Using clustering algorithms in medical data analysis*

Michał Tarnawa[1]

[1]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, Poland

Abstract

Many people are suffering from liver diseases, such as Hepatitis, Fibrosis and Cirrhosis. These three illnesses are very dangerous, because they are destroying our organs. To prevent it I decided to create an artificial intelligence (AI) system that will automatically detect them, increasing the chances of survival of patients. I will compare two types of algorithms: KNN, which focuses on categorization, and K-Means, which is used to cluster data, and explain the differences between them. Before we do that, I will analyze the structure of the data and I will point the most vital indicators of each disease, which will be presented on graphs, as well as the correlation between data. It is worth saying that some components of data are connected to only a healthy person, which makes classification much easier. After doing that we will explore the mathematical model of both algorithms, providing necessary math formulas and constants, and compare them according to the pseudocode.After doing this, the last but not least step is a comparison of results tested with different train and test sets. Moreover, parameters of the models will be changed too. Then I will explain the differences between KNN and K-means and their effect on the result of the algorithm. The main goal of this project is to reach 90% accuracy in the diagnosis of these illnesses.

Keywords

KNN, K-Means, PCA

1. Introduction

Nowadays, more and more people are suffering from chronic disease [1]. Particularly important is the analysis and rapid detection of the health condition to minimize the effects. It is worth noting that in today's times, systems based on clustering [2, 3] and machine learning methods are essential [4, 5, 6]. This is due to the possibility of rapid detection and analysis of various human health conditions [7, 8, 9]. Due to our lifestyle, health problems such as hepatitis are becoming more and more common. Hepatitis C, Fibrosis and Cirrhosis are very dangerous illnesses.

The first disease is caused by a virus that is destroying the liver, one of the most important organs in the human body. You can get infected by having contact with infected blood. The second one is the result of the previous problem. Due to the infection, the liver stops functioning properly, causing serious health issues in the body. The third is the last stage of development of this disease. It is absolute destruction of the liver, causing death. Due to this fact, I decided to make a system that will allow us to detect them and speed up the recovery process. The earlier you detect hepatitis, the better your chances of survival.

I chose a datasheet from Kaggle, having over 600 records of patients. We will classify data using two types of algorithms: **KNN** and **K-MEANS** [10, 11].

*IVUS 2025: Information Society and University Studies 2025, May 15, Kaunas, Lithuania

mt311107@student.polsl.pl (M. Tarnawa)

2. Methodology

2.1. K-Nearest Neighbors

The math model of the K Nearest Neighbor [12, 13] algorithm focuses on calculating the distance between the chosen vector and then choosing the k nearest points (with the least distance value between them). Distance can be calculated based on the distance between two points using Euclidean metrics:

$$d = \sqrt{(X_1 - x_1)^2 + (X_2 - x_2)^2 + ... + (X_i - x_i)^2} \tag{1}$$

where: d — distance between two points, X_i - i-th coordinate of point, that is being classified, x_i i-th coordinate of another point from data set.

Also, different metrics can be used as:

$$d = \sqrt[i]{(X_1 - x_1)^j + (X_2 - x_2)^j + \cdots + (X_i - x_i)^j} \tag{2}$$

where: d — distance between two points, X_i - i-th coordinate of the point that is being classified, x_i i-th coordinate of another point from the data set, j - exponent of the metric

After the calculation was done, another step is to find the k-nearest neighbors. If we assume that $a_1, a_2, a_3, \cdot, a_l$ are the numbers of items in every abstraction class of our dataset, then the classification looks like this:

$$classification = max(|a_1|, |a_2|..|ai|) \tag{3}$$

where: $|a_i|$ is number of nearest items in a_i abstraction class.

The classification will return the most suitable class - the one with the most appropriate result.

