Application of Data Mining and Machine Learning Methods to Develop a Disease Diagnosis System Based on Gene Expression Data

COLLECTIVE MONOGRAPH

Sergii Babichev, Ihor Liakh, Bohdan Durnyak

State Enterprise All-Ukrainian Specialized Publishing House "Svit"
Lviv, 2025

UDC: 004.048;004.94

Recommended for publication by the Academic Council of Kherson State University, Protocol No. 5 dated 21.11.2024.

Babichev S., Liakh I. and Durnyak B. Application of Data Mining and Machine Learning Methods to Develop a Disease Diagnosis System Based on Gene Expression Data.

This monograph addresses the pressing scientific and practical problem of developing and applying methodological foundations of information technology for processing gene expression data. This technology integrates gene ontology analysis, cluster-bicluster analysis, and deep learning methods for solving tasks in the field of bioinformatics. Its distinctive feature is higher adequacy in disease diagnosis compared to existing methods, achieved through hybridising existing methods and algorithms for big data processing, optimization of model hyperparameters using quantitative quality criteria, and considering the type of data being studied. The relevance of the research topic is underscored by the current absence of adequate information technology for processing gene expression data to identify significant and mutually correlated genes capable of diagnosing the diseases with high confidence and predicting their further development at the genetic level by modelling changes in the expression of target genes and their impact on the studied object. The efficiency of the diagnostic process can be enhanced through model hybridization, which involves the comprehensive application of various methods and algorithms to improve the reliability of decision-making at the corresponding stage. This approach necessitates the development of hybrid quality criteria for evaluating the outcome at each stage. Another way to enhance the effectiveness of gene expression data processing technology is the use of method ensembles, followed by comparing the results obtained by each method using appropriate quality criteria and calculating a comprehensive criterion for making the final decision on the model structure.

The monograph can be interested for scientists specialized in the fields of both development and applying data science techniques in various fields of scientific research.

Reviewers:
1. Prof. *Aleksandr Gozhyj*, DSc. (Petro Mohyla Black Sea National University, Ukraine)
2. Doc. *Viktor Mashkov*, DSc. (Jan Evangelista Purkyně University in Ústí nad Labem, Czech Republic)
3. Prof. *Volodymyr Hnatushenko*, DSc. (Dnipro University of Technology, Ukraine)

ISBN: **978-966-914-476-8**

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Relevance of the Problem

In the fields of technical cybernetics, bioinformatics, and precision medicine, the processing of gene expression data stands as a cornerstone for developing early diagnosis systems for advanced diseases. The intricate relationship between gene expression patterns and disease phenotypes offers a fertile ground for exploring novel diagnostic approaches. This exploration is particularly critical given the rising prevalence of complex diseases such as cancer, Parkinson's and Alzheimer's, which pose significant challenges to healthcare systems worldwide. The actuality of improving gene expression data processing methodologies is underscored by the potential to significantly enhance the accuracy and efficacy of disease diagnosis, ultimately leading to more personalized and effective treatment strategies.

Numerous scientific studies are currently focused on developing various disease diagnosis systems based on gene expression data. For example, in [6], the authors tackle the pressing issues posed by the COVID-19 pandemic by focusing on the lack of early diagnosis methods and comprehensive treatment solutions. By collecting key genes associated with COVID-19 and applying centrality and controllability analysis within Protein-Protein Interaction (PPI) networks and disease-related signalling pathways, the study identifies crucial hub and driver genes that play a significant role in the disease's formation and progression. The study [57] delves into the crucial link between epigenetics and the prognosis of colorectal cancer (CRC) patients by constructing a predictive model that assesses the treatment potential of epigenetic factors in CRC, utilizing gene expression data and identifying prognosis-related epigenetic genes through comprehensive analyses. The study successfully establishes a risk score model based on eight epigenetic-related genes, demonstrating its efficacy in predicting CRC patient outcomes in both training and validation sets, and further integrates it with clinical characteristics to enhance prognostic predictions

and suggest targeted therapeutic approaches. In [33], the authors focus on identifying key genes associated with Alzheimer's disease (AD) by analyzing microarray datasets to pinpoint differentially expressed genes (DEGs), employing bioinformatics tools for Gene Ontology and pathway enrichment, and constructing a protein-protein interaction network to isolate hub genes, whose predictive value was further validated through principal component analysis and histological examination of an AD mouse model. In [75], the authors delve into personalized therapy strategies for liver hepatocellular carcinoma (LIHC) patients by analyzing gene expression profiles and inflammation-related phenotypes to identify characteristic genes and lncRNAs linked to LIHC prognosis, subsequently developing a machine learning-based prognostic model, the Inf-PR model, which demonstrates superior predictive accuracy over traditional prognostic factors and existing models through ten-fold cross-validation. The model not only differentiates drug sensitivity and immune targets across prognostic risk groups, revealing distinct responses to FDA-approved drugs like lovastatin, sorafenib, doxorubicin, and lenvatinib, but also suggests that high-risk patients may benefit more from combining treatment with immunotherapy, offering a novel, individualized precision approach to augment current LIHC treatments. In [63], the authors investigate diabetic kidney disease (DKD), the deadliest complication of diabetes, by focusing on diverse programmed cell death (PCD) pathways as key indicators of renal function decline and potential drug research targets, utilizing microarray and single-nucleus RNA sequencing data to identify and analyze the activity of 13 PCD-related genes across different renal cell types. Through extensive analysis, including gene set variation and weighted gene co-expression network analysis, four core PCD pathways (entotic cell death, apoptosis, necroptosis, and pyroptosis) were identified and linked to significant roles in DKD progression, culminating in the development of a cell death-related signature (CDS) risk score that effectively predicts DKD diagnosis, immune cell infiltration levels, and glomerular filtration rates, underscoring the potential of these pathways as therapeutic targets.

Despite certain achievements in using gene expression data for disease diagnosis and treatment, significant challenges remain in early diagnosis, treatment personalization, model validation, and therapeutic target identification. Addressing these challenges requires integrated approaches that combine genetic data with clinical insights, advanced computational models, and thorough validation studies. The unsolved parts of the general problem include:

- There is a continuous need for the development of methods that can diagnose diseases at an earlier stage, as highlighted by the study on COVID-19. Early diagnosis is crucial for diseases such as cancers and neurodegenerative diseases, where early intervention can significantly alter the prognosis.

- Identifying gene expression patterns that can guide comprehensive treatment

strategies, including personalized therapy approaches, remains a challenge. Studies on colorectal cancer and liver hepatocellular carcinoma have made strides in linking gene expression with treatment options, but a gap exists in translating these findings into universally effective therapies.

- Effectively integrating genetic data with traditional clinical characteristics to enhance prognostic predictions and therapeutic decisions is still a developing area. While some studies have begun integrating these aspects, a more cohesive approach is needed across different diseases.

- While predictive models based on gene expression data have shown promise, increasing their accuracy, validating them across diverse populations, and ensuring they are applicable in clinical settings remain areas for further exploration.

- The effective use of varied data types, such as single-nucleus RNA sequencing and microarray data, in disease diagnosis and developing treatment strategies is an ongoing challenge. The ability to integrate and analyze these diverse data types to provide coherent insights into disease mechanisms and treatment responses needs further development.

The rationale behind the integrated use of gene ontology analysis, cluster and bicluster analysis, and deep learning techniques in addressing this challenge is multifaceted. Firstly, gene ontology analysis provides a structured framework for interpreting gene expression data, enabling the identification of biological processes and pathways that are most relevant to the disease under investigation. This fact is crucial for decreasing the vast array of genomic data. Secondly, cluster and bicluster analysis techniques facilitate the segmentation of gene expression data into meaningful groups or biclusters, where genes within a group exhibit similar expression patterns across a subset of conditions or samples. This segmentation is instrumental in uncovering the complex relationships between genes and disease phenotypes, which are often not linear and involve interactions among multiple genes. Finally, deep learning techniques offer the computational power and sophistication required to model these complex relationships, providing the ability to predict disease presence or progression with high accuracy based on gene expression profiles.

