Application of clustering methods for detecting critical acute coronary syndrome patients

Magoev, K.; Krzhizhanovskaya, V.V.; Kovalchuk, S.V.

DOI
10.1016/j.procs.2018.08.277

Publication date
2018

Document Version
Final published version

Published in
Procedia Computer Science

License
CC BY-NC-ND

Link to publication

Citation for published version (APA):
Magoev, K., Krzhizhanovskaya, V. V., & Kovalchuk, S. V. (2018). Application of clustering methods for detecting critical acute coronary syndrome patients. *Procedia Computer Science*, *136*, 370-379. https://doi.org/10.1016/j.procs.2018.08.277

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 136 (2018) 370–379

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Young Scientist Conference on Computational Science

Application of clustering methods for detecting critical acute coronary syndrome patients

Kirill Magoev[a,]*, Valeria V. Krzhizhanovskaya[a,b], Sergey V. Kovalchuk[a]

[a]ITMO University, 197101, 49 Kronverksky pr., St Petersburg, Russia
[b]University of Amsterdam, 1012 WX Amsterdam, The Netherlands

Abstract

We investigate the applicability of two clustering algorithms, DBSCAN and k-means, to detection of critical (died) patients using medical parameter time series. In addition, we perform preliminary cluster analysis of outliers. The most important motivation behind this paper is the potential use of these methods in real clinical setting within an automatic early warning system to potentially decrease the in-hospital mortality rates. Investigation of the outlier clustering is an important step towards the outlier analysis that will help finding out the cause of these outliers: a human error in the records, a technical mistake in the lab, or an actual complication of a patient's disease. Our results demonstrate that specific clustering algorithms achieve a moderate performance in critical patient detection (F1 scores0.489 for DBSCAN and 0.495 for k-means). The patient classification boundary is very complex and could not be accurately detected by thesealgorithms, but there are methods capable to determine such boundaries, e.g. the high-degree polynomial kernel SVM. Regarding the outlier clustering, we perform a preliminary analysis showing that it is a viable option for potential outlier classification and analysis.

Keywords: Machine Learning; Anomaly Detection; Acute Coronary Syndrome; Clustering; Classification

1. Introduction

As large databases of electronic health records (EHR) become available for scientists (e.g. public Medical Information Mart for Intensive Care dataset, a.k.a. MIMIC[1]), more and more researchers use this opportunity to

* Corresponding author. *E-mail address:*magoev.kirill@niuitmo.ru

10.1016/j.procs.2018.08.277

study the applicability of various statistical, machine learning, and data mining methods with this data to develop various automated early warning or decision support systems and predictive models to use in hospitals. Therefore, novel models and methods aimed at predicting the treatment outcomes, diagnosis or emergency situations are being developed. As a result, computational approach is fast becoming a key instrument in the field of medical outcome prediction. This paper intends to determine the extent to which such approach is viable for the real-time medical application using the time series of various blood parameters of patients with Acute Coronary Syndrome. The set of investigated methods includes DBSCAN and k-means clustering algorithms.

Some researchers have conducted empirical studies investigating the applicability of various machine learning methods with available EHR databases. For instance, Yakovlev et al. [2]examined the k-nearest neighbours algorithm, logistic regression, random forest, and naive Bayes classifiers for early in-hospital mortality prediction and artificial neural network for length of stay prediction. Celi et al. [3]suggested a use of logistic regression, Bayesian network and artificial neural network to develop a mortality prediction model for patient data from MIMIC database. Steenard et al. [4]developed a model for predicting the risk of developing congestive heart failure using more than 3000 various medical, demographics and lifestyle-related risk factors from more than a million patients using decision tree and artificial neural network.

The predictive power of models that use features extracted from free-text medical notes have also been examined. Ghassemi et al. [5]utilized Latent Dirichlet Allocation to decompose the unstructured text data into meaningful features. In order to identify the classification boundaries, the authors applied a linear kernel SVM. Grnarova et al. [6]employed a two-layer system: the first layer maps sentences to sentence vectors, the second one constructs an individual patient representation from a combination of these vectors. For both levels, convolutional neural network with max-pooling is adopted.

Wojtusiak et al. [7]described a machine learning approach and proposed a computational model C-LACE that aims to improve the popular models LACE[8]and LACE+[9]. The authors tested a variety of machine learning methods including logistic regression, random forest, naive Bayes, and SVM to arrive at the most accurately predicting model.Presbitero et al. [10]applied the methods of anomaly detection in time series to the clinical data. The authors applied metric based indicators of parameter distribution moments, time-varying autoregressive model and Mann-Kendall trend test for detection of increase or decrease in the anomaly indicators.

