Diagnostic and Prognostic models for Ebola Machine Learning

Department of Computer Science, EPF Lausanne, Switzerland
Jean Naftalski, Cédric Roy, Lavinia Schlyter

***Abstract*—Machine learning is associated to various applications with a popular one being classification; where the goal is to distinguish two (binary) or multiple classes using collected data. The goal of our project is to use different interpretive models to help doctors in the diagnosis and prognosis of Ebola. These should highlight important symptoms and other features that are related to the virus. We will also estimate the gCO2eq for running our models. The best predictive model for both diagnosis and prognosis was SVC with 0.69 AUC, 0.73 accuracy and 0.61 AUC, 0.64 accuracy respectively. The most imporant features for the outcome of the contaminated patient was IV care.**

I. Introduction

The Ebola virus disease (EVD) first appeared in 1976 [1] in Central Africa and its largest outbreak occurred between 2014-2016 in West Africa. However new outbreaks have been identified since, with the most recent one announced on the 1st of June 2020 and officially terminated on November 19th according to the Health Minister of the Democratic Republic of Congo [2]. There is no proven cure for Ebola, but if diagnosed early, the chances of survival can increase dramatically. The overall fatality rate is about 50 percent, according to the World Health Organization (WHO), although this figure is considered to be lower for rich nations. The first challenge lies here, it is difficult to differentiate EVD from other infectious diseases such as typhoid, malaria, but also meningitis, all of which have similar symptoms.

This paper presents several models, the findings of which are intended to be used by physicians/doctors to minimize the fatality rate by using a more effective diagnostic method or even a prognostic tool.

During the severe outbreak of 2014 the sharing of information among different medical centers was sparse. They undoubtedly worked for the common objective of saving lives and shared some of their findings, they did however lack coordination regarding the formatting of the laboratory results and clinical observations. This last step was undertaken by the Infectious Diseases Data Observatory (IDDO) which assembles information of diseases on a collaborative platform used not only by the health but also by the but also by the humanitarian and scientific communities.

The dataset assembled for Ebola by IDDO contains 14 studies each with a different number of files, containing patient data, laboratory data, clinical data and several others. A number of studies were curated but after some data analysis it turned out that the following two were the most appropriate for the project. EIXUZQ was data collected by Médecin Sans Frontières (MSF) in Foya, Liberia and EGOYQN by MSF as well, in Guéckédou, Guinea. We focused mostly on the second study as it had both the 'linelist' and the 'clinical' data.

To run, share and collaborate on the code, the open-source web application Jupyter Notebook [3] is used with Python3 as well as GIT [4].

II. The Data

A. The Datasets

As mentioned in the introduction, we worked on three different datasets.

Study 1: Consisting of a line list file which is a summary of the patient, that contains information such as the sex, age, name (confidential) but also the start date for the symptoms and the 'FinalStatus' feature which represents whether the patient tested positive for Ebola.

Study 2: For the second study we were provided with a line list dataset which was similar to the previous study but also a clinical one. The later corresponds to the daily observations of each patient who has been sent to the hospital. The patients were examined several times a day, and for the same patient, the number of examinations differed daily. The characteristics are presented in a similar format to the linelist, but with a greater focus on the patient's physical condition, dropping the less medical-related details, such as his civil status, formation, etc.

B. Visualization of the data

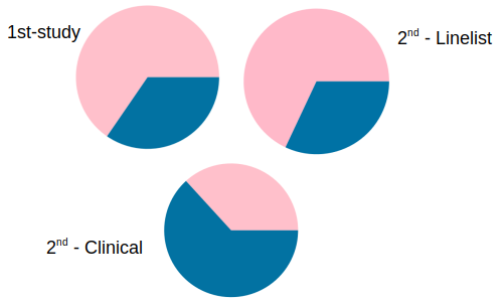

Fig. 1. Proportion of missing data (pink) and valid data (blue).

Study 1: The first study contains 871 patients and 86 features. Figure 1 shows the distribution of missing data for each feature, where dark blue denotes the available data and and pink the missing data.

Study 2-linelist: The second study contains much more information. The linelist file contains 2500 patients and 187 features. However, a large proportion of values are missing for each feature, as seen in figure 1.

Study 2-clinical: The clinical file is event larger with over 13000 rows of observations. Even if it was less empty than the previous datasets it still needed a strong overhaul as many rows complement each other and could be simplified as one.