One of the primary challenges in the field of gene expression data processing is the high dimensionality of the initial experimental data, which complicates the identification of meaningful patterns and relationships. Additionally, the heterogeneity of disease mechanisms often results in subtle and complex gene expression changes, making it difficult to distinguish between disease states. Moreover, many current systems lack the integration of comprehensive biological knowledge, such as gene ontology, into the analysis process, which can limit the interpretability and

biological relevance of the findings. The actuality of research in this subject area is further magnified by the ongoing need to overcome these limitations and harness the full potential of gene expression data for disease diagnosis. An integrated approach combining gene ontology analysis, clustering and biclustering techniques, and deep learning models holds the promise of transcending these barriers, paving the way for developing next-generation diagnostic systems. Such systems would offer improved diagnostic accuracy and contribute to a deeper understanding of disease mechanisms at the molecular level, facilitating the discovery of novel therapeutic targets and personalized medicine approaches.

Figure 1.1 presents the block diagram of the proposed model for gene expression data processing. The components of this model are described and implemented in detail in the following chapters of this thesis. Its implementation involves the

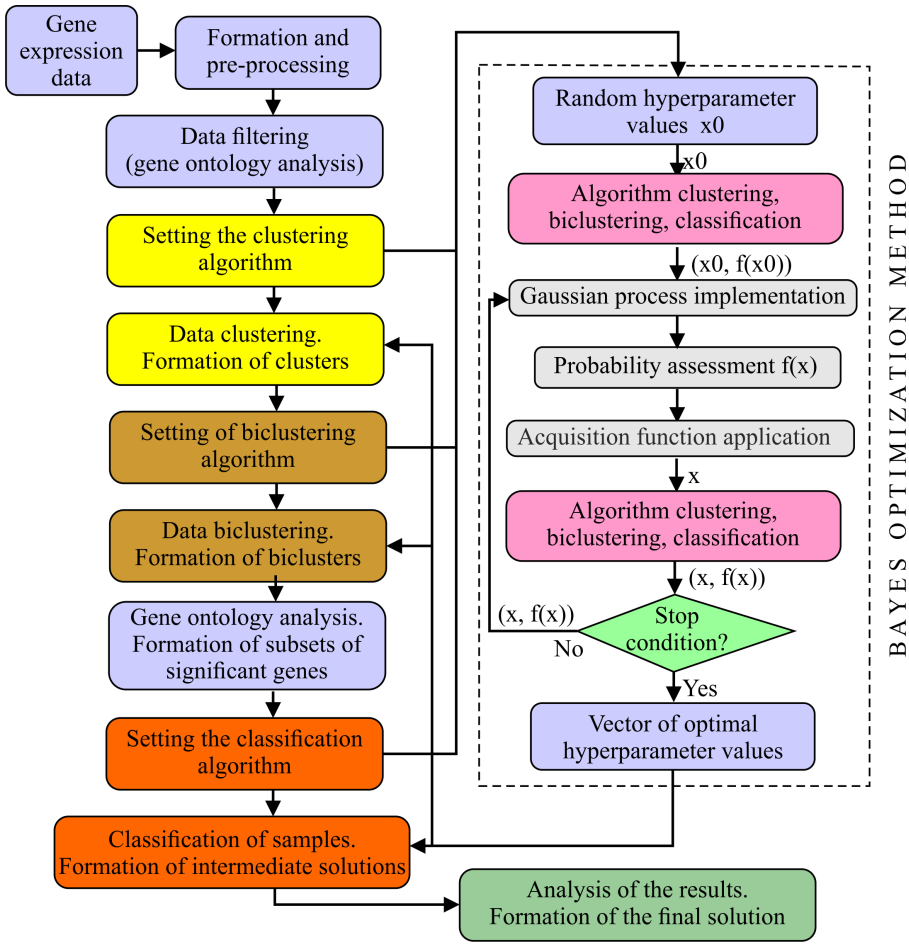

Figure 1.1: Model of the disease diagnosis system based on gene expression data

following stages:

1. Formation and preprocessing of gene expression data.

    1.1. Formation of the gene expression matrix. The practical implementation of this step depends on the type of experiment used to generate the experimental data. When applying experiments based on DNA microarrays, the process of forming and preprocessing experimental data involves four stages: background correction, normalization, PM correction, and summarization. Each stage can affect the accuracy of estimating the activity level of a specific gene under different experimental conditions in certain ways. For this reason, despite the lower cost of experimenting, the limitation of successful application of this method in early diagnosis systems of objects' conditions is determined by the low accuracy of determining the value of gene expression compared to the technique based on RNA molecule sequencing. Experimental data obtained by applying the method based on RNA molecule sequencing, in their initial state, contain a matrix of gene counts, varying values within a very high range. In this case, forming experimental data involves transforming the gene count values into a more convenient interval.

    1.2. Preprocessing of gene expression data. This step involves normalising gene expression data by removing unexpressed genes for all samples under study.

2. Data filtering and formation of subsets of significant and co-expressed genes.

    2.1. Application of gene ontology analysis for removing insignificant genes considering the type of biological organism and the number of genes determining the object's state. The threshold value can vary for different data within the range from 95% (p-value = 0.05) to 99% (p-value = 0.01), depending on the set goal and obtained results.

    2.2. Clustering of gene expression profiles using the Bayesian optimization algorithm to optimize the clustering algorithm hyperparameters. Within the framework of our research, we applied the spectral clustering algorithm [78, 73, 86, 62] Self Organizing Tree Algorithm (SOTA) with a correlation metric [38, 41], which is focused on clustering high-dimensional data, including gene expression profiles.

    2.3. Bicluster analysis of gene expression data allocated into clusters using the ensemble biclustering algorithm [52], the optimal hyperparameters of which are also determined using the Bayesian optimization algorithm.

The effectiveness of the ensemble biclustering algorithm for gene expression data processing was justified in [23]. Formation of biclusters of coherent gene expression data for each subset of data formed at the previous step of this procedure.

2.4. The second stage of data filtration involves applying gene ontology analysis to the data in formed biclusters to identify the most significant ontologies, taking into account the nature of grouping of both the genes and samples in the allocated biclusters. The next step involves identifying vectors of identifiers for significant genes for each ontology. The final step entails creating unique gene identifiers for each biclustering, followed by the formation of subsets of significant and mutually correlated gene expression data for each cluster.

3. Classification of objects, the attributes of which are the formed subsets of significant and mutually expressed genes.

   3.1. Tuning of the classifier involves selecting the optimal architecture and hyperparameter values by applying the Bayesian optimization algorithm with 10-fold cross-validation at each epoch of the Bayesian optimization algorithm implementation.

   3.2. Training, validation, and testing of the model. Forming intermediate decisions regarding the state of the object based on the application of gene expression data from the allocated clusters.

4. Analysis of the obtained results.

   4.1. Analysis of intermediate decisions for forming a final decision on the state of the object by evaluating the consistency of classification results of samples across different subsets of gene expression data. In this case, the state of the object is considered to be unambiguously identified if the classification results across different data sets are consistent. Otherwise, the state of the object is considered undetermined, requiring further clinical studies.

The following charts present the research results regarding the implementation of the stages of the hereinbefore presented procedure.

1.2 Organisation of the Monograph

The thesis is organized as follows:

Chapter 2 discusses the theoretical and experimental research on developing models for forming subsets of mutually expressed and significant gene expression

profiles. This includes various data mining techniques such as gene ontology (GO) analysis and the further application of the processed data in diagnostic systems. Typically, approximately 25,000 genes are active in the human genome. Creating a diagnostic system based on the complete set of genes is problematic due to the substantial computational resources required and the high degree of subjectivity in model results caused by the presence of complex noise components (genes not related to the respective disease being studied). As current gene expression database analysis shows, many genes are weakly expressed across all studied objects and can be removed without significant loss of information. This step can be implemented by applying various quantitative statistical and entropy criteria that allow the division of genes into informative and non-informative categories. Therefore, there is a need to form subsets of mutually expressed and significant genes through the comprehensive application of modern data clustering algorithms, fuzzy simulation, and gene ontology analysis. These issues are addressed in this chapter.

