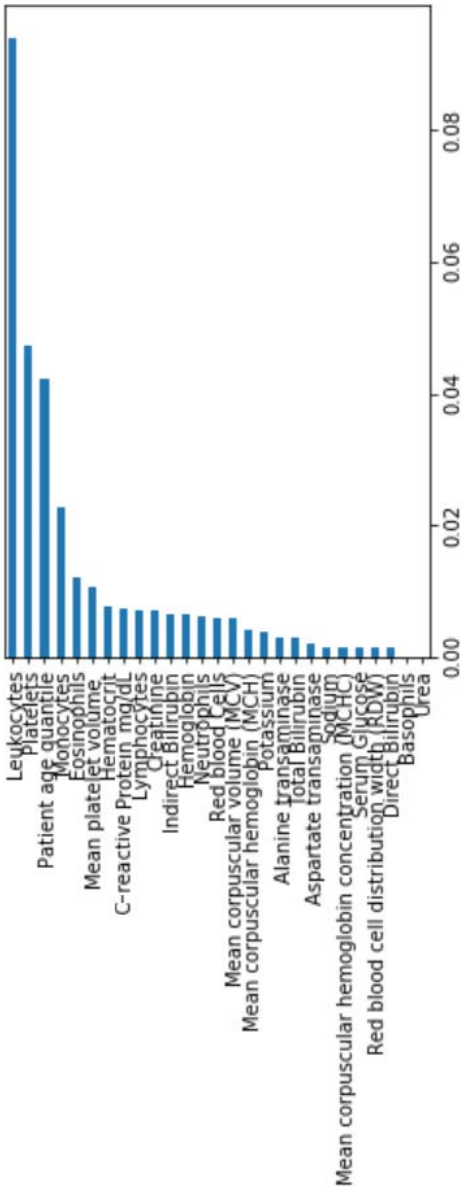

Fig. 4: Feature importance in COVID-19 risk estimation

to automatically find the best AI model and tune its parameter in order to match the context of the problem, to maximize the evaluation metrics like precision, recall or AUC.

After running the algorithm multiple times, we presented the top three models and the best of them was selected for preliminary patient filtering. The algorithm with the highest score was the Extra Tree Classifier with 199 estimators and it obtained an f1-score of 98.2% with a precision of 99.8% and 96.8% of recall, which is a really impressive result.

In the last part of the article we used feature importance in order to explain the results of the selected Extra Tree classifier. We explained the impact of each feature from the training data, this way helping the doctors to understand, interpret and evaluate the decisions of the algorithm. By explaining and understanding each result and decision of the algorithm, we tried to increase the trustworthiness and the usability of this algorithm in hospitals.

As new and more complete datasets regarding COVID-19 will be published in different countries, it would be very useful and practical to setup a continuous learning pipeline, in order to improve the classification performance of the model. It would be useful to monitor the phenomena of "concept drifts", which in this case would mean a change in the behavior of the virus (symptomatic changes) in time or related to different geographical regions or populations.

There is also a need for more clinical tests in order to see if the theoretically very good results are validated in the clinical practice. The input from the currant doctors would also influence the evolution of our research.

REFERENCES

[1] Wang et al. *Detection of sars-cov-2 in different types of clinical specimens*, JAMA, 2020.

[2] Udugama B, Kadhiresan P, Kozlowski HN, Malekjahani A, Osborne M, Li VYC, Chen H, Mubareka S, Gubbay JB, Chan WCW. *Diagnosing COVID-19: The Disease and Tools for Detection*, ACS Nano, 9th of April, 2020

[3] Wu Yi-Chia, Chen Ching-Sunga, Chan Yu-Jiuna *The outbreak of COVID-19*, Journal of the Chinese Medical Association, March 2020, Volume 83, Issue 3, p. 217-220

[4] Ying Liu, Albert A Gayle, Annelies Wilder-Smith, Joacim Rocklöv, *The reproductive number of COVID-19 is higher compared to SARS coronavirus*, Journal of Travel Medicine, March 2020, Volume 27, Issue 2

[5] Barstuğan Mücahid, Özkaya Umut and Öztürk Şaban, *Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods*, arXiv, March, 2020

[6] Barstuğan Mücahid, Özkaya Umut and Öztürk Şaban, *Classification of Coronavirus Images using Shrunken Features*, arXiv, April, 2020

[7] Sivaramakrishnan Rajaraman, Jen Siegelman, Philip O. Alderson, Lucas S. Folio, Les R. Folio and Sameer K. Antani *Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-rays*, arXiv, 16th of April, 2020

[8] Ronneberger O., Fischer P., Brox T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer, vol 9351, 2015

[9] S. U. K. Bukhari, *The diagnostic evaluation of Convolutional Neural Network (CNN) for the assessment of chest X-ray of patients infected*, medRxiv, 26th of March, 2020

[10] Linda Wang and Alexander Wong, *COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images*, arXiv, April, 2020

[11] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.
[12] Ali Narin, Ceren Kaya and Ziynet Pamuk, *Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks*, arXiv, March, 2020
[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens and Zbigniew Wojna, *Rethinking the Inception Architecture for Computer Vision*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
[14] Szegedy, Christian and Ioffe, Sergey and Vanhoucke, Vincent and Alemi, Alexander A., *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017
[15] Kaggle, 2020, Einstein Data4u, accessed 22th of April, 2020, https://www.kaggle.com/einsteindata4u/covid19/version/4
[16] Adankon Mathias M. and Cheriet Mohamed, *Support Vector Machine*, Springer US, 2009
[17] Vadakkenveettil Bino. *Grey Level Co-Occurrence Matrices: Generalisation and Some New Features*, International Journal of Computer Science, Engineering and Information Technology, 2012
[18] T. Jabid, H. Kabir, and O. Chae. Gender *Classification using local directional pattern (LDP)*, Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR'10, pages 2162–2165, 2010.
[19] Lifexsoft, 2018, accessed on 23th April, 2020, https://www.lifexsoft.org/index.php/resources/19-texture/radiomic-features/68-grey-level-run-length-matrix-glrlm
[20] Guillaume Thibault, Bernard Fertil, Claire Navarro, Sandrine Pereira, Pierre Cau, Nicolas Lévy, Jean Sequeira and Jean-Luc Mari, *Texture indexes and gray level size zone matrix. Application to cell nuclei classification*, 10th International Conference on Pattern Recognition and Information Processing, 2009
[21] Furht, Borko, *Discrete Wavelet Transform (DWT)*, Springer US, 2008
[22] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research 16, 2002
[23] Mishra Sidharth, Sarkar Uttam, Taraphder Subhash, Datta Sanjoy, Swain Devi, Saikhom Reshma, Panda Sasmita and Laishram, Menalsh, *Principal Component Analysis*, International Journal of Livestock Research, 2017
[24] André Altmann, Laura Toloşi, Oliver Sander, Thomas Lengauer, *Permutation importance: a corrected feature importance measure*, Bioinformatics, Volume 26, Issue 10, 15 May 2010, Pages 1340–1347
[25] Czako Zoltan, *AutomaticAI - Platform for Automatic Evaluation, Optimization and Selection of Artificial Intelligence Algorithms*, Dissertation, Technical University of Cluj-Napoca, 2019