C. *Preprocessing and feature engineering*

1) Dealing with missing and incoherent data:

The datasets were attached with dictionaries, but some features had unknown correspondences, we had to discard these features. Several features had legends which were not uniform for the same meaning. For example, "no", "negative", "Negative" with some even containing typos. We therefore had to apply several dictionaries to all datasets.

Study 1: In order to deal with the important set of missing data, we proceeded by dropping features with zero "non-null" elements. 13 patients with missing age were discarded. Incoherent data was also found such as having two F716 in the feature "numero" which corresponds to a patient's unique identifier. One was renamed "F716_bis". Negative referral times, which are interpreted as the patient having its case reported before the date that the illness started, were also dropped. 359 patients with all symptoms unknown were dropped as they cannot bring further information to the model. In order to keep as many features and samples as possible, patients with several unknowns were assumed to be negative in the associated symptom.

Study 2-linelist: As with the previous study, there is a lot of missing data, so the preprocessing required a lot of effort and adjustments as our expertise grew on the subject. In the end, three different ways of preprocessing were done. We began by ensuring that at least one symptom was present for all patients. All entries under the symptoms were left blank for 612 patients, so we deleted them. For the patients whose symptoms were known, they all had at least one positive symptom so no other patients were removed. Then, in order to obtain a simple model on which we could get our first results, we decided to remove all missing values. This data frame is named *df_simple*. This process was done using an algorithm that first removes features with more than 50% missing values, then eliminates patients with at least one missing value. This naive preprocessing leads to a loss of about 70% of the data. The second preprocessing enabled us to limit the loss of data. Rather than removing the missing values, we tried adding an indicator column to note where the missing data was before being replaced. This data frame is named *df_extra*, and a detailed explanation of the approach is given in the *Feature expansion* section. Finally, further improvements allowed us to obtain a third data frame named *df_ml*, on which we achieved our best scores (presented in Results). An explanation of our approach is also given in the *Feature expansion* section.

Study 2-clinical: The clinical file also had several missing entries. Either completely blank or with a message among "Unknown", "ABSENT", ".", etc. A few rows with negative entries or other incoherent content were also simply left out.

2) Feature expansion: For each of the studies, we encoded categorical data through one-hot encoding [5]. In the first study this operation is essentially the only kind of feature expansion we applied.

Study 2-linelist: For the first data frame *df_simple* of the linelist file, no feature expansion was performed since we only removed missing values. Nevertheless, for the second data frame *df_extra*, we added for each feature an additional feature called "indicator", which will give information on whether the value is missing or not.

For the third data frame *df_ml*, we have additional information about the patients movements and contacts. For these features a one-hot encoding was also performed. Then we divided the symptoms into separate classes, following our supervisor's idea. The first group of symptoms contains almost almost all observations, the second group the symptoms with 75% missing values and a third one the symptoms with over 96% missing values. This third group has been removed. For the first group, we removed all the patients with unknowns.
Concerning the second group, the steps are more complicated since some patients have only 'unknown's in symptoms (1348 patients), others have both filled and blank entries (22 patients), and and thus only less than a quarter (433 patients) have no missing values. Knowing this, the idea is to simplify the one-hot encoding: instead of adding a column indicating the position of the missing values for each symptom, a single column is added at the end of the data frame indicating whether the patient has only completed entries (value $= 0$) or only empty entries (value $= 1$). To do this, we have to delete the 22 patients who have both missing and existing values. In short:

- the first group of patients does not require one hot encoding as all their symptoms are known;
- for the second group a special one hot encoding has been performed, as described in the previous paragraph;
- the third group has been deleted since too many values were missing.