Chapter 3 focuses on applying deep learning methods in hybrid models for processing gene expression data. The suitability of using deep learning methods for this purpose is determined by the structure and large volume of experimental data, which typically contain thousands of objects and more than ten thousand attributes. The primary advantages of deep learning methods include their ability to handle complex and unstructured data, identify specific patterns in hierarchical representations, and form functions that allow for the high-accuracy identification of studied objects. Additionally, deep learning-based models can generate corresponding functions directly from raw data, enabling the discovery of hidden patterns and complex relationships within the data, which are difficult to detect using traditional methods. Deep learning models are inherently scalable, meaning they can effectively process large volumes of data through parallel or distributed computing architectures, significantly speeding up training and inference processes. The correct application of deep learning methods can improve the efficiency of diagnostic systems for complex objects by enhancing the accuracy of identifying the studied objects and increasing the objectivity of determining the state of the object through parallel processing of information.

Chapter 4 presents research on the development and application of hybrid models of data mining and machine learning (including deep learning) for forming subsets of significant genes to enhance the objectivity of object identification based on gene expression data. The chapter explores various deep-learning models for classifying objects using the complete set of genes, whose expression values are attributes of the studied samples. The analysis of research results demonstrates the high efficiency of the hybrid neural networks employed, although the models show considerable sensitivity to hyperparameter values, which determine the network's performance. This sensitivity indicates that hyperparameters must depend on the number of attributes

of the identified objects, complicating the application of deep neural networks to gene expression data within identified clusters. Consequently, there is a necessity to develop an adaptive model for object identification that utilizes an ensemble of methods from data mining and machine learning, where hyperparameter values adapt to the dimensionality of the formed subset of gene expression data. This chapter presents research findings on addressing this challenge through the application of modern cluster and bicluster analyses, GO analysis, and machine learning techniques.

Chapter 5 details the appication of the proposed models, methods, and algorithms in diagnostic systems based on gene expression data. The first subsection offers a detailed description of the experimental data used in the modeling process. Gene expression data from subjects tested for Alzheimer's disease were used, obtained through DNA microarray experiments. These data included samples from subjects with clinically identified diseases and samples where these diseases were not detected. The second set of experimental data comprised gene expression data related to various types of cancer. These data also included samples from both affected subjects and subjects without clinically identified cancer, with this set obtained through RNA sequencing methods. The following subsections present the step-by-step implementation results of the proposed information technology stages: from data preprocessing using the functions and modules of the "Bioconductor" package in the R programming language, forming subsets of significant and mutually expressed gene expression profiles using methods based on gene ontology analysis, cluster and bicluster analyses, to diagnosing the state of the studied object by applying a classifier based on deep learning methods to the formed subsets of gene expression data, with the final decision on the object's state being made at the last stage.

Each chapter builds on the previous, culminating in a comprehensive approach to developing and implementing advanced diagnostic systems based on gene expression data.

Chapter 2

Theoretical Studies on the Formation of Subsets of Co-expressed and Significant Gene Expression Profiles

This chapter contains parts of the papers [58, 28, 81, 30, 27, 82, 83, 29, 13, 10, 22].

2.1 Introduction

Gene expression data used for gene regulatory networks reconstruction are usually presented as a matrix $e_{i,j},\ i = \overline{1,n},\ j = \overline{1,m}$, where n and m are the numbers of genes and studied objects respectively. After deleting zero-expressed genes for all objects (unexpressed genes), approximately 25000 genes remain that define the genome of the corresponding biological organism. It should be noted that a large number of genes are lowly expressed for all objects, they determine certain processes occurring in the biological organism, but are not decisive in terms of the health status of the object (disease identified and studied). Therefore, in the first step, it is advisable to remove low-expressed genes for all objects. In the second step, it is advisable to remove genes whose expression values changed slightly when analyzing different types of objects (by variance or standard deviation) or change randomly (high Shannon entropy value), which corresponds to noise. These gene expression profiles do not allow us by the level of expression to unambiguously identify relevant objects according to the health status of the biological organism, and they can also be deleted from the database. The gene expression profile in this case means the vector of expression values of the corresponding gene, which are determined for all

investigated objects. In the context of this definition, co-expressed genes are genes whose expression values change accordingly for all studied objects. At the same time, the profiles of co-expressed genes allow identifying objects with high accuracy considering the state of health of the biological organism.

The procedure for generating co-expressed gene expression profiles assumes two stages. The first stage is to reduce the number of genes with low absolute value in the first step and low variance and high Shannon entropy in the second step. This raises the problem of determining the thresholding coefficient for each of the used criteria. Typically, the thresholding coefficients are determined empirically during the simulation process implementation taking into account the approximate number of genes that should remain after the implementation of this stage.However, taking into account the fact that co-expressed gene expression profiles should allow the identification of objects with the highest possible accuracy, the values of the thresholding coefficients for each of the criteria can be determined by the maximum value of the classification accuracy of the studied objects. Thus, the implementation of the concept of the formation of co-expressed gene expression profiles assumes the use of a hybrid model that involves the joint use of both data mining methods and machine learning techniques.

2.2 Fuzzy Model of Removing the Non-informative Gene Expression Profiles by Statistical Criteria and Shannon Entropy

The problem of removing non-informative gene expression profiles by statistical and entropic criteria was solved in [30, 22]. In the authors' mind, the gene expression profile was considered informative if the maximum expression value of this profile and variance is greater, and Shannon's entropy is less than the corresponding threshold values (boundary):

$$e_{ij} = \begin{Bmatrix} \underset{i=\overline{1,n}}{max}(e_{ij} \geq e_{bound}),\ and\, var(e_{ij}) \geq var_{bound}, \\ and\, entr(e_{ij}) \leq entr_{bound} \end{Bmatrix},\ j = \overline{1,m} \qquad (2.1)$$

where: n is the number of samples or objects to be examined; m is the number of genes.

The boundary values, in this case, were determined empirically during the simulation process, taking into account the approximate number of genes that should make up a subset of experimental data for further simulation. However, it should be noted that the concept proposed by the authors has a significant drawback. A high value of the variance of the corresponding gene expression profile or a low Shannon entropy value (according to these criteria, this gene expression profile is considered

informative) when low absolute values of gene expression for all studied objects does not mean that this gene expression profile is informative since, in terms of absolute values, this gene does not contribute to the high accuracy of identification of the studied objects. Thus, there is a necessity to set priorities for the relevant operations, either by entering the sequence of their application or by initializing the weights of a particular operation. However, in this instance, it is necessary to justify the choice of the value of the appropriate weight.

Within the framework of current research, this problem is solved on the basis of fuzzy logic inference system application [88, 76, 45, 87], and the priority of one or another operation is taken into account when creating a base of fuzzy rules that form the basis of fuzzy model. The formation of fuzzy rules requires the following steps:

- define the set of input variables: $X = x_1, x_2, ..., x_n$ with the corresponding terms for each variable: $T_{input} = t_i^p,\ i = \overline{1,n},\ p = \overline{1,q}$, where q is the number of terms corresponding to the i-th input variable;
- define the set of output variables: $T_{out} = t^r,\ r = \overline{1,q}$, where q, in this case, is the number of terms corresponding to the output variable;
- generate a finite set of fuzzy rules agreed with appropriate input and output variables:

$$\bigcup_{k=1}^{m} [\bigcap_{p=1}^{q} (x_i = t_p^q),\ when\ \omega_k] \longrightarrow (y = t^r),\ i = \overline{1,n},\ r = \overline{1,q} \tag{2.2}$$

  where: $k = \overline{1,m}$ is a number of fuzzy rules that make up a fuzzy database; ω_k is the weight of the k-th rule (determined in the case of priority rules existance).