Algorithm 1 KNN pseudo-code

Data: X_{test}, y_{test}, X_{train}, y_{train}, vector, k
Result: class of object
1 Set metric function (e.g., distance)
2 distances ← empty list
3 **foreach** P *in* X_{train} **do**
4 | Calculate the distance between P and the vector
5 | Append (P, distance) to `distances`
6 **end**
7 Choose k-nearest neighbors from `distances`
8 **return** *the most occurring abstraction class of* k*-nearest neighbors*

2.2. K-Means

K-means [14, 15] is a clustering algorithm that allows us to understand the structure of data. This algorithm divides the dataset in k abstract classes using centroids. What is a centroid? Centroid can be described as a point that is located in the geometrical center of our abstract

class, provided it consists of vectors. In other words centroid is the most "typical" individual of a set. In our case, our centroid will be a "person", who is suffering from a particular disease or not. The main idea of this algorithm is to choose random or selected points and put them into abstract classes. Later, we will classify all of our vectors by inserting them into the most corresponding/nearest classes by calculating the distance between them and the centroids (using formula (1) or (2)). After doing this, we will calculate the new centroid using this formula:

$$X_i = \frac{1}{p} \sum_{i=1}^{j} x_i \tag{4}$$

where X_i - i-th coordinate of new centroid, x_i - i-th coordinate of point from abstract class for each vector in the class. As a result, we will get a new geometric center of our class. We repeat this process until we decide to stop it.

By repeating previous steps it is possible to find accurate visualization of our data structure, however we must remember about the fact, that we are setting the exact number of abstract classes to divide.

Algorithm 2 K-means Clustering Pseudocode

Data: X_{test}, y_{test}, iterations
Result: Centroids and final abstract classes
Choose the number of abstract classes (clusters)
Initialize the first centroids randomly
for $i \leftarrow 0$ **to** *iterations* **do**
    Clear contents of all abstract classes (clusters)
    foreach P *in* X_{test} **do**
        Calculate distances between P and all centroids
        Assign P to the abstract class of the nearest centroid
    end
    Update centroids as the mean of the points in each abstract class
end
return *final centroids and abstract classes*

3. Experiments

3.1. DataFrame and Correlation Matrix

3.1.1. DataFrame Description

The Hepatitis C Prediction Dataset, sourced from Kaggle [16], contains 615 records of blood donors and Hepatitis C patients, with 14 features including demographic and laboratory measurements. The target attribute “Category” consists of five classes: “Blood Donor”, “Suspect Blood Donor”, “Hepatitis”, “Fibrosis”, and “Cirrhosis”. The dataset includes missing values for some laboratory measurements, which were removed during preprocessing; as a result, the

number of records was reduced to 519 entries (which do not contain null values). The columns of the dataframe are as follows:

- **ID**: Unique patient identifier (integer).
- **Category**: Diagnosis of patient (mapped to: “0=Blood Donor”, “0s=Suspect Blood Donor”, “1=Hepatitis”, “2=Fibrosis”, “3=Cirrhosis”).
- **Age**: Patient age in years (integer, 19-77).
- **Sex**: Patient sex (two values: “m” for male, “f” for female).
- **ALB**: Albumin level in g/dL (float, 14.9-82.2).
- **ALP**: Alkaline phosphatase level in U/L (float, 11.3-416.6).
- **ALT**: Alanine aminotransferase level in U/L (float, 0.9-170.3).
- **AST**: Aspartate aminotransferase level in U/L (float, 10.6-324.0).
- **BIL**: Bilirubin level in mg/dL (float, 0.8-189.0).
- **CHE**: Cholinesterase level in U/L (float, 1.4-16.4).
- **CHOL**: Cholesterol level in mg/dL (float, 1.4-9.1).
- **CREA**: Creatinine level in mg/dL (float, 8.0-490.0).
- **GGT**: Gamma-glutamyl transferase level in U/L (float, 4.5-650.9).
- **PROT**: Total protein level in g/dL (float, 44.8-90.0).

The most vital information about the 615 patients is as follows: gender is distributed approximately as 55% male (338 patients) and 45% female (277 patients), consistent with patterns observed in similar studies [1]. Regarding health status, approximately 75% of patients (461) are classified as healthy (“Blood Donor” or “Suspect Blood Donor”), while 25% (154) are diagnosed with Hepatitis C-related conditions (“Hepatitis”, “Fibrosis”, or “Cirrhosis”).

3.1.2. Correlation Matrix

Now we are going to analyze the correlation between the data.

The correlation matrix in Figure 1 illustrates the relationships between the dataset features, highlighting strong correlations (e.g., between age and category) that inform the clustering process.

As we can see, the biggest correlation is between category and AST, so we can assume that this position will be vital for our classification. Moreover, the second biggest correlation can be observed between ALB and Proteins(0.57), which means that these two organic compounds are likely connected to the same organ. It is worth noticing the big influence of BIL and the category. Correlation indices above 0.40 could be observed in a few cases: GGT and category, GGT and AST, and CHOL and CHEL. The last one should not be surprising.