It is a widely held view that the larger the training dataset is, the better the machine learning methods perform. In contrast, some researchers [11,12] hypothesize that using only data of subset of similar patients instead of the whole dataset for training machine learning methods might improve the accuracy of mortality prediction for individual patients. The authors employed a cosine similarity[11] and random forest proximity measure [12] as patient similarity metric to identify a subset of the most similar patients and use it to train one of three predictive models: death counting - empirical mortality rate across the most similar patients; logistic regression, and decision tree. The results confirm the hypothesis that all the methods perform best when trained with the limited set of the most similar patients.

While application of machine learning methods to medical data is a popular research topic, wide majority of scientists prefer to utilize supervised approaches, e.g. random forest, neural network, SVM, or logistic regression, because they tend to perform better than unsupervised learning methods. However, unsupervised methods have their own advantages: by definition, they do not requirea priory label specification, which makes them applicable to a wider range of datasets. Furthermore, they are generally less expensive computationally and simpler, therefore the results they produce are easier to interpret. In the present paper we investigate the applicability of two unsupervised learning methods,DBSCAN and k-means clustering algorithms, for real-time anomaly detection based on the time series data of 3650 acute coronary syndrome patients. We assess the performance of the algorithms on different time intervals and describe the highest clustering performance we could achieve on each of them. In addition, we test the viability of DBSCAN clustering algorithm for outlier clustering.

The paper is structured as follows: the section after the current one describes the methodology of data extraction, preprocessing, and feature selection, as well as general experimentation algorithm and description of specific clustering methods. The next section contains results, which include sensitivity analysis for each method and show achieved performance. We close the paper with conclusion and description of future directions for the research.

2. Methodology

2.1. Data preprocessing

The data for this research has been provided by the Almazov National Medical Research Centre (St. Petersburg, Russia). It consists of 6372 electronic health records describing patients with Acute Coronary Syndrome. Each record is an XML filethat includes medical observations, test results and reports compiled and maintained by the medical doctors and nurses of the Centre. The original set of records was combined and transformed into a structurally consistent form. 4196 records with patients having time series data were extracted. In 4095 of these cases treatment was successful and outcome was positive, while a small minority of patients (namely, 101) deceased. Time series of 506 unique “parameter name – units” pairs were extracted from the initial dataset.

Specifics of time series of medical parameters is that they are usually inconsistent, sparse, and heterogeneous, have varying lengths and irregular sampling intervals. Apart from this, different parameters of the dataset have vastly varying population, which poses additional challenges. There are also issues specific when the data is used by machine learning methods. For instance, 195 parameters were measured only for patients who were treated successfully, and 7 parameters – only for patients who passed away. Such features are examples of parameters that are often impossible to use with classification methods.

Figure 1 shows the properties of the dataset. We see that the majority of parameters were measured in a small number of patients and just a few times over the treatment history of a patient (low number of samples per time series). There is however a group of parameters having both significant number of patients and relatively long time series (encircled in Figure 1). We therefore pick the features having both significant number of patients and long time series. Out of 506 parameters, we pick a set of 16 features, which includes platelet distribution width, mean platelet volume, mean red blood cells volume, glucose concentration, red blood cells, monocytes, lymphocytes, mean hemoglobin per red blood cell concentration, white blood cells, platelets, creatinine concentration, hemoglobin concentration, mean hemoglobin per red blood cell, red blood cells distribution width, hematocrit, and neutrophils. The data has been filtered so it contains only patients having measurements of all these parameters. The resulting dataset contains 3650 patients, 93 of which died (critical patients) and the rest were discharged (normal patients).

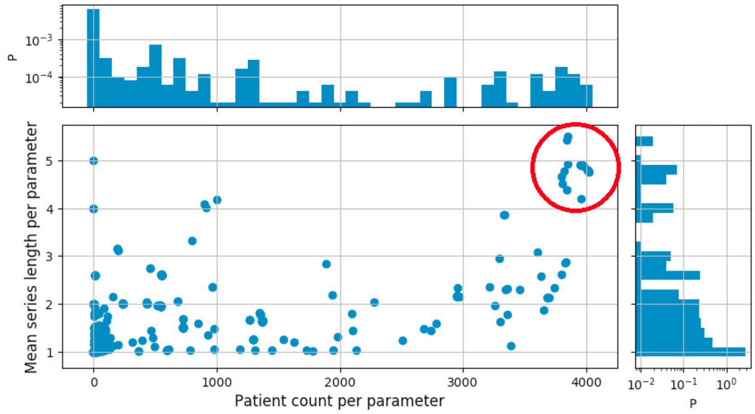


Figure 1. Distributions of patient count andmeannumber of samples in a time series per parameter. Each point in the scatter plot represents one parameter (out of 4196). Top and right graphs show the probability density functions. Encircled are the 16 parameters used in this paper, with a sufficiently high number of patients and long time series.