In the general case, the fuzzy inference procedure involves the following steps:

- *fuzzification* or matching between the specific values of the input variables used in the model and the values of the corresponding membership functions, taking into account the relevant term corresponding to this membership function. At the stage of fuzzification, the membership functions, which are predetermined on the input variables, are applied to their input values, in other words, the values of the membership functions $\mu^{t_i^k}(x_i)$ of the input variable x_i for the term t_i^k are determined. The result of the fuzzification step implementation is a matrix of values of membership functions for all input variables, which are defined for all fuzzy rules included in the fuzzy database;
- *aggregation* or determination of the degree of truth of the conditions for each of the fuzzy rules by finding the level of "clipping" for the preconditions of each

rule using the operation *min*:

$$\alpha_k = \bigwedge_{i=1}^{n} [\mu^{t_i^k}(x_i)] \tag{2.3}$$

- *activating* or finding the degree of truth for each of the fuzzy rules by forming the truncated membership functions of the fuzzy sets for each of the fuzzy rules:

$$\mu'_k(y) = \alpha_k \wedge \mu_k(X) \tag{2.4}$$

where: $\mu_k(X)$ are the truncated membership functions for the vector of input variables corresponding to the k-th rule; $\mu'_k(y)$ is the resulting membership function for the output variable, which corresponds to the k-th rule;

- *accumulation* or formation of the membership function of the resulting fuzzy set for the output variable using the operation *max*:

$$\mu^{\Sigma}(y) = \bigvee_{k=1}^{m} [\mu'_k(y)] \tag{2.5}$$

- *defuzzification* or finding a crisp value of the output variable by applying the appropriate operation to the obtained membership function of the resulting fuzzy set. The defuzzification operation can be implemented using various methods: calculation of the obtained function gravity centre, the centre of the figure area, and the left or right modal values. Within the framework of the proposed model, the most common centre of gravity method has been used:

$$Y = \frac{\int_{min}^{max} y \cdot \mu_{\Sigma}(y) dy}{\int_{min}^{max} \mu_{\Sigma}(y) dy} \tag{2.6}$$

The practical implementation of the fuzzy inference model within the framework of the research assumes the following steps:

1. Determining the ranges of variation of the values of the input statistical criteria, Shannon entropy and the output parameter (the significance of the profile in the ability to identify the object).
2. Defining the fuzzy sets membership functions for input and output parameters.
3. Formation of a base of fuzzy rules that form a fuzzy inference.
4. Choice of fuzzy inference algorithm and method to form the crisp value of output variable.

5. Determining the quantitative criteria for assessing the adequacy of the model for its testing.

The range of the input parameters values variation within the proposed model was determined by analyzing the general statistics, while the absolute values of gene expressions in the first step were determined by the maximum value of expression for each profile. Then, the general statistic was formed for the obtained vector of maximum values of gene expressions, vector of variance of gene expression profiles and Shannon entropy. To create a fuzzy model, the interquartile ranges of the appropriate criteria values variation were used. The formed ranges were divided into three subranges with the corresponding terms. For variance and maximum absolute values of gene expressions, these ranges were the following: $0\% \leq x < 25\%$ – "Low" (Low); $25\% \leq x < 75\%$ – "Medium" (Md); $x \geq 75\%$ – "High" (Hg). For Shannon's entropy: $x \geq 75\%$ – "High" (Hg); $25\% \leq x < 75\%$ – "Medium" (Md); $x < 25\%$ – "Low" (Low). The range of the output parameter variation (significance of the profile) in the proposed models varied from 0 to 100 and was divided into five equal intervals: $0\% \leq x < 20\%$ – "Very low" (VLow); $20\% \leq x < 40\%$ – "Low" (Low); $40\% \leq x < 60\%$ – "Medium" (Md); $60\% \leq x < 80\%$ – "High" (Hg); $80\% \leq x \leq 100\%$ – "Very high" (VHg). Regarding the fuzzy sets membership functions, the trapezoidal membership functions were used for the input parameters for the values with the terms "Low" and "High", and the triangular membership functions was used for the medium range of values (Md). The triangular fuzzy sets membership functions were applied for all subranges of the output parameter. It should be noted that the parameters of the fuzzy sets membership functions of input parameters involve an adjustment during the simulation process implementation taking into account the nature of the distribution of gene expression values in the studied experimental data.

Table 2.1 presents the terms of the fuzzy rules base which were used during the fuzzy model creation.

As can be seen from Table 2.1, the priority parameter for identifying the significance of the gene expression profile is the maximum of the gene profile expression values, which are determined for all studied objects. As noted hereinbefore, genes whose expression values are reletive low for all objects are not decisive to the identification of objects and can be deleted despite high variance and/or low Shannon entropy values. The combination of Shannon entropy and variance values, in this case, are corrective ones.

Fuzzy logical equations linking the values of the membership functions of the input and output variables in the proposed model can be represented as follows:

$$\mu^{VHg}(y) = \mu^{Hg}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr) \tag{2.7}$$

Table 2.1: Terms of the fuzzy model knowledge base to form the subsets of co-expressed gene expression profiles

No	Maximum expr. value	Variance	Shannon entropy	Significance of profile
1	Hg	Hg	Low	VHg
2	Hg	Md	Low	Hg
3	Hg	Hg	Md	Hg
4	Hg	Md	Md	Hg
5	Hg	Low	Md	Hg
6	Hg	Hg	Hg	Hg
7	Hg	Low	Hg	Md
8	Hg	Md	Hg	Md
9	Md	Hg	Low	Hg
10	Hg	Low	Low	Hg
11	Md	Md	Low	Md
12	Md	Low	Low	Md
13	Md	Hg	Md	Md
14	Md	Md	Md	Md
15	Md	Low	Md	Md
16	Md	Hg	Hg	Md
17	Md	Low	Hg	Low
18	Low	Hg	Low	Low
19	Low	Md	Low	Low
20	Low	Hg	Md	Low
21	Low	Low	Low	Low
22	Low	Md	Md	Low
23	Low	Low	Hg	VLow

$$\begin{aligned}\mu^{Hg}(y) = {} & [\mu^{Hg}(max_e xpr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Low}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Hg}(entr)]\end{aligned} \tag{2.8}$$

$$\begin{aligned}\mu^{Md}(y) = {} & [\mu^{Hg}(max_e xpr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Low}(var) \wedge \mu^{Md}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Hg}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Hg}(max_e xpr) \wedge \mu^{Md}(var) \wedge \mu^{Hg}(entr)] \vee \\ & [\mu^{Md}(max_e xpr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)]\end{aligned} \tag{2.9}$$

$$\begin{aligned}\mu^{Low}(y) = [\mu^{Md}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr)] \vee \\ [\mu^{Low}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Low}(entr)] \vee \\ [\mu^{Low}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Low}(entr)] \vee \\ [\mu^{Low}(max_expr) \wedge \mu^{Hg}(var) \wedge \mu^{Md}(entr)] \vee \\ [\mu^{Low}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Low}(entr)] \vee \\ [\mu^{Low}(max_expr) \wedge \mu^{Md}(var) \wedge \mu^{Md}(entr)]\end{aligned} \tag{2.10}$$

$$\mu^{VLow}(y) = \mu^{Low}(max_expr) \wedge \mu^{Low}(var) \wedge \mu^{Hg}(entr) \tag{2.11}$$

The membership function of the final fuzzy subset for the output variable "Significance of the profile" is formed according to the following equation:

$$\mu^{\Sigma}(y) = \mu^{VHg}(y) \vee \mu^{Hg}(y) \vee \mu^{Md}(y) \vee \mu^{Low}(y) \vee \mu^{VLow}(y) \tag{2.12}$$

The last step determines the crisp value of the output variable as the gravity center of the resulting figure in accordance with formula (2.6).

2.2.1 Simulation Regarding Practical Implementation of the Proposed Fuzzy Logic Inference Model

Approbation of the proposed technique of forming groups of co-expressed gene expression profiles based on both the statistical criteria and Shannon entropy was carried out using the gene expressions dataset of patients studied in the early stages of lung cancer. GSE19188 data [47] were taken from the freely available Gene Expression Omnibus database [2] and contained gene expression data from 156 patients, of whom 65 were identified as healthy in clinical trials and 91 had early-stage cancers. Data processing was performed by using the tools of the Bioconductor package [1] of the programming language R [67]. In the initial state, the data contained 54675 genes. In the first stage, for each gene expression profile, the maximum value of gene expressions, variance and Shannon entropy was calculated using the James-Stein shrinkage estimator. Figure 2.1 shows the box plots of the obtained values distribution, the analysis of which allows us to conclude regarding the reasonable of using the hereinbefore set ranges of relevant criteria values for setting up a fuzzy logic inference model. Really, the most informative gene expression profiles have high values of expression and variance and low values of Shannon entropy.