The exact zero correlation is between CREA and ALB, leaving them an important factor of our diagnosis, because they are not dependent on each other. Such a low correlation can be seen as well between category and ALP, category and PROT. Knowing the fact, that our database is dominated by healthy ones, and there is big disproportion in the distribution of protein levels among them, we can assume that this will be a key element for diagnosis of various illnesses.

Low correlation level could be seen between; CREA and AST, CHOL and GGT, BILL and ALP, ALT and AGE and many, many more data.

To summarize, we extracted vital correlation information from the data, which allows us to know more about our dataset. What is more, we can assume the most important data columns for our categorization. Nevertheless, the biggest level on dependency between data is 0.65, which is too low to reduce the number of dimensions in our dataset. If we do it, there will be a huge chance of decreasing the efficiency of our algorithm.

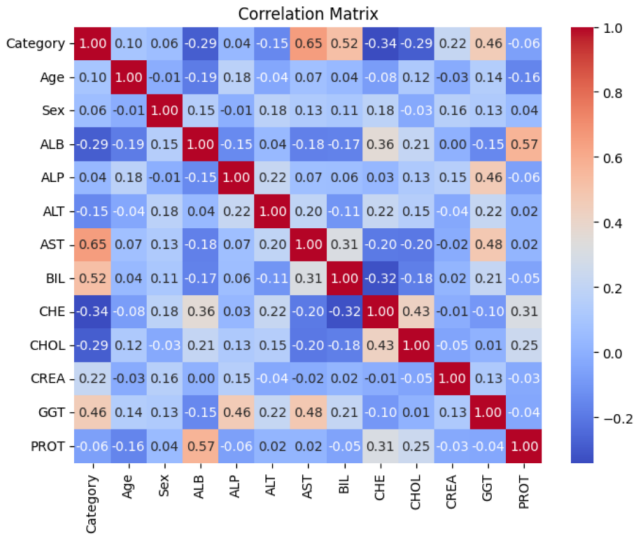


Figure 1: Correlation matrix of medical dataset features.

3.2. Data Distribution

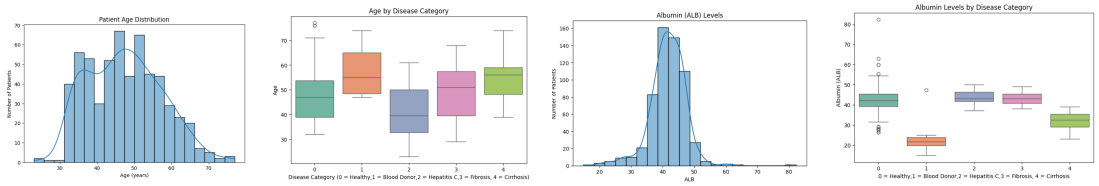
In this section, I am going to analyze the structure of the dataset [16], comparing the distribution of all records, as well as their impact on the final classification. This will allow us to understand our topic more clearly. Moreover, this will provide knowledge about the most important indicators of hepatitis and those that may be the least important.

After that, I will be able to draw a conclusion about how neighbors in the KNN algorithm (k-nearest neighbors) are located, as well as where the largest cluster of “Blood Donor”, “Suspect Blood Donor”, “Hepatitis”, “Fibrosis”, and “Cirrhosis” patients is situated.

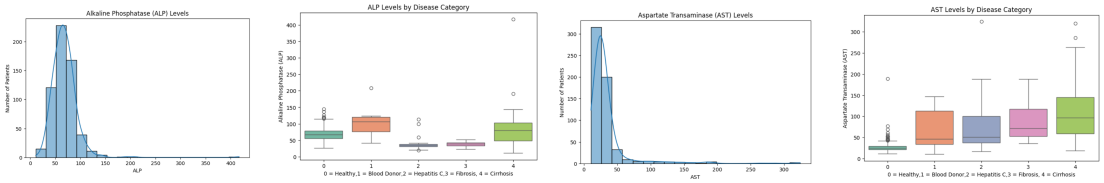
3.2.1. Age, ALB, ALP, AST figures

Conclusion: As we can see (Fig. 2a, Fig. 2b) most of the patients are between 30 and 60 years old, as a result our algorithm will be the most efficient for them. Moreover it can be observed that people with Hepatitis and Cirrhosis are older than healthy and other ones.

