Features selection throught autoencoder filtering and DeepShap: an iterative algorithm

Edoardo De Rose[1,*], Carlo Adornetto[1], Francesco Calimeri[1] and Gianluigi Greco[1]

[1]*Department of Mathematics and Computer Science, University of Calabria, Via Pietro Bucci, Rende, Italy*

Abstract

In many fields, such as functional genomics or finance, data analysis, and predictive modeling are always challenging for the course of dimensionality and noisy data. In these cases, effective feature selection algorithms, based on Machine and Deep Learning, can perform and improve the identification of important features, leading to more treatable problems in terms of dimensionality. The paper proposes a novel algorithm to perform Feature Selection on highly dimensional data, which exploits the reconstruction capabilities of autoencoders and an ad-hoc defined Explainable Artificial Intelligence-based score to select the most informative feature for predictions. We benchmark such an approach on several state-of-the-art datasets and against the previously proposed algorithm in the literature, showcasing its effectiveness.

Keywords

Deep Learning, Explainable AI, Genomics

1. Introduction

In the field *functional genomics*, starting from the results of the Human Genome Project, the evolution of sequencing techniques provides big volumes of data for each single patient by taking advantage of the *high-throughput* and *next-generation sequencing* i.e., a set of time and cost-effective techniques for sequencing DNA and RNA. By means of them, it is possible to measure the expression of thousands of genes for each individual and hence to collect quantitative gene expression profiles (GEP) to be used for research and clinical purposes. But despite GEP datasets represent a valuable source of information in healthcare—they are indeed used for diagnosis, prevention, and precision medicine—their analysis results challenging for three main reasons. The first one is the *course of dimensionality*: a genomics dataset typically consists of a very large number of features (genes) and a small number of samples (patients); the second problem concerns *imbalanced classes*: in the analysis of different groups of patients, genomics data are often stratified in classes according to different pathologies. In most cases, there is a significant difference between the number of instances in each class; finally, sequencing data are typically collected from multiple sources, different laboratories, and sequencing tools. This results in *noisy datasets* which are difficult to analyze [1].
In recent years, Machine Learning (ML) and Deep Learning (DL) have been widely adopted in this field, providing breakthrough results and meaningful insights into the relationship between genomics and cancer [2, 3]. Although still very promising, DL models are in general not immediately interpretable, meaning that it is difficult to understand the causal relationship between the inputs and their outcomes. This is an even more severe problem in the bioinformatics domain, where it is crucial to understand, for example, in the case of genomics, how the expression of a gene can affect the progression of oncological patients.

We propose a new algorithm, based on DL and Explainable Artificial Intelligence (XAI), for genomics whose aim is threefold: first, select the most meaningful genes for a regression/classification problem;

3rd AIxIA Workshop on Artificial Intelligence For Healthcare (HC@AIxIA 2024), 25-28 November 2024, Bolzano, Italy

[*] Corresponding author.

edoardo.derose@unical.it (E. D. Rose); carlo.adornetto@unical.it (C. Adornetto); francesco.calimeri@unical.it (F. Calimeri); gianluigi.greco@unical.it (G. Greco)

0000-0002-0032-9434 (E. D. Rose); 0000-0002-9734-1017 (C. Adornetto); 0000-0002-0866-0834 (F. Calimeri); 0000-0002-5799-6828 (G. Greco)

second, provide a more accurate prediction model; third, quantify and evaluate the effect of features on the predictions, through XAI. We used our algorithm for the GEP analysis of acute lymphoblastic leukemia (ALL) patients, identifying a meaningful subset of genes for the disease prognosis. The following sections are organized as follows. First, we review the most relevant related works in Section 2, and we then give a formal definition of the algorithm in Section 3. The application and the results obtained by the algorithm for the CLL study are discussed in Section 4. Finally, directions for further research are proposed in Section 5.

2. Related Works

A number of recent studies propose and evaluate new approaches for feature selection (FS) on GEP datasets for cancer diagnosis and prognosis[2]. Such methodologies mainly aim at selecting the most informative genes, which are able to characterize classes and identify groups of patients. In this context, the adoption of XAI methods has started to gain momentum for interpretability purposes as well as to enhance FS[4, 5, 6]. A widely used approach to overcome the *course of dimensionality* problem is to perform dimensionality reduction using AEs [7]. While this has been proven to be effective, the encoding is typically a non-linear projection of the variables into a lower-dimensional space, which makes it difficult to provide the proper interpretations of the results. In this work we propose a novel approach, which uses AEs for selecting the most informative genes without any change into the original features space, hence enhancing the explainability of the results, and still exploiting the representation abilities of AEs.