Study 2-clinical: The clinical data frame contains one row for each observation (where several may occur during a same day). In order to get a "time-variant" model, we first decided to group the observations per day; For this each type of features needed a different fusion operation. The features under the list "symptoms" had either entries 1 (for positive), 0 (for negative) or NaN (for unknown). We considered that if the patient had a symptom at one point during the day, the patient would have it during the whole day. This assumption is backed by the fact that questions about symptoms may not be asked several times a day. We therefore had to make a custom logic gate which prioritizes ones over zeros and zeros over NaN's. For more continuous data, like body temperature or heart rate, we fused them by just taking the mean of their values across the day. And lastly for the categorical data we decided that, similarly to symptoms, either the patient fell into a single category for the day, or it was left blank. We then rearrange the data frame to have only one row per patient with all his observations aligned chronologically. This reshaping of dataset from a tall matrix to a long one allows for time interpretation in our models. However the right side of said matrix is mostly empty as patients did not all stay under observation for the same duration. We had to compromise between many features but most of them left blank and fewer features with more complete information, and decided to only take into account the 5 first days of observations. The figure 2 shows the correlation matrix for that new data frame. There are two main observations to note from this matrix; First, the 5×5 structure can still be faintly seen. Second, following that block structure, three to four diagonal stripes stand out going through these blocks. They represent the correlations between a feature and itself the following day(s), This indicates that a patient's condition has some degree of dependence to the previous day's condition, which is to be expected.

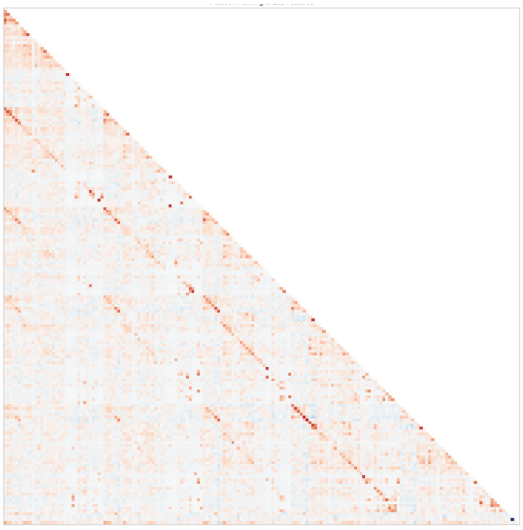

Fig. 2. Correlation Matrix for clinical once rearranged. Darker points represent stronger correlation.

3) Split of the dataset and representativity of the training set: In supervised learning, it is required to have labeled training data but for this to be effective, the training set must contain a certain number of the possible labels. In both studies we used a stratified train test split, that is, the ratio of both labels between the training and the test set will remain constant. This choice is to make sure both sets will contain both labels but also even if the proportion of Ebola positive patients was well over half, it is what we expect during an epidemic at the hospital.

4) Feature scaling: The feature scaling method is the same for all datasets. Standardizing helps with methods such as SVC and logistic regression. It is important to note that the test set must be standardized using the same shift and scaling factor as the training set.

5) Feature selection: The datasets are rich in information, several redundant and unnecessary features should thus be discarded. Firstly, we remove features with no variance. In addition to providing no additional information (the feature is constant), they can lead to matrix singularity problems. Secondly, we look at the high correlation between features, as they also can negatively impact algorithms by making the matrix hardly invertible.

Study 1: For the data set, we decided to keep the symptoms (list given in the notebook), information on the patient such as the sex, age and referral time which corresponds to the time between the date of the illness started and the admittance to the clinic. We also kept the target label "FinalStatus", which states whether the patient was tested negative or positive. In order to refine our results, we performed backward elimination, recursive feature elimination and ensemble methods to select the most important features.

Study 2-linelist: In the second study, as in the first one, we selected all the symptoms as well as information about the patient (age, sex, etc.). The main difference with the previous study is that we have, for the *df_ml* data frame, added more than fifty features providing information on the patient's residence, their occupations, their contacts, etc. We also used the feature selection methods which are explained in the following paragraph (study 2-clinical), to refine our models.

Study 2-clinical: In this last study we used more advanced tools to compute the feature importance. Before the fine tuning of the methods and hyper-parameter choice, we ask the machine to find which features to keep based on their contribution to the outcome. We decided to use a univariate method called ANOVA f-test, which looks at whether the mean of various sample data come from the same distribution. Targets independent of the target variable will be removed when training the model as they bring no utility for prediction. We use 'SelectKBest' from the scikit-learn [6] with "f_classif" as a scoring function.

D. *Models and Methods*

The goal is not to get the best accuracy and area under the curve by just running **black box models** but to run an interpretable model. Where as stated by Miller "Interpretability is the degree to which a human can understand the cause of a decision" [7].

We have thus chosen the models described below, a more exhaustive list of interpretable models can be found in Molnar's book [8]:

- Decision Tree
- Random Forest [RF]
- Logistic regression
- SVC
- XGBoost: This algorithm which stands for "Extreme Gradient Boosting" is decision-tree based that uses gradient boosting in order to increase speed and performance. This model was run for the first study, to see whether results would improve.