Analyzing ALB level (Fig. 2c, Fig. 2d), we can see that it has disturbed Gauss distribution and vary between 30-50 g/dL. Subjects 0, 2 and 3 have the biggest levels of ALB, nevertheless subjects 1 have the lowest. For hepatitis it is lower than average.



(a) Distribution of patient age. (b) Box plot of age by patient group. (c) Distribution of ALB levels. (d) Box plot of ALB by patient group.



(e) Distribution of ALP levels. (f) Box plot of ALP by patient group. (g) Distribution of AST levels. (h) Box plot of AST by patient group.

Figure 2: Distributions and box plots of Age, ALB, ALP, and AST in the medical dataset.

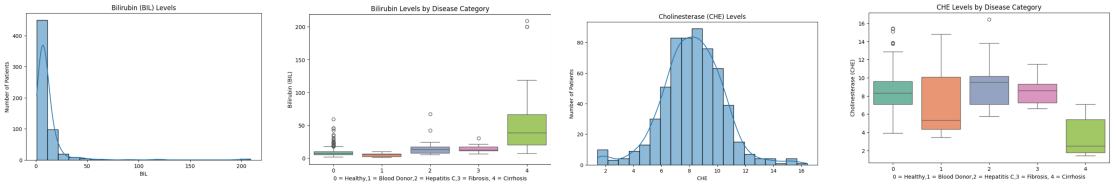
Analyzing ALP level (Fig. 2e, Fig. 2f), we can see that it has disturbed Gauss distribution with max spiking up to 400 U/L. We can assume that this spike is created by seriously ill patients. Subjects 2 and 3 have the lowest level, on the other hand subjects 4 have the biggest disproportions in value. AST levels (Fig. 2g, Fig. 2h) are between 50 and 100 U/L for most of the patients, but it rockets up to approximately 350 U/L. For 0 and 1 mean of AST level is the lowest, however 2,3 and 4 have the highest. Hepatitis and Cirrhosis have the biggest variety in value.

3.2.2. BIL, CHE, CHOL, CREATINE figures

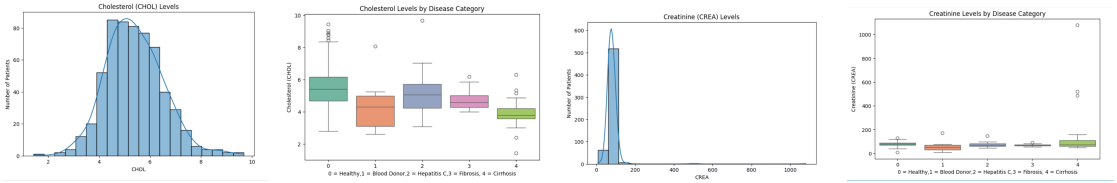
Conclusion: For bilirubin (Fig. 3a, Fig. 3b) we can see that, most of patients have zero or very low level of it, however it is spiking up to 200 mg/dL. Moreover for subject 4 we can see the highest concentrations of it, with the biggest variety. On the one hand rest of the categories have lower level of it, but on the other hand 0 and 2 have huge disproportion in values.

Speaking about cholinesterase (Fig. 3c, Fig. 3d), which distribution is similar to Gauss normal distribution. Most of the patients have CHE level between 4 and 11 U/L, with 9 U/L being the most common value. For each category we can observe big disproportion in data. For 1-4 values are diverse.

Looking at cholesterol (Fig. 3e, Fig. 3f), it's distribution is similar to Gauss normal distribution. Most of the patients are located between 4 and 7 mg/dL. The most popular value is around 5-6 mg/dL. Healthy ones have the biggest range of values, although having it's mean almost perfectly in the middle of the box. Creatinine levels (Fig. 3g, Fig. 3h) are between 0 to 150 mg/dL for most of the people, rocketing up to over 490 mg/dL. We can assume that this spike in describing ill persons. The level of creatinine between 5 abstract classes is similar, with only number 4 deviate from rest. Big disproportion could be seen in 0 as well.



(a) Distribution of bilirubin levels. (b) Box plot of bilirubin by patient group. (c) Distribution of cholinesterase levels. (d) Box plot of cholinesterase.