In order to overcome some of the issues of medical time series, such as sampling inconsistency and varying length, researchers utilize various approaches. For instance, Ghassemi et al.[13]proposed to use the technique of transforming time series data into a new latent space using the hyperparameters of Multi-Task Gaussian Process (MTGP) models. Batal et al. [14]developed the STF-Mine algorithm (Segmented Time series Feature Mine) to collapse multivariate time series data into automatically generated temporal features for classification using frequent pattern mining method; Caballero and Akella [15] chose to fill out the missing values using Regularized Expectation

Maximization method. In the first stage of our work, we resample the time series data over regular intervals (one day by default) using linear interpolation for the points that are within the original sampling boundaries and extrapolate by a nearest neighbour when a new point is outside. Time values of the series were recalculated as their offsets from the last measurement. In addition, all the measurements were normalized by median of the parameter they belong to. As a result, each patient is represented by a combination of 16 regularly sampled time series having equal lengths.

There are a number of approaches to applying the clustering algorithms to the time series data. For instance, it is possible to perform cluster analysis to the values at each moment in the time series and investigate the dynamics, or develop some novel representation for such data [13,14]. In this research, we decided to investigate the performance of the algorithms on the binned vector values on various time intervals. Preliminary examination of this approach includes analysis of dynamics of centroids and standard deviations for both groups of patients (critical and normal) on these time intervals.

In Figure 2 (left) we show an example of parameters exhibiting significant difference between the centroids of two groups of patients. Moreover, this difference grows over time. Parameters showing the most noticeable distinction between the two groups of patients are creatinine, neutrophils, glucose, and white blood cells concentrations. The remaining 12 parameters show almost no difference between the patient groups, see for example Figure 2 (right). Time intervals are chosen based on the assumption that the longer ago the test was taken, the less important it is for indication of the current state of the patient.

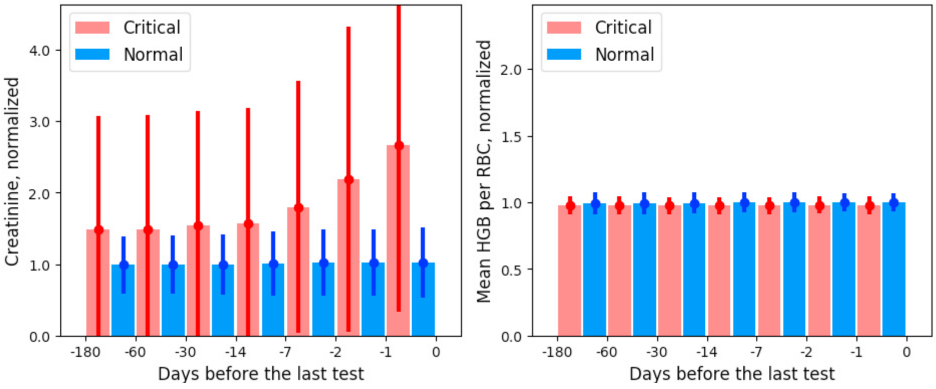


Figure 2. Examples of the mean and standard deviation dynamics. Each bar represents a mean value on the time interval it belongs to, i.e. both bars on any given interval represent a binned value over the same time interval, but for different patient groups

2.2. General experimentations algorithm

First, we perform sensitivity analysis to determine optimal parameters for each method with given data. Then, we assess the performance of the method with full feature set and only with selected features. During the experiments, we determined that the creatinine-neutrophils pair performs the best out of all the parameter combinations. Therefore, selected feature set includes only creatinine and neutrophils for all the results described in the paper. We chose precision, recall, and F1 score as metrics of performance, because of the properties of the data we have. Since only 2.5% of patients were critical, it would be unreasonable to use the fraction of correctly predicted outcomes.

All the experiments described in the following section are performed using Python programming language. All the implemented functions are based on respective implementations from the Scikit-learn library.

2.3. Clustering methods description

2.3.1. Density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm

Density-based spatial clustering of applications with noise (DBSCAN) algorithm is a density-based clustering algorithm[16]. For each point in a given dataset, it determines a set of neighbouring points that are within a specified distance (epsilon). If the number of these core points is higher than a specified threshold, then initial point becomes a new core point (a cluster is started), otherwise the point is marked as an outlier. When a cluster is started, sufficiently big neighbourhoods of each neighbourare added to this cluster. The algorithm keeps working until all the points are either in a cluster or outliers.

The parameters defining the performance of the method are epsilon, minimum samples in a neighbourhood to start (or to be added to) a cluster, and the function to calculate distance between points.

We chose to use Euclidean metric to calculate distance between points. Manhattan distance, albeit having lower computational cost, has shown itself to be less convenient to use because of higher variations of optimal epsilon.