We moreover use an ad-hoc defined XAI-based score in order to iteratively select the features by taking advantage of the Shapely Additive ex-Planation method (SHAP)[8], a cooperative game theory-based approach for computing the *shapely values.* Such values measure, locally (at the sample level), the contribution of each feature to the predictions of an ML model. In particular, for a given sample x, the set of features F, the contribution of the feature $j \in F$ is defined as:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{j\}}(x_{S \cup \{j\}}) - f_S(x_S) \right] \tag{1}$$

with $\phi_j \in \mathbb{R}$ and where $f_{S \cup \{j\}}$ and $x_{S \cup \{j\}}$ denote the prediction model and the sample considering the only subset of features S without the j-th one. In words, SHAP computes the contribution of a feature by comparing the model predictions obtained with and without a feature, for all the possible combinations S. Since the computation of the Equation 1 is inefficient in the case of NN as a prediction model—a NN should be re-trained for each combination of features ($2^{|F|}$)—the authors demonstrate in [8] that shapely values can be computed by solving a weighted linear least square regression with the proper shapely kernel. Although we used such an alternative method, we omitted the details and focus on the only definition of shapely values.

3. The Algorithm

The proposed algorithm is based on two main ideas: (1) we use a clustered correlation matrix in order to group features that enclose similar patterns and we then filter the redundant information for each group by using AEs. In contrast with previous works, in which AEs are used for dimensionality reduction, we still work at the level of the original features. In particular, we take advantage of the encoding and reconstruction abilities of the AEs assuming that the more accurate is the reconstruction of a feature, the more that feature is representative of the cluster it belongs to. We hence provide a more treatable dataset in terms of dimensionality, without loss of representativeness, by filtering redundant features; (2) we train NNs and we iteratively select the most meaningful features using a new ad-hoc defined SHAP score. We repeat the analysis by removing at each iteration, the previously selected features. We