(e) Distribution of cholesterol levels. (f) Box plot of cholesterol by patient group. (g) Distribution of creatinine levels. (h) Box plot of creatinine by patient group.

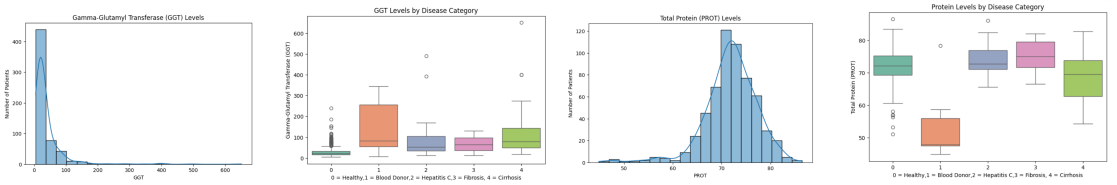
Figure 3: Distributions and box plots of BIL, CHE, CHOL, and CREA in the medical dataset.

3.2.3. GGT, PROTEINS figure

Looking at Figure 4: Speaking about GGT (Fig. 4a, Fig. 4b), we can see that most of people have low level of it in blood, below 100 U/L, spiking up to 650 U/L.

Subjects 1 have the biggest disproportion in range of value, as well as in it's mean. We can assume that GGT is identifying blood donors. Such a big data variety could be seen as well in 4.

Analyzing protein levels (Fig. 4c, Fig. 4d) it can be stated that most of patients have average level between 65 and 80 g/dL. Blood donors have the lowest level of it and mean for them is very low, almost unnoticeable. It is worth to say, that the biggest disproportion of value is in both examples of cirrhosis and healthy.



(a) Distribution of GGT levels. (b) Box plot of GGT by patient group. (c) Distribution of total protein levels. (d) Box plot of total proteins.

Figure 4: Distributions and box plots of GGT and PROT in the medical dataset.

3.3. Data set division

As mention previously hepatitis data sheet consist of 615 record, however only 519 are being considered, due to the fact, that rest consist of null values. The set was divided into 4 others

sets: X_{train}, y_{train}, X_{test}, y_{test} using train split test from sklearn library. Training sets consist of 0.7 records of the original data, remaining 0.3 belongs to test.

3.4. KNN - test

KNN algorithm has been tested using 4 different k values: k=2, k=4, k=8, k=16 and 3 samples of data. The results can be seen bellow:

Table 1
Classification accuracy results of KNN algorithm for hepatitis detection using different k values across three random seed configurations on the hepatitis dataset.

k	Accuracy (%), Seed 43	Accuracy (%), Seed 40	Accuracy (%), Seed 7
2	92.31	91.07	93.22
4	89.11	90.29	92.66
8	88.70	89.23	92.06
16	89.23	89.23	92.06

As we can see in Table 1, my algorithm achieves an accuracy of around 88-93.5%, which is very good information for us. However, there is a paradox: a low number of neighbors yields the best result. Our intuition suggests that the lowest number of neighbors should provide the lowest accuracy. This occurs because of the characteristics of the dataset.

3.5. K-Means test

K-Means algorithm was tested using the same data set from knn. I chose 5 abstract class and 5 random points from data frame as a main centroids. Each point belongs to different abstract class. After testing:

Table 2
Accuracy of K-Means clustering for different numbers of iterations with randomly chosen centroids for each abstract class.

# of Iterations	Accuracy (%)
1000	11.24
2000	11.29
10000	11.29

As we can see in Table 2, the accuracy is extremely low. I suspect that selecting random points from the dataset was not a good idea, so I decided to change the method of choosing centroids. I selected every point from each category, divided them into training and test sets, and calculated their centroids. This process has been described in the previous paragraphs. After the new calculations:

Taking a look at Table 3, we can see that the accuracy remains almost the same no matter how many iterations we will do. Excluding one exception (seed 7), our accuracy ranges between 75.42% and 88.83%. We also reached 4.46% accuracy; however, this is an anomaly connected

Table 3
Accuracy of K-Means clustering for different numbers of iterations and random seeds, using train and test sets for every abstract class.

# of Iterations	Accuracy (%), Seed 40	Accuracy (%), Seed 7	Accuracy (%), Seed 43
100	75.41	4.46	88.82
1000	75.41	4.46	88.83
2500	75.42	4.46	88.83
10000	75.42	4.46	88.83

to the way of choosing points. As mentioned previously, our dataset is dominated by healthy individuals, and due to this fact, the splitting method did not work correctly. The solution to this problem is to create a new division.