Figure 2.2 shows the membership functions of both the input and the output parameters fuzzy sets used in the proposed model. Figure 2.3 presents the stepwise procedure described hereinbefore and implemented during the simulation procedure that allows us both to form the subsets of co-expressed gene expression profiles and to evaluate the effectiveness of the proposed fuzzy model by analyzing the results of

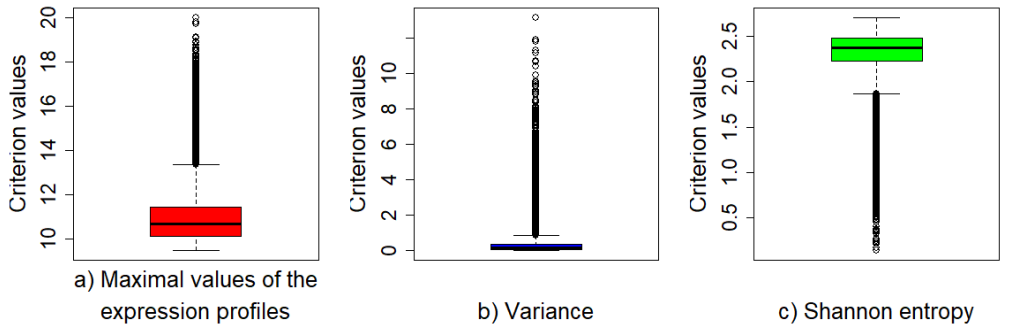

Figure 2.1: The nature of the distribution of statistical criteria and Shannon entropy of gene expression profiles of patients studied for early-stage lung cancer

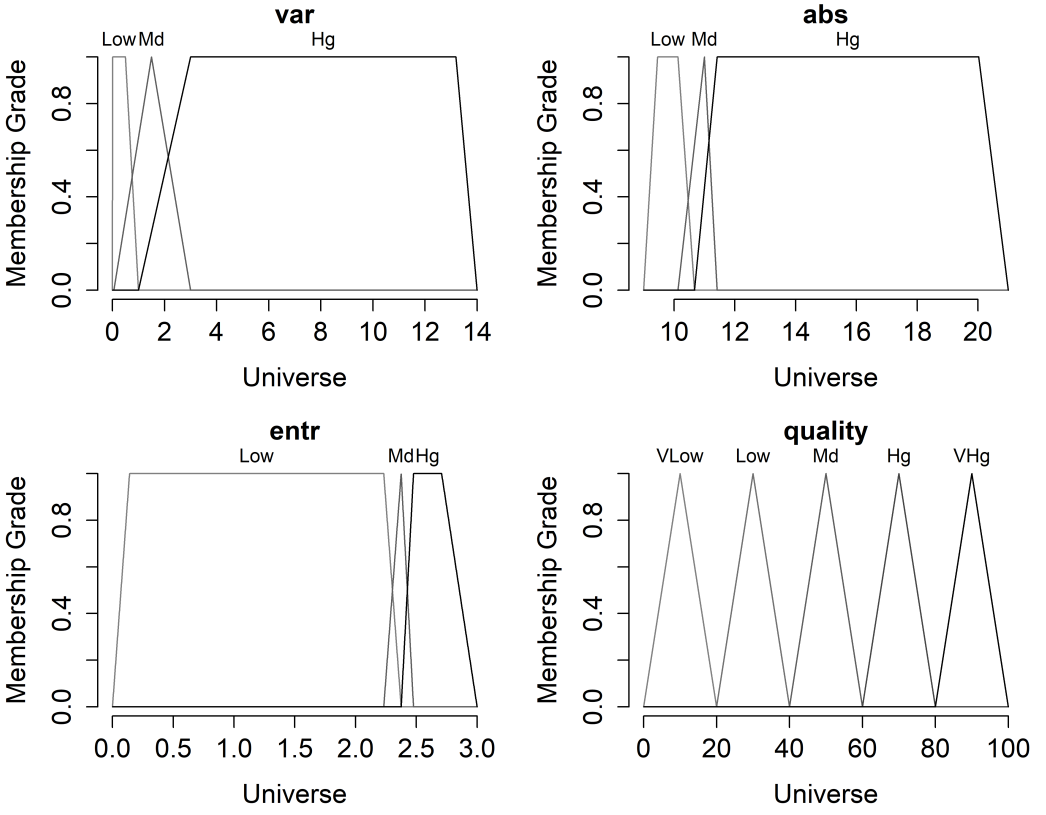

Figure 2.2: The membership functions of fuzzy sets of input and output parameters used in the fuzzy model of generating co-expressed gene expression profiles

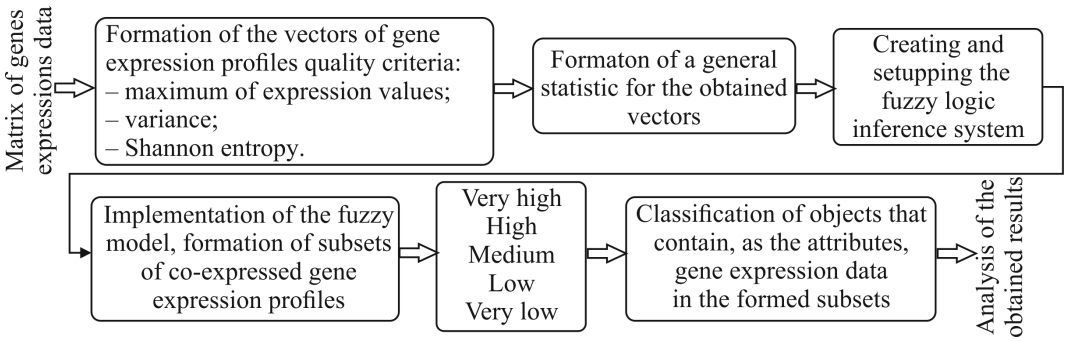

Figure 2.3: Structural block chart of a stepwise procedure to form subsets of co-expressed gene expression profiles based on the joint use of fuzzy logic inference system and objects classification technique

investigated objects classification that contain, as the attributes, formed subsets of gene expression data.

Algorithm 1 presents the stepwise procedure of the fuzzy logic inference model model implementation.

Figure 2.4 presents the results of the simulation regarding the application of the fuzzy inference model for the formation of gene expression profiles subsets of the different significance levels. Considering that only 29 genes from 54675 ones were identified as "Very High" significance, the groups containing genes with "Very High" and "High" significance levels were pooled for further simulation. The analysis of the obtained results allows us to conclude about the adequacy of the proposed fuzzy inference model for dividing a set of genes into corresponding subsets by the number of genes. It is well known that the human genome consists of approximately 25000 active genes. From this point of view, the allocation of 13734 genes of very high and high significance and 13096 genes of medium significance for further processing is reasonable. Genes with low and very low significance can be removed from the data as uninformative ones.

The next stage that may confirm or refute the conclusion regarding the adequacy of the results obtained by applying the fuzzy inference system to remove the non-informative gene expression profiles according to the used criteria is to apply a classifier to identify objects that contain, as attributes, the allocated gene expression data in the corresponding subsets.

2.2.2 Assessing the Fuzzy Inference Model Adequacy by Applying the Gene Expression Data Classification Technique

The assessment of the objects classification quality within the framework of the research was carried out using errors of both the first and second kind. The following

Algorithm 1: Stepwise Procedure for Gene Expression Profiles Analysis and Processing Using the Fuzzy Logic Inference Model

Data: Gene expression profiles

Stage I: Formation of quality criteria vectors

- **Step 1.1: Calculate metrics for each gene expression profile**
  - **foreach** *gene expression profile* **do**
    - Calculate the expression maximum value;
    - Calculate the variance;
    - Calculate the Shannon entropy;
- **Step 1.2: Calculate general statistics**
  - Calculate ranges of the appropriate values variation;
  - Calculate quantiles (25%, 50%, 75%) of variation of the appropriate range;

Stage II: Creation, debugging, and implementation of fuzzy logic inference system

- **Step 2.1: Form the structure of the fuzzy logic inference system**
  - Formalize the model structure;
  - Determine fuzzy sets membership functions for input and output parameters;
  - Form the model fuzzy rules base;
- **Step 2.2: Apply the fuzzy logic inference model to gene expression profiles**
  - Form subsets of significand gene expression profiles;
  - Consider statistical criteria and Shannon entropy values;

Stage III: Assessing the fuzzy model adequacy by applying a classifier

- **Step 3.1: Choose a classifier and form classification quality criteria**
  - Choose a classifier based on experimental data type;
  - Form classification quality criteria;
- **Step 3.2: Implement the objects classification procedure**
  - Classify objects with attributes containing gene expression data from formed subsets;
- **Step 3.3: Calculate classification quality criteria**
  - Calculate the quality criteria of the classification;