Hyper-parameter tuning: Each model has a number of different hyper-parameters defined as parameters which are to be tuned before the learning process begins.

We find the best combination of hyper-parameters by using a grid-search and evaluating them with cross-validations. When doing the grid-search we used a pipeline to combine the model and the best 'anova__K' value. For the first study we defined well known ranges and for the studies that followed we used validation curves in order to better visualize the effects of the various hyper-parameters on the model. At first we used stratified K fold, but in order to further decrease the variance we used repeated stratified K folds.

Metrics for accuracy: Different metrics may be used when analysing the performance of a model, since we have a fairly unbalanced dataset we use "Area Under the Receiver Operating Characteristic curve" (AUC_ROC). It is defined at the expectation that a uniformly drawn random positive is ranked before a uniformly drawn random negative. It tells how much the model can distinguish between two classes rather than calculating the correct number of predictions on the set (accuracy).

Carbon footprint: In this paper we also used a tool to quantify the carbon footprint of our Notebooks named "cumulator 0.0.7" developed by iGH [9]. Results are reported in Table II.

III. Results

A. *Discussion*

The confusion matrix of our best model for diagnosis is shown in Table I. This matrix is a key element in the medical field, where it is necessary to minimise false negatives (sick people

TABLE I
SUMMARY REPORT FOR DIAGNOSIS IN SECOND STUDY

True class	Confusion Matrix		Classification Report		
Ebola (-)	67	49	0.60	0.58	0.59
Ebola (+)	45	184	0.79	0.80	0.80
Predicted	Ebola (-)	Ebola (+)	Precision	Recall	F1

TABLE II
SUMMARY OF RESULTS FOR SECOND STUDY.

		Ref	SVC	Logistic	RF	gCO_2eq
Diagnosis	Linelist	AUC: 0.5 ACC: 0.66 F1: 0.8	0.69 0.73 0.80	0.65 0.70 0.78	0.65 0.73 0.81	226.8
Prognosis	Linelist	AUC: 0.5 ACC: 0.65 F1: 0.79	0.56 0.63 0.74	0.51 0.61 0.75	0.54 - -	967.8
	Clinical	AUC: 0.5 ACC: 0.58 F1: 0.73	0.61 0.64 0.71	0.58 0.62 0.71	0.60 0.64 0.73	345.3

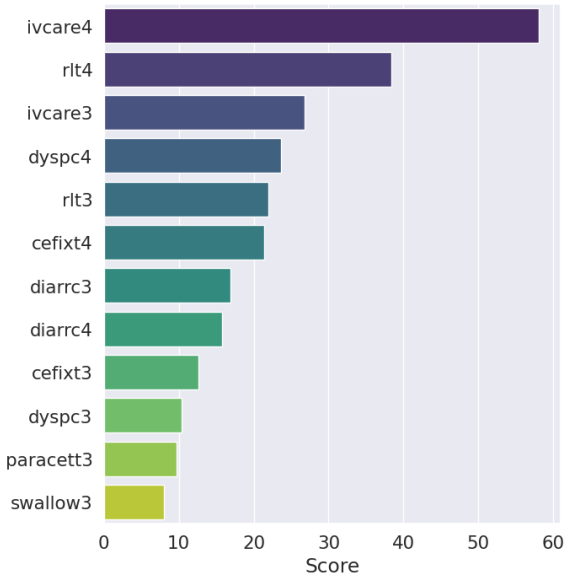

Fig. 3. Feature importance for clinical dataset

incorrectly identified as healthy). Indeed, missing an infected patient not only will put his life at risk but also expose all his entourage to the infection. In our case, 45 out of 229 EVD positive patients were false negatives, but to interpret more precisely the impact of this value on our model, let us calculate 3 quantities derived from this matrix. First, the precision which can be seen as a measure of the classifier's exactness. It gives the percentage of our Ebola predictions that were correct. Then, the recall measures the classifiers completeness, i.e. its ability to detect all the Ebola positive instances. Finally, the F1 score gives the percentage of positive predictions that were correct. In our case, as the Ebola class should absolutely be correctly determined, we are particularly interested in the values of precision, recall and F1 for positive Ebola class score. All these values are around 80%.