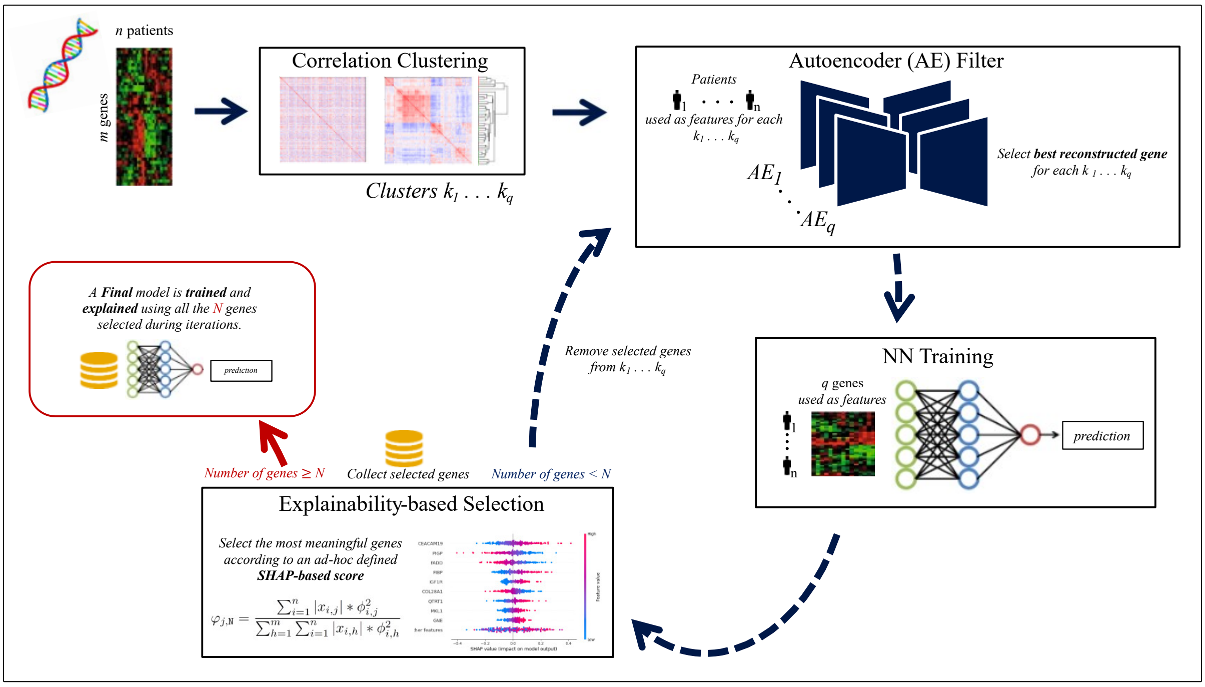

n patients
m genes
Correlation Clustering
Clusters $k_1 \dots k_q$
Autoencoder (AE) Filter
Patients
1 . . . n
used as features for each $k_1 \dots k_q$
AE_1
AE_q
Select best reconstructed gene for each $k_1 \dots k_q$
NN Training
q genes used as features
1
n
prediction
Remove selected genes from $k_1 \dots k_q$
A Final model is trained and explained using all the N genes selected during iterations.
prediction
Number of genes ≥ N
Collect selected genes
Number of genes < N
Explainability-based Selection
Select the most meaningful genes according to an ad-hoc defined SHAP-based score
$\varphi_{j,\mathtt{N}} = \frac{\sum_{i=1}^{n} |x_{i,j}| * \phi_{i,j}^2}{\sum_{h=1}^{m}\sum_{i=1}^{n} |x_{i,h}| * \phi_{i,h}^2}$
CEACAM19
PIGP
FADD
FIBP
IGF1R
COL28A1
QTRT1
MKL1
GNE
her features
−0.4
−0.2
0.0
0.2
0.4
SHAP value (impact on model output)
High
Feature value
Low

Figure 1: The Proposed Algorithm.

eventually use the set of selected features (from all the iterations) to train and explain a final model. Figure 1 shows the main algorithm phases.

3.1. Formal Setting

Let be $\mathcal{D} = \{X, Y\}$ a dataset such that $X \in \mathbb{R}^{n \times m}$ is the matrix of inputs, and $Y \in \mathbb{R}^{n \times l}$ is the matrix of the corresponding labels. Let us further assume $m \gg n$ meaning that the dataset is characterized by a way larger set of features with respect to the number of samples.

As a novelty contribution, we introduce a new impact score, which, by means of the SHAP local explanation, measures the global impact of each feature on model predictions. We hence associate to each feature (column) j of X, used to train a model $\mathtt{N}$, a couple $(\rho_{j,\mathtt{N}}\ , \varphi_{j,\mathtt{N}})$ were $\rho_{j,\mathtt{N}}$ is the correlation between the j-th columns of X and their shapely values $\{\phi_{1,j}, ..., \phi_{n,j}\}$, and $\varphi_{j,\mathtt{N}}$ is defined as follows:

$$\varphi_{j,\mathtt{N}} = \frac{\sum_{i=1}^{n} |x_{i,j}| * \phi_{i,j}^2}{\sum_{h=1}^{m}\sum_{i=1}^{n} |x_{i,h}| * \phi_{i,h}^2} \tag{2}$$

With $\rho_{j,\mathtt{N}}$ and $\varphi_{j,\mathtt{N}}$ we want to emphasize *how* and *how much*, respectively, a feature globally affect the predictions of $\mathtt{N}$.

3.2. Algorithm

For sake of clarity, we introduce our algorithm by first defining a set of sub-procedures. The first one (Algorithm 1) computes the pairwise correlation matrix $C \in \mathbb{R}^{m \times m}$ between the features (columns) of a generic real-valued matrix M. Finally it clusters C in order to return a set $K = \{k_1, ..., k_q\}$ such that for each $i = 1, .., q$, k_i is a set of indexes—a partition (cluster) for the columns of M.
The second sub-procedure, defined in Algorithm 2, trains an AE for each cluster, by using the transpose of the input matrix M—meaning that, for the AE model, each feature represents a sample and vice versa. The rationale here is that we assume the best-reconstructed feature (over the samples) to be the most representative of the cluster it belongs to. We denote $M_{k_i} \in \mathbb{R}^{n \times |k_i|}$ as a matrix including the only columns of M which indexes are in k_i. The $evaluate$ function provides the column indexes

of M_k^T associated with the best-reconstructed feature. Finally, the sub-procedure returns a set J of q indexes—one for each cluster.