Stage IV: Analysis of the obtained results

- **Step 4.1: Make decisions concerning the adequacy of the fuzzy logic inference model**
  - Analyze correlation between classification results and gene expression profiles significance;

Result: Subsets of various significance level gene expression profiles

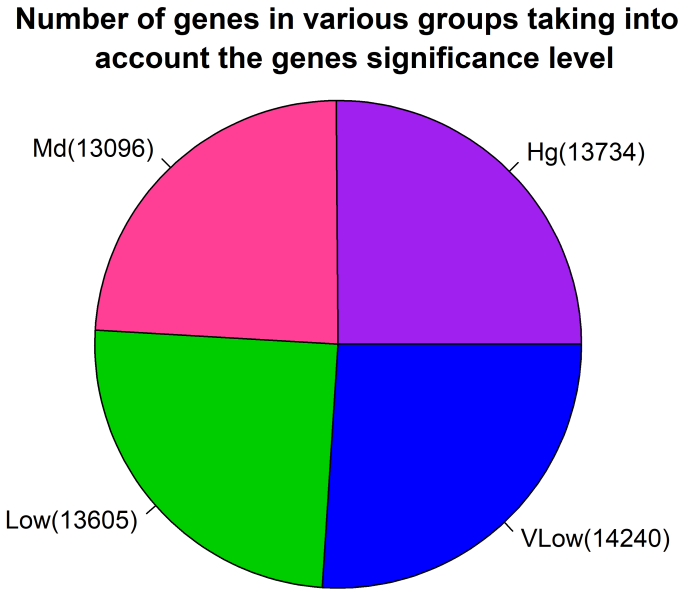

Figure 2.4: Simulation results regarding the application of the fuzzy logic inference model for the formation of subsets of gene expression profiles of different significance levels according to statistical criteria and Shannon entropy

criteria were applied as the quality criteria:

- objects classification accuracy:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.13}$$

  where: TP and TN are the correctly identified positive and negative cases respectively (for example, the presence or absence of the disease); FP and FN are the mistakenly identified positive and negative cases (the errors of the first and the second kind);

- F-measure is defined as the harmonic average of Precision (PR) and Recall (RC):

$$F = \frac{2 \cdot PR \cdot RC}{PR + RC} \tag{2.14}$$

  where:

$$PR = \frac{TP}{TP + FP}; \quad RC = \frac{TP}{TP + FN}$$

Table 2.2: The confusion table to identify the errors of the first and the second kind

State of the object by the results of clinical testing	Results of the objects classification	
	Patient(True - 1)	Healthy(False - 0)
Patient(True - 1)	TP(True Positives)	FN(False Negatives)
Healthy(False - 0)	FP(False Positives)	TN(True Negatives)

- Matthews Correlation Coefficient (MCC):

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{2.15}$$

Table 2.2 presents the procedure for forming a confusion matrix, which is the basis for calculating the classification quality criteria in accordance with formulas (2.13) -(2.15).

In this instance, the classification accuracy is maximal one (100%) if all objects are correctly identified and errors of the first (FP) and second (FN) kind are absent. The values of the F-index and MCC criterion are also maximal ones and they are equal to 1.

The second type of criterion which was used in the research to assess the object classification quality is based on ROC (Received Operating Characteristic) analysis. Application of this criterion assumes the calculating the area under the ROC curve AUC (Area Under Curve). A larger area corresponds to a higher quality of the object classification.

The choice of the classifier is determined by the peculiarity of experimental data. When the gene expression data is used as the experimental ones, the key feature is a large number of attributes (the number of genes that determine the state of the studied object). As was noted hereinbefore, 156 patients were used as gene expression data during the simulation process, of which 65 were identified as healthy by the clinical trials and 91 were identified as ill with early-stage of cancers tumour. The initial number of genes (54675) was divided into four groups according to statistical criteria and Shannon entropy (Figure 2.4). Each group contained approximately 13000 genes which were used as the input data of the classifier. In this case, the classifier should be focused on big data. In [37, 32, 56], the authors present the results of research regarding the use of convolutional neural networks as a classifier for the identification of objects based on gene expression data. The authors investigated different topological structures of this type of network and proved their effectiveness for the classification of objects based on high-dimensional gene expression data. However, it should be noted that the correct use of convolutional neural networks involves the formation of convolutional layers to supplement the data with profiles with zero expression values to obtain the required number of genes. In the current

Table 2.3: The results of the simulation regarding the classification of objects based on gene expression data of various significance level

Significance of genes	Classification quality criteria				
	Accuracy,%	Sensitivity	Specifity	F-measure	MCC
High	98.4	1	0.973	0.992	0.967
Medium	93.5	1	0.9	0.967	0.873
Low	90.3	0.917	0.917	0.894	0.801
Very Low	85.5	0.870	0.846	0.862	0.701

research, this step may affect the results, which is undesirable. For this reason, the use of convolutional neural networks at this stage of simulation is not reasonable. In [22, 30], the authors presented the results of studies focused on comparing binary classifiers to identify objects based on high-dimensional gene expression data. The authors have shown that the Random Forest Binary Classification Algorithm is more efficient for identifying objects based on gene expression data than other similar classifiers. For this reason, this classifier was used in the current research.

The simulation results regarding the identification of objects that contain gene expression data as attributes are presented in Table 2.3 and Figure 2.5. An analysis of the obtained results confirms the expediency of using the proposed fuzzy logic inference model for the formation of subsets of gene expression profiles of different significance levels according to statistical criteria and Shannon entropy. The values of the classification quality criteria presented in Table 2.3 gradually increase with the transition from subsets of gene expression profiles with a very low significance level to a subset of gene expression profiles with a high significance level. The F-measure and MCC criterion values, in these cases, also match within the margin of admissible error.

An analysis of the AUC criterion values (Area Under ROC-curve) also confirms the feasibility of using a fuzzy inference system to divide gene expression profiles into subsets of genes of various significance levels. However, it should be noted that this type of criterion is significantly less sensitive for gene expression profiles, which are allocated into the subsets with high and medium significance levels. Moreover, the AUC criterion value for a subset of genes with very low significance is higher than the similar value for the subset of gene expression profiles with low significance, which is not correct and contradicts the values of the classification quality criteria presented in Table 2.3. However, it should be noted that this criterion allows us to divide the set of gene expression profiles into two subsets: a subset of informative genes in this case contain genes with high and medium significance levels; other genes are removed as uninformative ones.

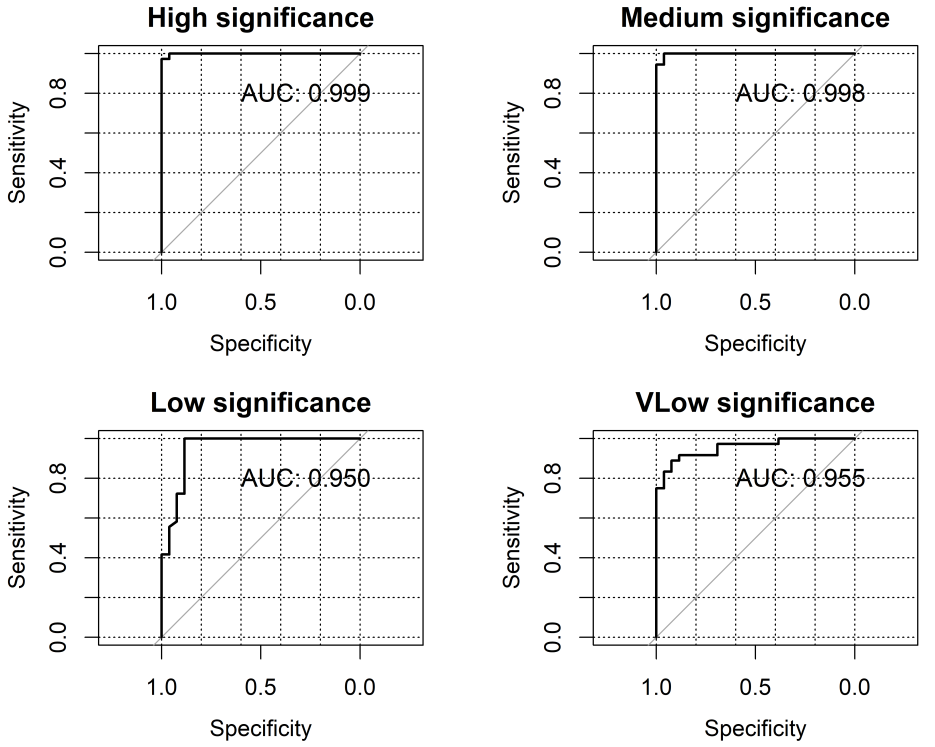

Figure 2.5: The results of ROC analysis to assess the effectiveness of a fuzzy model for the formation of subsets of gene expression profiles by the level of their significance