2.3.2. K-means clustering algorithm

K-means [17] is a simple centroid-based clustering algorithm. It requires setting the number of clusters (K) beforehand. After initialising the centres of K clusters (initialization method depends on implementation), the algorithm determines the closest centre for each point and assigns this point to respective cluster. Then, cluster centres get recalculated as centroids of all the points belonging to the respective cluster. The algorithm keeps iterating until cluster centres converge or iteration limit is reached.

The parameters of the method we looked into are the centre initialization method and iteration limit. Available in the Scikit-learn centre initialization options are “random”; a more sophisticated “k-means++” method, which allows to speed up convergence; and an array of manually defined cluster centres.

2.4. Outlier clustering

To perform outlier clustering, we first apply the DBSCAN algorithm with parameter values specified during the sensitivity analysis to detect outliers. Then, we filter out clusters and perform clustering of outliers alone. For this preliminary investigation we use only two factors showing the best clustering performance: creatinine and neutrophils and only the last measurements data, since it shows the highest dispersion.

3. Results of data analysis, sensitivity study and performance evaluation

3.1. Data analysis

Figure 3presents the dynamics of the two parameters that exhibit distinguishably different trends between normal and critical patients as presented in the previous section: creatinine and neutrophils. As it was mentioned in the previous section, all the values are normalized by the median of respective parameter, therefore point (1;1)represents the median values.

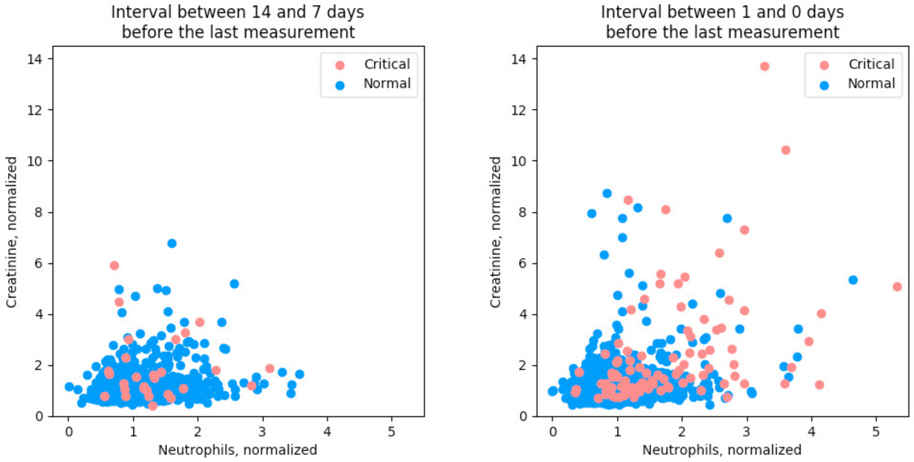


Figure 3. Creatinine and neutrophils dynamics. Normalized to creatinine median (86.0 μmol/L) and neutrophils median (6.2·109particles/L).

As seen from Figure 3, values in the interval between 14 and 7 days before the last measurement are largely contained within relatively small area around medians. On the other hand, the last measurement values tend to spread across larger area, and we see significant number of critical patients located far away from medians.

3.2. Sensitivity study

3.2.1. Sensitivity of DBSCAN clustering algorithm

The parameters we looked into for the DBSCAN algorithm are epsilon and neighbourhood size threshold. We tested epsilon values from 0.05 to 3 with step of 0.05 and neighbourhood size threshold values from 5 to 60 with step of 5. We assess the performance based on the highest F1 score from all the clusters. Out of all the selected parameter combinations, pair of neutrophils and creatinine showed the highest F1 score. The resultsof sensitivity analysis are presented in Figure 4. We can see that the neighbourhood range (*epsilon*) significantly affects the algorithm's performance, while the neighbourhoodsize threshold (*min samples*) causes less variation. For the final experiments, we set the algorithm parameters to the values showing the highest overall performance for each time interval.