Algorithm 1 Corr. & Clustering

```
function CorrClustering($M$)
    $C \leftarrow corr(M)$
    $K \leftarrow clustering(C)$
    return $K$
end function
```

Algorithm 2 AE Filtering

```
function AEFiltering($M$,$K$)
    $J \leftarrow \emptyset$
    for $k \in K$ do
        $\texttt{AE} \leftarrow train(M_k^T)$
        $J \leftarrow J \cup evaluate(\texttt{AE}\ , M_k^T)$
    end for
return $J$
end function
```

The last sub-procedure, reported in Algorithm 3, takes as input: the data, a matrix of shapely values Φ and the threshold $\beta \in \mathbb{R}$, with $\beta \in [0, 1]$. It first computes the correlations between each column of M and the corresponding columns of Φ. Subsequently, it computes the intensity for each feature following the definition of equation 2. It then selects the column indexes according to β and the mean intensity, to finally provide a set $\tilde{J}$ of column indexes for M.

Algorithm 3 Selection

```
function select($\Phi, M, \beta$)
    $\boldsymbol{c} \leftarrow computeCorrelation(\Phi, M)$
    $\boldsymbol{d} \leftarrow computeIntensity(\Phi, M)$
    $\mu \leftarrow \frac{1}{|\boldsymbol{d}|} \sum_{d \in \boldsymbol{d}} d$
    $\tilde{J} \leftarrow \{j \mid |\rho_j| > \beta \ \wedge \ \varphi_j > \mu, \ \forall \rho_j \in \boldsymbol{c}, \forall \varphi_j \in \boldsymbol{d}\}$
    return $\tilde{J}$
end function
```

The main procedure is described by Algorithm 4. After clustering the correlation matrix, it selects a set of meaningful features index to be added to S. It then removes the selected indexes from their corresponding clusters in K and proceeds by repeating the analysis. Here we denote $X_J \in \mathbb{R}^{n \times |J|}$ (and accordingly X_S) as a matrix including the columns of X which indexes are in J, and $\texttt{N}_J$ (and accordingly $\texttt{N}_S$) as a NN trained on $\{X_J, Y\}$. The iterative analysis stops when $\alpha \in \mathbb{N}$, $\alpha \leq m$ features are selected or on a maximum number of iterations. The algorithm eventually trains and explains a final NN using the set S.

Algorithm 4

```
Require: $X, Y$
    $K \leftarrow$ CorrClustering($X$)
    $S \leftarrow \emptyset$
    while $|S| < \alpha \vee$ not $maxIter$ do
        $J \leftarrow$AEFiltering($X, K$)
        $X_J^b, Y^b \leftarrow$dataBalancing($X_J, Y$)
        $\texttt{N}_J \leftarrow$ findModel($X_J^b, Y^b$)                ▷ Model Selection & Training
        $\Phi \leftarrow$ Shap($\texttt{N}_J, X_J$)                      ▷ matrix of shapely values $\Phi \in \mathbb{R}^{n \times |J|}$
        $\tilde{J} \leftarrow$ select($\Phi, X_J$)
        $S \leftarrow S \ \cup \ \{j_i \in J \mid i \in \tilde{J}\}$
        $K \leftarrow K \setminus S$                                   ▷ remove from their corresponding cluster
    end while
    $\texttt{N}_S \leftarrow$ findModel($X_S, Y$)
    $\Phi \leftarrow$ Shap($\texttt{N}_S, X_S$)
    $J^* \leftarrow$ select($\Phi, X_S$)
```

4. A Use Case: Leukemia-ALLAML

4.1. Materials and Methods

To validate the effectiveness of the method, we tested it on a synthetic toy dataset. This allowed us to verify that the method correctly selected the centroid features for each cluster, ensuring that the

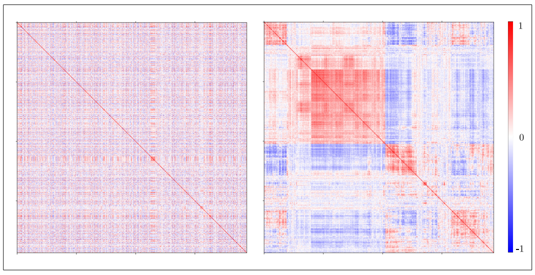


Figure 2: Genes clustered correlation matrix.

Iteration	Accuracy (CI 95%)
1	97.0%-99.0%
2	95.0%-97.0%
3	92.0%-95.0%
4	92.0%-95.0%
final	**99.0%-100.0%**

Table 1: Results over iterations.