2.3 Formation of Gene Expression Profile Subsets Based on Statistical and Entropy Criteria Using the Harrington Desirability Function

The main drawbacks of the fuzzy model for the formation of subsets of mutually expressed gene expression profiles based on the analysis of statistical criteria and Shannon entropy, presented in the previous section, are the high labor intensity of information processing and the high sensitivity to model parameters, including the fuzzy knowledge base. An alternative method for solving multicriteria problems is the method based on the Harrington desirability function, which is currently successfully applied in various fields of scientific research [66, 50]. The Harrington desirability method is based on the following equation:

$$d = \exp(-\exp(-Y)) \tag{2.16}$$

where:

- Y is a dimensionless parameter, whose value ranges from -2 to 5;
- d is the individual desirability, which corresponds to one of the criteria used in the process of forming a generalized indicator.

Figure 2.6 shows the Harrington desirability function, which forms the basis for the formation of a generalized criterion, the value of which determines the appropriate decision-making. It is evident that the boundaries that separate the extreme

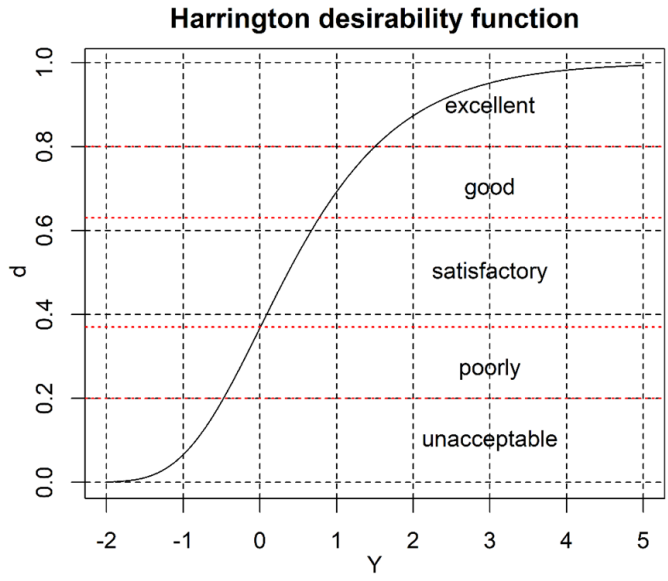

Figure 2.6: The Harrington desirability function and standard marks on the desirability scale

intervals of desirability values 0.2 (unacceptable - poor) and 0.8 (good - excellent) are conditional and can be adjusted depending on the nature of the changes in parameter values inputted into the model. The boundaries that separate the corresponding intervals within the range $0.37 = \frac{1}{e}$ (poor - satisfactory) and $0.63 = 1 - \frac{1}{e}$ (good - excellent) are fixed and correspond to the points of intersection of the desirability function. Within the proposed model, it is assumed that the nature of the change in the parameter Y and the values of the criteria inputted into the model follow a linear law.

The stepwise procedure for calculating the generalized significance index of gene expression profiles using the Harrington desirability method includes the following steps:

1. Determining the coefficients of the linear equations for transforming the values of statistical criteria and Shannon entropy into the value of the indicator Y

considering the boundary values of the respective criteria and the nature of their change:

$$\begin{aligned} Y_{min} &= a_1 + b_1 \cdot max\_expr_{min} \\ Y_{max} &= a_1 + b_1 \cdot max\_expr_{max} \\ Y_{min} &= a_2 + b_2 \cdot var\_min \\ Y_{max} &= a_2 + b_2 \cdot var\_max \\ Y_{min} &= a_3 - b_3 \cdot entr\_max \\ Y_{max} &= a_3 - b_3 \cdot entr\_min \end{aligned} \tag{2.17}$$

where $Y_{min} = -2$; $Y_{\max} = 5$.

2. Determining the values of the parameter Y for each of the criteria used in the model as input data:

$$\begin{aligned} Y_{max\_expr} &= a_1 + b_1 \cdot max\_expr \\ Y_{var} &= a_2 + b_2 \cdot var \\ Y_{entr} &= a_3 - b_3 \cdot entr \end{aligned} \tag{2.18}$$

3. Calculating the private desirability for each value of the gene expression profiles significance criteria:

$$\begin{aligned} d_{max\_expr} &= exp(-exp(-Y_{max\_expr})) \\ d_{var} &= exp(-exp(-Y_{var})) \\ d_{entr} &= exp(-exp(-Y_{entr})) \end{aligned} \tag{2.19}$$

4. Calculating the generalized significance index of gene expression profiles as the geometric average of the private desirabilities:

$$GI = \sqrt[3]{d_{max\_expr} \cdot d_{var} \cdot d_{entr}} \tag{2.20}$$

A higher value of the generalized index (2.20) corresponds to a higher level of significance of the gene expression profile. Algorithm 2 shows the implementation of the hereinbefore presented procedure. In Figure 2.7, box plots of the private desirabilities and the generalized index, which determines the significance level of gene expression profiles, are shown. These were calculated for the maximum expression values, variance, and Shannon entropy of gene expression profiles in patients studied for lung cancer.

The analysis of the obtained diagrams allows us to conclude that, based on both the private desirability values and the generalized significance index of the gene expression profiles, according to standard desirability scale estimates (Figure 2.6),

Algorithm 2: Stepwise Procedure for Gene Expression Profiles Analysis and Processing Using the Harrington Desirability Method

Data: Gene expression profiles

Stage I: Forming the Vectors of Criteria

- **Step 1.1: Calculate metrics for each gene expression profile**
  - **foreach** *gene expression profile* **do**
    - Calculate the maximum expression value;
    - Calculate the variance;
    - Calculate the Shannon entropy;
    - Form vectors of the obtained values;

Stage II: Forming Subsets of Mutually Expressed Gene Expression Profiles Using the Harrington Desirability Method

- **Step 2.1: Calculate coefficients for transforming scales**
  - Calculate coefficients of the linear equations according to equations (2.17);
- **Step 2.2: Calculate indicator Y values**
  - **foreach** *value of the statistical criteria and Shannon entropy* **do**
    - Calculate the value of the indicator Y using formula (2.18);
- **Step 2.3: Calculate private desirabilities and significance index**
  - Calculate the private desirabilities for each criterion using formula (2.19);
  - Calculate the generalized significance index using formula (2.20);
- **Step 2.4: Form subsets of gene expression profiles**
  - Form subsets of gene expression profiles of varying significance degrees;

Stage III: Applying Classification Technique

- **Step 3.1: Select and configure the classifier**
  - Select and configure the classifier;
  - Form classification quality criteria;
- **Step 3.2: Classify objects with gene expression data**
  - Classify objects that contain the allocated gene expression data of as attributes;
- **Step 3.3: Calculate classification quality criteria**
  - Calculate the quality criteria for data classification;
  - Form vectors of these criteria values;

Stage IV: Analyzing the Obtained Results

- **Step 4.1: Analyze the classification results**
  - Analyze the results of classifying objects;
  - Form a conclusion regarding the adequacy of the proposed model;