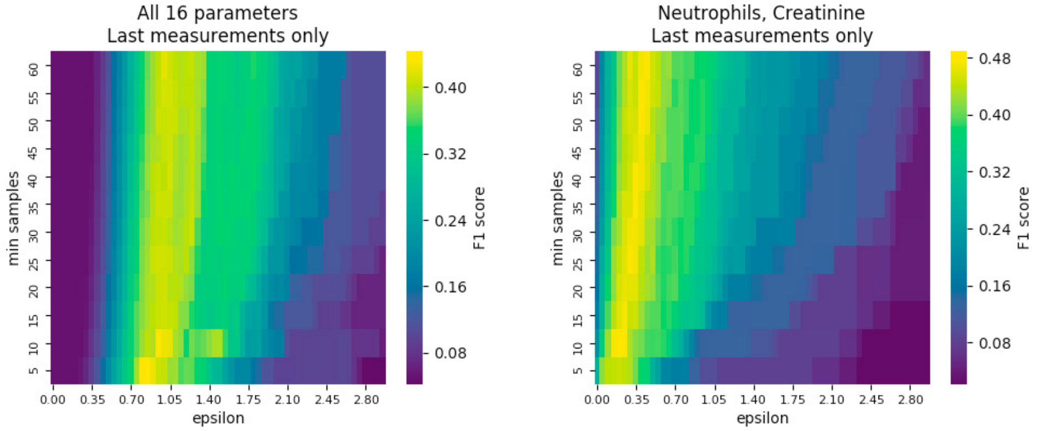


Figure 4. DBSCAN sensitivity analysis results. Colour represents the F1 score achieved with respective algorithm parameters

3.2.2. Sensitivity of K-means clustering algorithm

Sensitivity analysis revealed that when applied to the patients with full feature set with default iteration limit (namely, 300), the algorithm shows poor performance: the highest F1 score the algorithm achieved was 0.177, interestingly enough, at the time interval from two weeks to a week before the last measurement.

It is also worth noting that maximum iterations limitation coupled with random ("k-means++" is still random) centre initialization gives highly inconsistent results and doesn't help to fit the algorithm's parameters.

When applied to the "creatinine – neutrophils" vectors with default iteration limit, the algorithm performs better (F1 score of 0.336 based on the last measurements only) using "k-means++" initialization, but still vastly underperforms by comparison with DBSCAN. As was mentioned earlier, critical patients do not form a cluster, but rather, they spread away from the normal condition, which explains poor performance of clustering algorithms. Furthermore, it is preferred to use k-means clustering when the data form equal-sized (geometrically speaking) clusters, while the clusters we are dealing with are assumed to be vastly different.

To find the optimal centre placement on the "creatinine – neutrophils" plane, we tested each point on the grid from (0;0) to (5;5) with step of 0.1 as the centre for critical cluster, while the centre for normal patients was fixed at (1;1) (medians of both parameters). Figure 5shows the F1 score for different placement of the critical patients' cluster centre.

The iteration limit was set to one during the centre placement analysis. It was revealed that increasing of iteration limit leads to degrading performance. Since optimal centres placement varies depending on specific time interval, we saved optimal centre coordinates for each interval to use them for further performance assessment.

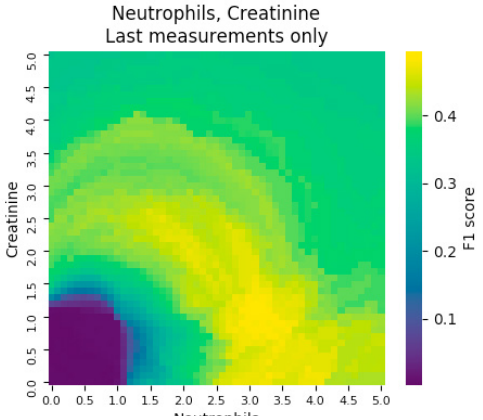


Figure 5. K-means sensitivity analysis. F1 score for different coordinates of the critical cluster centre

3.3. Performance evaluation

Current section includes the resulting precision, recall and F1 score for both methods using parameter values derived in Section 3.2.

3.3.1. DBSCAN

Performance of the DBSCAN algorithm on different time intervals is shown in Figure 6. The overall classification accuracy of the algorithm is rather modest. Even though the F1 score is a little higher when applied to the selected parameters, at the peak of its performance, which is achieved at the last day of treatment, the algorithm is not sufficient to determine the outcome. Another issue that is making this algorithm less usable for real-time medical applications is the need to empirically derive its parameters. It's also worth noting that in each case the best result was achieved when epsilon was high enough to collect the vast majority of points into one cluster, while the outliers included the substantial fraction of critical patients, which supports the conclusion from the previous subsection that critical patients do not form a cluster, but significant part of them scatter away from the norm enough to be considered outliers.