4. Conclusion

To summarize we see that KNN is much more effective than K-Means, but why? Well first of all KNN is calculating distance between every point allowing to find the nearest solution. It has a very good accuracy between 84% and 93%. On the other hand k-means precision is between 74% and 88% excluding anomalies. The key difference can be found in purpose of the algorithm.

KNN is used to categorize the data, while K-Means is clustering it. That means the first one is trying to find the best solution, while other is looking for suitable abstract class for a point. For instance: if one vector must be put into one of two abstract classes, while we know that it fits to another 3rd undeclared class, it's result will be less appropriate. It is possible, that in our case instead of 5 abstract class there should be more for example:

- hepatitis and heart disease
- fibrosis and diabetes
- and more

And this is the main reason of lower accuracy compared to knn.

In the end we can say that both methods are accurate, however knn is more, due to the fact that this data set is perfect for it. Big disproportions in certain blood elements between ill and healthy persons are the best thing that could happen for this type of algorithm(but not for patients).

Declaration on Generative AI

During the preparation of this work, the author used Grammarly in order to: Grammar and spelling check. Further, the author used Overleaf built-in editor features in order to: LaTeX syntax checking and document formatting assistance. After using these tools, the author reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Akif, M. S. Qusar, M. R. Islam, The impact of chronic diseases on mental health: an overview and recommendations for care programs, Current Psychiatry Reports 26 (2024) 394–404.

[2] Y. Ding, L. Li, W. Wang, Y. Yang, Clustering propagation for universal medical image segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 3357–3369.

[3] Z. Liu, Fermatean fuzzy similarity measures based on tanimoto and sørensen coefficients with applications to pattern classification, medical diagnosis and clustering analysis, Engineering Applications of Artificial Intelligence 132 (2024) 107878.

[4] K. Prokop, Feature fusion using deep learning modules and fuzzy c-means for medical disease recognition, in: 2025 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2025, pp. 1–6.

[5] P. Żerdziński, Fantastic fishes and how to classify them, in: International Conference on Information and Software Technologies, Springer, 2024, pp. 26–37.

[6] D. Połap, M. Woźniak, Bacteria shape classification by the use of region covariance and convolutional neural network, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–7.

[7] M. Wieczorek, J. Siłka, K. Wiltos, M. Woźniak, Transformer based semantic segmentation network for medical imaging application, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2024, pp. 380–389.

[8] J. Zhang, Z. Luan, L. Ni, L. Qi, X. Gong, Msdanet: A multi-scale dilation attention network for medical image segmentation, Biomedical Signal Processing and Control 90 (2024) 105889.

[9] D. Połap, A. Jaszcz, Decentralized medical image classification system using dual-input cnn enhanced by spatial attention and heuristic support, Expert Systems with Applications 253 (2024) 124343.

[10] H. Lubis, I. Lubis, H. Harahap, T. Tommy, R. Siregar, Integration of probabilistic multi-class labeling and adaptive k-means clustering with knn classification: Application to weather data, Journal of Computer Science, Information Technology and Telecommunication Engineering 5 (2024) 615–627.

[11] D. C. Veeraiah, C. Anuradha, P. S. Nithin, P. A. Kumari, College suggesta: Enhancing choices with k-means, knn, and cosine similarity using flutter and django, in: 2024 International Conference on Signal Processing and Advance Research in Computing (SPARC), volume 1, IEEE, 2024, pp. 1–8.

[12] A. Jaszcz, Reducing the number of calculations in k-nn by class representatives atb voting., in: SYSTEM, 2021, pp. 26–31.

[13] K. Prokop, Grey wolf optimizer combined with k-nn algorithm for clustering problem., in: IVUS, 2022, pp. 14–19.

[14] B. Chong, et al., K-means clustering algorithm: a brief review, Academic Journal of Computing & Information Science 4 (2021) 37–40.

[15] H. Hu, J. Liu, X. Zhang, M. Fang, An effective and adaptable k-means algorithm for big data cluster analysis, Pattern Recognition 139 (2023) 109404.

[16] F. Soriano, Hepatitis c prediction dataset, https://www.kaggle.com/datasets/fedesoriano/hepatitis-c-dataset, 2020.