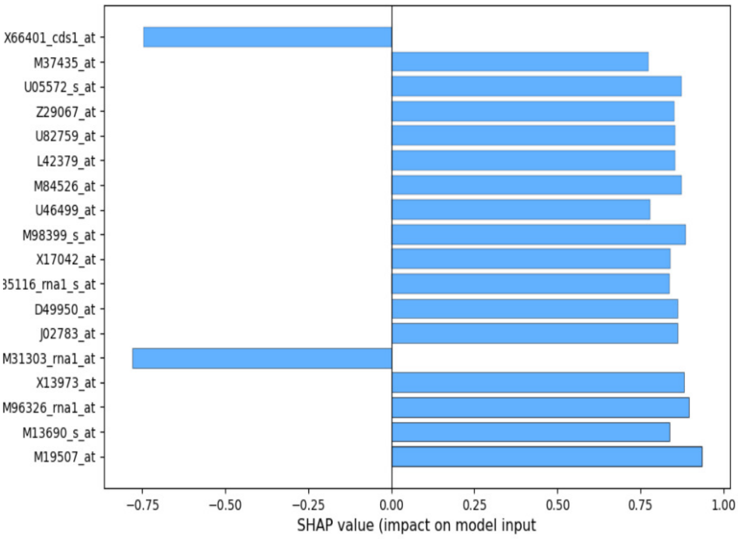
most representative features were identified. We applied our algorithm for analyzing GEP of patients from Leukemia from the Kent Ridge biomedical data repository [9]. The leukemia dataset consists of two classes of acute leukemia known as acute lymphoblastic leukemia (ALL), arising under lymphoid precursors, and acute myeloid leukemia (AML), arising under myeloid precursors. There are 72 bone marrow samples in the dataset with 47 ALL and 25 AML cases and each contain 7129 gene probes. We used the proposed algorithm for training a NN to solve such a classification problem as well as to identify a set of meaningful genes over the whole set of 7129. We additionally provide insight into the prognostic power of such genes. The genes were initially clustered in groups based on their feature correlations. First, we computed the correlation matrix of the features, capturing the pairwise correlations between genes. We then applied a correlation threshold to define significant relationships between features. Specifically, if the absolute value of the correlation between two genes exceeded a predefined threshold, we considered them to be correlated. We then identified clusters of correlated genes by detecting the connected components. Each connected component represents a group of genes that are strongly correlated with each other. This method allowed us to group the genes into distinct clusters, capturing the structure of the data without relying on predefined assumptions about the number of clusters. These clusters were then used for further analysis and filtering. The AE filtering selects genes and we further applied a statistical filter in order to select 50 genes. After re-balancing the classes with the Synthetic Minority Over-sampling Technique (SMOTE), we perform model selection with 10-fold cross-validation in order to find the best (in terms of binary accuracy on the test set) NN for solving the classification problem.

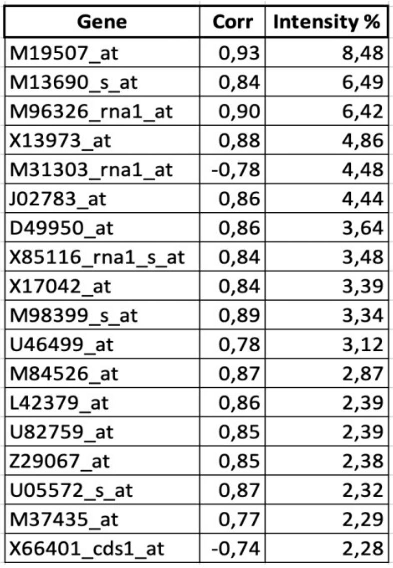
We finally use our SHAP scores (defined in Section 3.1) to select the most meaningful genes, by setting $\beta = 0.85$. After selecting a set of $\alpha = 50$ genes through the iterations of the algorithm, we use them to train and explain a final NN.

The algorithm has been implemented using the Python (v3.8.11) programming language. NNs have been implemented by taking advantage of the Pytorch (v2.4.1) framework. XAI analysis was performed by means of the SHAP library [8].

4.2. Results

The overall results are reported in Table 1. In particular, for each iteration of the algorithm, we measured the accuracy of all the models obtained during cross-validation, for which we report the confidence interval. As we expected, the classification accuracy decreases with the algorithm iterations: the reason is that the previously chosen features—expected to be the most representative of each cluster—are no more considered for the subsequent analysis. An improvement in accuracy is instead reported for the final step of the algorithm, by which a model is trained using the set of genes selected during each iteration. The accuracy of the best final model is 100%.