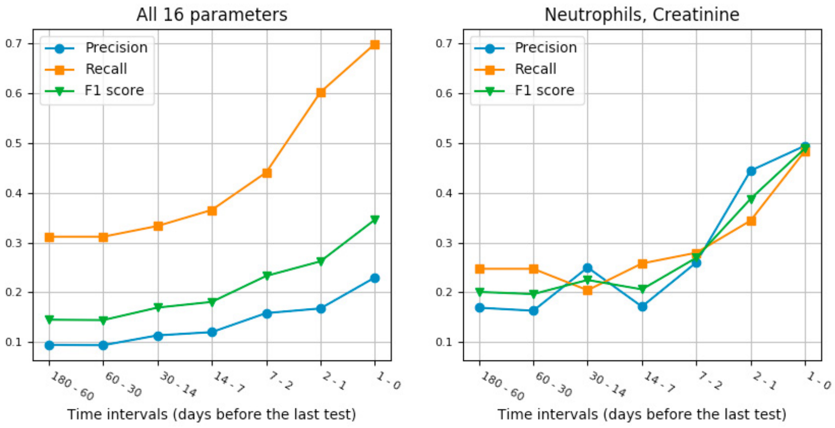
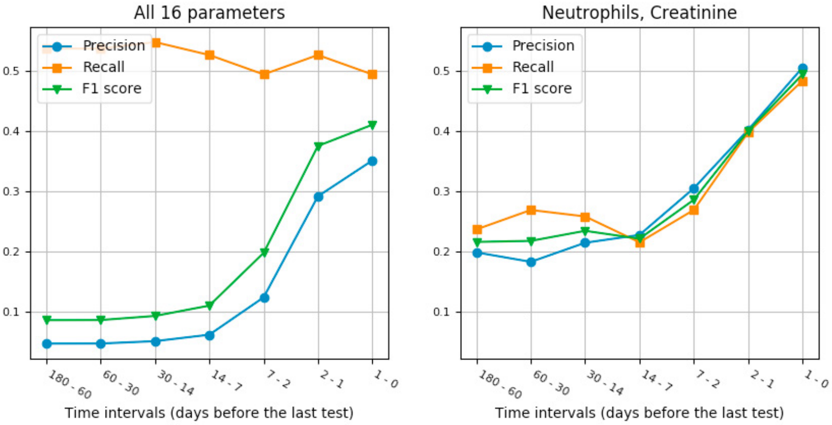


Figure 6. DBSCAN precision, recall and F1 score on different time intervals

Another issue which prevents the DBSCAN algorithm to be a viable option for real-time application is that it does not track cluster centres and cannot simply be trained and used later with new data (although it still might work when applied to a combination of previous and new data). K-means, on the other hand, while being much simpler algorithm, explicitly stores cluster centres and can be used later to determine which cluster a new test vector belongs to.

3.3.2. K-means

K-means performance is shown in Figure 7. When applied to neutrophils – creatinine vectors, the algorithm uses optimal centres placement found in the previous section. When applied to a complete feature set, mean values are used as initial centre locations. Remarkably, the recall is high even at 180 days before the last test when all 16 parameters are used. For the purposes of this research, high recall is more important than high precision, since it is more important to detect the critical transition in most patients (even with false alarms) than to detect only a few of them without any false alarms.


All 16 parameters

Neutrophils, Creatinine

Figure 7. K-means precision, recall and F1 score on different time intervals

During the process of sensitivity analysis, we investigated that the process of iterative refinement of clusters makes the results worse. A single iteration of assigning cluster index by the closest initial cluster centre performs better than the normal way the algorithm operates. Application of the algorithm to selected feature patient vectors is a slight improvement compared to DBSCAN. However, there is a major drawback: in order to perform better, it requires *a priori* knowledge of optimal centres placement.

4. Results of outlier clustering

Here we use only two factors: creatinine and neutrophils and only the last measurements data. The DBSCAN algorithm was applied with parameters epsilon 0.4 and neighbourhood threshold 35 to the patient vectors, containing creatinine and neutrophils values. As a result, the algorithm detects one cluster, containing 3559 patients, and 91 outliers. After that, we perform sensitivity analysis to test the algorithm on outliers only. We start from neighbourhood size of 3 and from epsilon value of 0.5. Results of sensitivity analysis are shown in Table 1.

Table 1. Outlier clustering sensitivity analysis.

	Epsilon: 0.50		Epsilon: 0.60		Epsilon: 0.70		Epsilon: 0.80		Epsilon: 0.90		Epsilon: 1.00	
Min samples	Cluster count	Outlier fraction	Cluster count	Outlier fraction	Cluster count	Outlier fraction	Cluster count	Outlier fraction	Cluster count	Outlier fraction	Cluster count	Outlier fraction
3	4	0.2308	3	0.1758	2	0.1429	2	0.0989	2	0.0879	2	0.0769
4	4	0.3407	2	0.2308	1	0.1868	2	0.1209	2	0.0989	2	0.0769
5	1	0.4945	4	0.2967	1	0.2088	1	0.1978	1	0.1429	2	0.0879
6	2	0.5055	2	0.4835	1	0.2308	1	0.1978	1	0.1758	1	0.1429
7	1	0.6044	1	0.5714	1	0.2967	1	0.2198	1	0.1868	1	0.1538

8	1	0.6154	1	0.5824	2	0.3187	1	0.2198	1	0.1978	1	0.1538
9	1	0.6484	1	0.5824	2	0.4176	1	0.2747	1	0.2198	1	0.1648

Selected examples of resulting clusters are presented in Figure 8. Initial cluster was removed from the picture for better clarity. As seen from the figure and the table, clustering results highly depend on the algorithm's parameters. High epsilon values lead to the situation where all the points belong to one cluster. When neighbourhood threshold is set to 10 and higher, the algorithms detects only one cluster, which grows as epsilon increases. Overall, it is seen that some combinations of parameters allow to detect 3 or 4 clusters using only 2 features out of available feature set, which is a promising result for future research.