For diagnosis we achieved the best Area-Under-the-Curve (AUC), and accuracy (ACC) using SVC, 0.69 and and 0.73 respectively compared to the reference "Ref" results 0.5 AUC and 0.66 accuracy which correspond to predicting only Ebola positive patients[1]. Similar results have been presented in (A. Colubri, M-A Hartley et al.) [10], with AUC (0.70 - 0.79) and accuracy (0.64-0.74) using a multivariate logistic regression. It is important to note that they had access to the viral load which contributed over 5-fold the weighting of other features. We would therefore expect our results to improve had we have access to such data. For prognosis, the "linelist" dataset did not provide enough data, more importantly continuous data to make satisfactory predictions. In order to improve these results, we used the clinical dataset which provides daily observations. This increased the AUC to $0.61(+0.05)$ and accuracy to 0.64. Figure 3 depicts the 12 highest F values, it is very interesting to note that "the outcome" of the patient is correlated with features from the two days preceding the prediction (4th and 5th day, represented by 3 and 4 since labelling starts at 0). Intravenous care (IV) was selected as the most important feature, which is a frequently recommended intervention for patients with EVD even though its true impacts remain unclear [11]. Other features such as "rlt" (Ringer's lactate) used for fluid resuscitation [12], but also "dypspc" (shortness of breath) and diarrhea, both typical symptoms for Ebola [13].

B. *Limitations and future work*

We encountered several limitations such as:

- not having the viral load to better draw a prognostic;
- predicting for patients that stayed more than 5 days which could induce errors as features for those patients might change preceding the days of their outcome;
- the quality and amount of available data.

Further predictive models could involve:

- merging more studies together for more complete information;
- the assistance of external health experts to create interpretable trees from the models;
- continuously working to collect and share data, all this within privacy regulations.

C. *Conclusion*

In this paper we conducted a binary classification for the diagnosis and prognosis of Ebola. On the basis of two studies, we have been able to predict with 0.69 AUC and 0.73 accuracy if the patient is EVD positive, and with 0.61 AUC and 0.64 accuracy the outcome of the contaminated patient. We presented general models rather pushing the limits of accuracy for the specific data frames. A few recommendation points for future research were also given in order to possibly improve the results.

REFERENCES

[1] Ebola virus disease, fact sheet, world health organization, Feb 2020. https://www.who.int/news-room/fact-sheets/detail/ebola-virus-disease.
[2] Ebola virus disease, latest reported case, world health organization, Nov 2020. https://www.who.int/csr/don/18-november-2020-ebola-drc/en/.

[1]The results of study 1 are not presented here as they were part of out preminilary work but are present in the Jupyter Notebooks. We did not compute the accuracy and F1 score for RF as we pursued with a more complete dataset

[3] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[4] Tom Preston-Werner Chris Wanstrath, P. J. Hyett and Scott Chacon. Where the world builds software, 2008.

[5] Sarah Harris and David Harris. *Digital design and computer architecture: arm edition*. Morgan Kaufmann, 2015.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences, 2018.

[8] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[9] Trsitan Treabol. cumulator.

[10] Andres Colubri, Mary-Anne Hartley, Matthew Siakor, Vanessa Wolfman, August Felix, Tom Sesay, Jeffrey G Shaffer, Robert F Garry, Donald S Grant, Adam C Levine, et al. Machine-learning prognostic models from the 2014–16 ebola outbreak: data-harmonization challenges, validation strategies, and mhealth applications. *EClinicalMedicine*, 11:54–64, 2019.

[11] Adam R Aluisio, Derrick Yam, Jillian L Peters, Daniel K Cho, Shiromi M Perera, Stephen B Kennedy, Moses Massaquoi, Foday Sahr, Michael A Smit, Tao Liu, et al. Impact of intravenous fluid therapy on survival among patients with ebola virus disease: an international multisite retrospective cohort study. *Clinical Infectious Diseases*, 70(6):1038–1047, 2020.

[12] World Health Organization et al. Optimized supportive care for ebola virus disease: clinical management standard operating procedures. 2019.

[13] Eleonora Lalle, Mirella Biava, Emanuele Nicastri, Francesca Colavita, Antonino Di Caro, Francesco Vairo, Simone Lanini, Concetta Castilletti, Martin Langer, Alimuddin Zumla, et al. Pulmonary involvement during the ebola virus disease. *Viruses*, 11(9):780, 2019.