Gene	Corr	Intensity %
M19507_at	0,93	8,48
M13690_s_at	0,84	6,49
M96326_rna1_at	0,90	6,42
X13973_at	0,88	4,86
M31303_rna1_at	-0,78	4,48
J02783_at	0,86	4,44
D49950_at	0,86	3,64
X85116_rna1_s_at	0,84	3,48
X17042_at	0,84	3,39
M98399_s_at	0,89	3,34
U46499_at	0,78	3,12
M84526_at	0,87	2,87
L42379_at	0,86	2,39
U82759_at	0,85	2,39
Z29067_at	0,85	2,38
U05572_s_at	0,87	2,32
M37435_at	0,77	2,29
X66401_cds1_at	-0,74	2,28

Figure 3: Final selected genes.

Figure 3 reports, on the left side, a summarized representation of the shap values and, on the right side, the values for correlation and intensity for the most interesting genes found by our algorithm. In this context, it is important to compare our findings with the work of Al-Azani et al. [10] and Bennet et al [11]. Al-Azani et al. conducted an empirical study utilizing a feature selection technique that combined Chi-square (ChiS) and Information Gain (IG) methods. Their evaluation of various ensemble-based learning models, including bagging, random forests, stacking, voting, and boosting, culminated in a best classification accuracy of 96.88%. This study emphasizes the effectiveness of ensemble methods in improving model performance.

Conversely, Bennet et al. introduced a hybrid gene selection technique that integrates Support Vector Machine-Recursive Feature Elimination (SVM-RFE) with the Based Bayes Error Filter (BBF). Their approach involved ranking attributes with SVM-RFE and subsequently using BBF to eliminate redundant attributes, followed by classification with the SVM algorithm. Their efforts yielded an impressive classification accuracy of 97.2% on the Leukaemia dataset, underscoring the power of hybrid techniques in attribute selection.

5. Conclusions

The algorithm proposed in this work can be used as a valuable tool in genomics to identify protective (or not) sets of genes for a disease, suggesting potential pathways for further medical investigation. A natural direction for future development is to perform a large-scale assessment of the algorithm performances, by using state-of-the-art benchmark GEP datasets.

Acknowledgments

The research reported in the paper was partially supported by the PNRR projects “FAIR (PE00000013) - Spoke 9” and “Tech4You (ECS00000009) - Spoke 6”, under the NRRP MUR program funded by the NextGenerationEU. The National Plan for NRRP Complementary Investments (PNC, established with the decree-law 6 May 2021, n. 59, converted by law n. 101 of 2021) in the call for the funding of research initiatives for technologies and innovative trajectories in the health and care sectors (Directorial Decree n. 931 of 06-06-2022) - project n. PNC0000003 - AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM).

References

[1] L. Koumakis, Deep learning models in genomics; are we there yet?, Computational and Structural Biotechnology Journal 18 (2020) 1466–1473.

[2] E. Alhenawi, R. Al-Sayyed, A. Hudaib, S. Mirjalili, Feature selection methods on gene expression microarray data for cancer classification: A systematic review, Computers in Biology and Medicine 140 (2022) 105051.

[3] P. Bruno, F. Calimeri, A. S. Kitanidis, E. De Momi, Data reduction and data visualization for automatic diagnosis using gene expression and clinical data, Artificial Intelligence in Medicine 107 (2020) 101884.

[4] G. Graham, N. Csicsery, E. Stasiowski, G. Thouvenin, W. H. Mather, M. Ferry, S. Cookson, J. Hasty, Genome-scale transcriptional dynamics and environmental biosensing, Proceedings of the National Academy of Sciences 117 (2020).

[5] J. Meena, Y. Hasija, Application of explainable artificial intelligence in the identification of squamous cell carcinoma biomarkers, Computers in Biology and Medicine 146 (2022) 105505.

[6] M. R. Karim, M. Cochez, O. Beyan, S. Decker, C. Lange, Onconetexplainer: explainable predictions of cancer types based on gene expression data, in: 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), IEEE, 2019, pp. 415–422.

[7] P. Danaee, R. Ghaeini, D. A. Hendrix, A deep learning approach for cancer detection and relevant gene identification, in: Pacific symposium on biocomputing 2017, World Scientific, 2017, pp. 219–229.

[8] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[9] J. Li, H. Liu, Kent ridge biomedical data set repository. school of computer engineering, nanyang technological university, singapore, 2004.

[10] S. Al-Azani, O. S. Alkhnbashi, E. Ramadan, M. Alfarraj, Gene expression-based cancer classification for handling the class imbalance problem and curse of dimensionality, International Journal of Molecular Sciences 25 (2024) 2102.

[11] J. Bennet, C. Ganaprakasam, N. Kumar, A hybrid approach for gene selection and classification using support vector machine., International Arab Journal of Information Technology (IAJIT) 12 (2015).