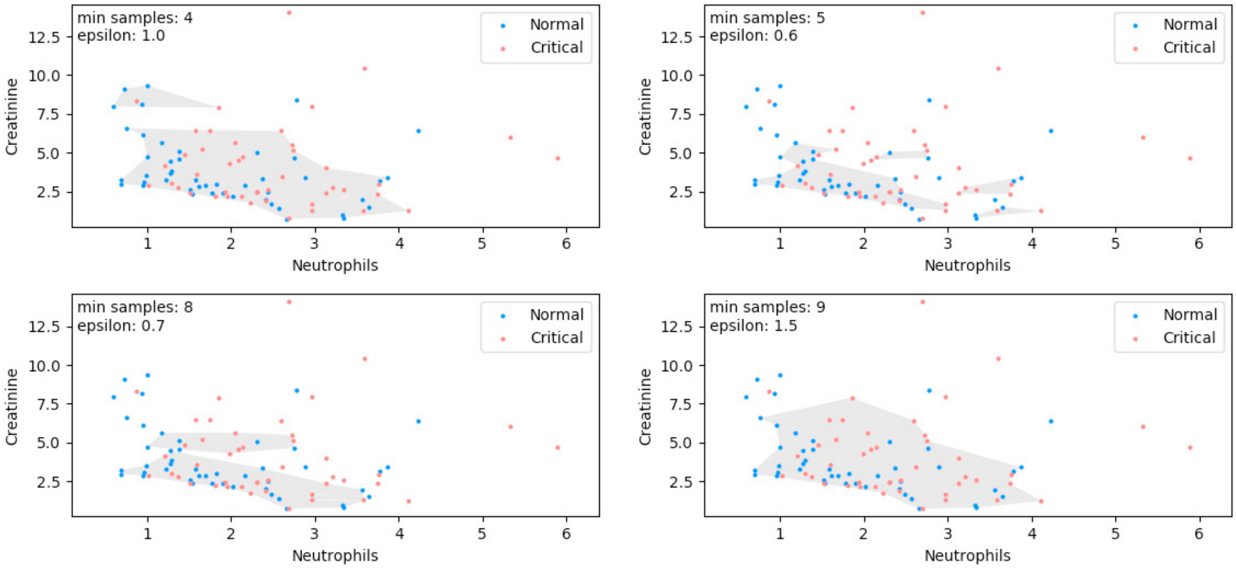


Figure 8. Outlier clustering examples

5. Conclusion and future work

There are important conclusions to be drawn from the research, even though results described in this paper show that investigated methods show a moderate performance when applied to the clinical data, since the data does not form clusters: there is neither clear boundary between two patient groups, nor density-based separation. First, critical patients tend to spread away from normal values and, thus, might be considered outliers by a density based clustering algorithm. Second, since some of the blood parameters show clear differentiation between normal and critical patients, applying just one iteration of k-means algorithm with known normal and critical centres might detect critical patient. In addition, clustering algorithms we tested seem to perform better when applied to a limited set of patient parameters showing the clearest differentiation between the two groups. Outlier clustering is a promising direction for clinical application, and our preliminary analysis shows that the method allows to detect clusters in outliers using just a pair of the best clusteringfeatures: creatinine and neutrophils.

There are a number of prospective directions this research might take. For instance, using more advanced ways to encode temporal features of time series would allow to create more dynamic model to use in real medical setting. Utilizing more advanced unsupervised methods, e.g. OPTICS clustering algorithm or supervised methods, such as random forest, neural network, SVM, or logistic regressionmight improve performance of classification. Our initial experiments with SVM have shown promising results: using high degree (10 and more) polynomial kernel, it has been able to separate the dataset into two groups achieving significantly higher F1 score (0.815) than clustering algorithms (0.5), since it is capable of drawing complex enough classification boundary. For outlier clustering, addition of more detailed information about patients and using more features will allow us to interpret the results in a more meaningful way, e.g. to determine the cause of a specific outlier.

In addition, we plan to investigate similar approaches for early warning of critical transitions in patients with chronic diseases. Unlike the acute coronary syndrome, chronic diseases develop slowly, therefore prediction is likely to be more accurate and timely.

Acknowledgements. This work is supported by the Ministry of Education and Science of the Russian Federation, Agreement #14.575.21.0161 (26/09/2017), Unique Identification RFMEFI57517X0161.We thank Alva Presbitero for important comments and suggestions during the paper preparation.

References

[1] A.E.W. Johnson, T.J. Pollard, L. Shen, L.W.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data. 3 (2016). doi:10.1038/sdata.2016.35.

[2] A. Yakovlev, O. Metsker, S. Kovalchuk, E. Bologova, Prediction of In-Hospital Mortality and Length of Stay in Acute Coronary Syndrome Patients Using Machine-Learning Methods, J. Am. Coll. Cardiol. 71 (2018) A242. doi:10.1016/S0735-1097(18)30783-6.

[3] L.A. Celi, S. Galvin, G. Davidzon, J. Lee, D. Scott, R. Mark, A database-driven decision support system: Customized mortality prediction, J. Pers. Med. 2 (2012) 138–148. doi:10.3390/jpm2040138.

[4] D. Steenhard, Y. Wei, Y. Dong, G. Andrews, V. Gopal, A Prediction Model to Identify Individuals at High Risk for Developing Heart Failure Using Administrative Data and Medical Records, J. Card. Fail. 23 (2017) S77. doi:10.1016/j.cardfail.2017.07.217.

[5] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, P. Szolovits, Unfolding Physiological State: Mortality Modelling in Intensive Care Units., KDD. 2014 (2014) 75–84. doi:10.1145/2623330.2623742.

[6] P. Grnarova, F. Schmidt, S.L. Hyland, C. Eickhoff, Neural Document Embeddings for Intensive Care Patient Mortality Prediction, ArXiv. (2016). http://arxiv.org/abs/1612.00467.

[7] J. Wojtusiak, E. Elashkar, R. Mogharab Nia, C-Lace: Computational Model to Predict 30-Day Post-Hospitalization Mortality, in: Proc. 10th Int. Jt. Conf. Biomed. Eng. Syst. Technol., SCITEPRESS - Science and Technology Publications, 2017: pp. 169–177. doi:10.5220/0006173901690177.

[8] C. Van Walraven, I.A. Dhalla, C. Bell, E. Etchells, I.G. Stiell, K. Zarnke, P.C. Austin, A.J. Forster, Derivation and validation of an index to predict early death or unplanned readmission after discharge from hospital to the community, CMAJ. 182 (2010) 551–557. doi:10.1503/cmaj.091117.

[9] C. van Walraven, J. Wong, A.J. Forster, LACE+ index: Extension of a validated index to predict early death or urgent readmission after hospital discharge using administrative data, Open Med. 6 (2012) 1–11.

[10] A. Presbitero, R. Quax, V. Krzhizhanovskaya, P. Sloot, Anomaly Detection in Clinical Data of Patients Undergoing Heart Surgery, in: Procedia Comput. Sci., 2017: pp. 99–108. doi:10.1016/j.procs.2017.05.002.

[11] J. Lee, D.M. Maslove, J.A. Dubin, Personalized mortality prediction driven by electronic medical data and a patient similarity metric, PLoS One. 10 (2015). doi:10.1371/journal.pone.0127428.

[12] J. Lee, Patient-Specific Predictive Modeling Using Random Forests: An Observational Study for the Critically Ill, JMIR Med. Informatics. 5 (2017) e3. doi:10.2196/medinform.6690.

[13] M. Ghassemi, M.A.F. Pimentel, T. Naumann, T. Brennan, D.A. Clifton, P. Szolovits, M. Feng, A Multivariate Timeseries Modeling Approach to Severity of Illness Assessment and Forecasting in ICU with Sparse, Heterogeneous Clinical Data, Proc. Conf. AAAI Artif. Intell. 2015 (2015) 446–453.

[14] I. Batal, L. Sacchi, R. Bellazzi, M. Hauskrecht, A temporal abstraction framework for classifying clinical temporal data., AMIA Annu. Symp. Proc. 2009 (2009) 29–33.

[15] K. Caballero, R. Akella, Dynamically Modeling Patient's Health State from Electronic Medical Records: A Time Series Approach, in: Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '15, 2015: pp. 69–78. doi:10.1145/2783258.2783289.

[16] J. Sander, Density-Based Clustering, in: Encycl. Mach. Learn. Data Min., Springer US, Boston, MA, 2017: pp. 349–353. doi:10.1007/978-1-4899-7687-1_70.

[17] X. Jin, J. Han, K-Means Clustering, in: Encycl. Mach. Learn. Data Min., Springer US, Boston, MA, 2017: pp. 695–697. doi:10.1007/978-1-4899-7687-1_431.