Result: Subsets of various significance gene expression profiles

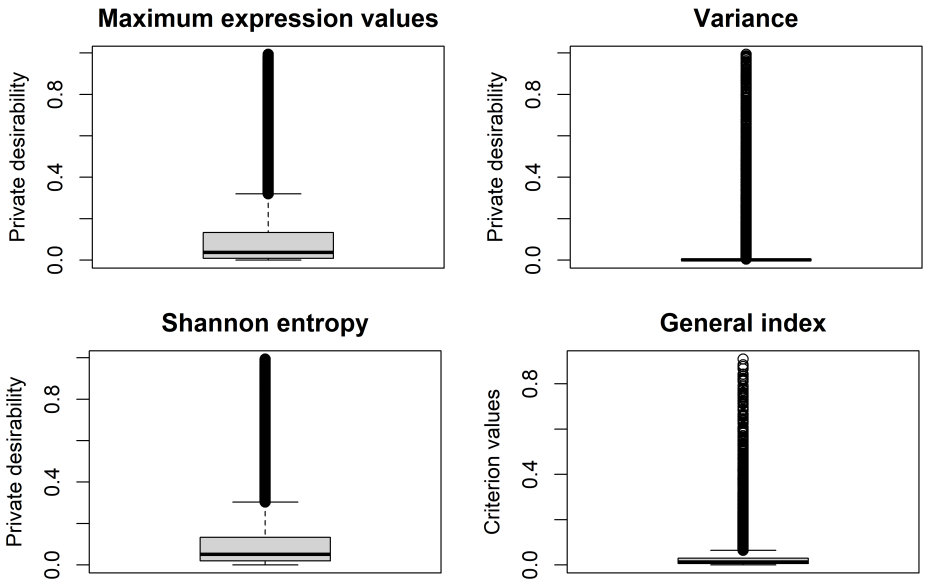

Figure 2.7: The box plots of the private desirabilities and the generalized index, which determine the significance level of gene expression profiles

most profiles are identified as having a very low level of significance. The modeling results showed that out of 54,675 profiles, 54,087 profiles were identified as having a very low level of significance. Meanwhile, 15 genes were identified as having a very high level of significance, 38 as genes with a high level of significance, and 142 and 393 profiles were identified as genes with medium and low levels of significance, respectively. However, it should be noted that the significant genes according to statistical criteria and Shannon entropy are not mutually expressed genes. This fact is confirmed by the results of the classification of objects that contain gene expression data of the corresponding subsets as attributes (Table 2.4).

Table 2.4: Modeling results for the classification of objects based on gene expression data of varying significance levels using the Harrington desirability method

Gene Significance	**Accuracy, %**	**Sensitivity**	**Specificity**	**F-index**	**MCC**
Very High	87.1	0.909	0.850	0.890	0.736
High	96.8	1	0.947	0.894	0.935
Medium	91.9	0.862	0.970	0.890	0.841
Low	96.8	1	0.947	0.894	0.935

At first glance, the results may not seem logical. However, this fact only indicates

that the application of the proposed technique allows us to divide the initial set of gene expression profiles into significant and non-significant based on the respective quantitative criteria. The number of significant genes is determined by the threshold value of the generalized significance index, which can vary depending on the nature of the experimental data. In the current studies, considering the nature of the changes in the values of the generalized significance index of gene expression profiles, the threshold value separating informative and non-informative profiles was chosen to be 0.04. This resulted in 9,630 gene expression profiles being selected. To evaluate the adequacy of the model using a classifier, this set of genes was divided into three subsets: $0.04 \leq GI < 0.2$ – medium significance (9,042 genes); $0.2 \leq GI < 0.37$ – high significance (393 genes); $GI \geq 0.37$ – very high significance (195 genes). The classifier was applied to the entire set of gene expression data (9,630 profiles) and to the gene expression profiles in the formed subsets. The classification results are presented in Table 2.5.

Table 2.5: Results of the model operation considering the Harrington desirability method

Gene Significance	**Accuracy, %**	**Sensitivity**	**Specificity**	**F-index**	**MCC**
Entire Set	91.9	1	0.878	0.958	0.842
Very High	91.9	0.862	0.970	0.890	0.841
High	96.8	1	0.947	0.984	0.935
Medium	91.9	1	0.878	0.958	0.842

As can be seen, in all cases, the classification criteria values are quite high but not maximal. Moreover, the results of classifying objects that contain a large number of gene expression values as attributes (9,630 and 9,042) are the same, indicating a large number of differentially expressed gene expression profiles, which can influence the classification accuracy. The highest classification quality is achieved when using 393 genes of high significance, which can be explained by the significantly smaller number of genes on the one hand and the presence of a larger number of mutually expressed genes on the other hand.

2.4 Model for Forming a Subset of Significant Genes Based on Gene Ontology Analysis

One of the modern methods used in bioinformatics to form subsets of significant genes based on statistical criteria, considering the type of object under study, is the method based on gene ontology (GO) analysis. The main idea of the method is to use gene ontology to identify genes that have significant biological importance in the context of a particular study or experiment. The key aspects of this method include the following:

- **Gene Ontology (GO)**: GO is a hierarchical database that classifies gene functions into three main categories: biological processes, molecular functions, and cellular components. Each gene is associated with certain GO terms that describe its role in the cell.
- **Selection of Significant Genes**: Within the GO analysis framework, genes that have a high degree of association with certain biological processes, molecular functions, or cellular components are identified. This can be done using statistical tests to compare the frequency of GO terms among the genes of interest with their frequency in the general population of genes.
- **Enrichment Analysis**: The main part of GO analysis involves determining whether certain GO terms are overrepresented (enriched) among a set of significant genes. This may indicate that these genes are collectively involved in specific biological processes or functions.
- **Functional Interpretation**: The results of GO analysis can be used for the functional interpretation of genomic data. For example, if it is found that genes associated with a particular disease are often linked to a specific biological process, this may indicate a key role of this process in the development of the disease.
- **Statistical Analysis**: Various methods such as Fisher's exact test or Chi-squared analysis are used to verify the statistical significance of GO term enrichment.

Figure 2.8 shows the structural diagram of the step-by-step procedure for applying GO analysis to identify significant genes based on GO annotation.

In general, the practical implementation of the above procedure involves the following steps:

1. **Data Preparation**. At this stage, a list of genes contained in the studied data is formed. In the next step, the genes are annotated using existing databases that provide information about their association with various GO terms.
2. **Creation of the Gene Ontology (GO) Object**. At this stage, an ontology object is created, which includes information about all GO terms and their relationships.
3. **Application of Test Statistics**. At this step, statistical tests are applied to the gene expression data, comparing the frequency of each GO term in the selected gene set with the frequency in the background set (general population of genes). In this studies, ANOVA and Fisher's tests were conducted at this stage.

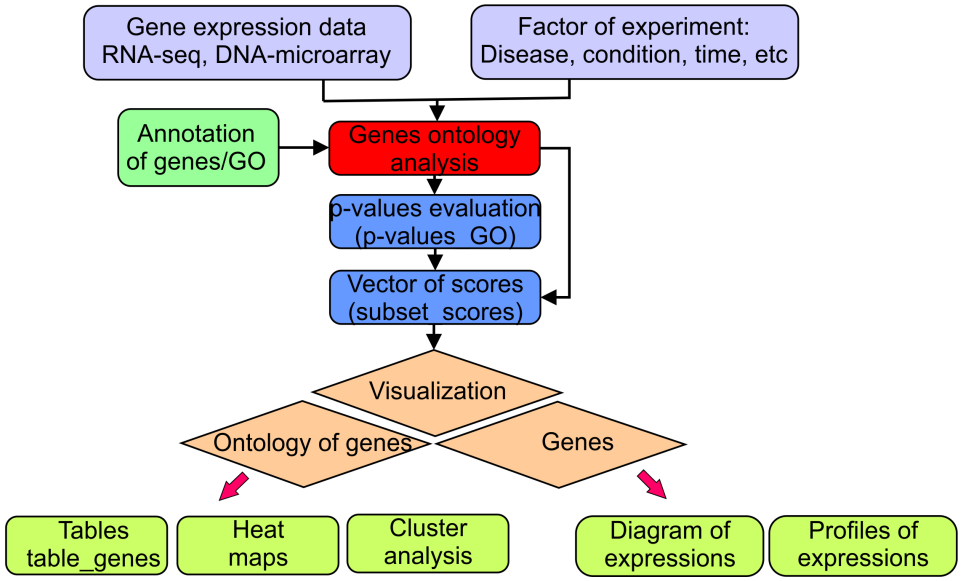

Figure 2.8: Structural diagram of the step-by-step procedure for applying GO analysis to identify significant genes based on GO annotation

4. **GO Term Enrichment Analysis**. This procedure involves assessing whether certain GO terms are overrepresented (enriched) among the selected genes. At this stage, the p-value for each GO term is also calculated, indicating the likelihood that the number of genes with this term is obtained by chance.

5. **Correction for Multiple Comparisons**. The necessity of this step is determined by the fact that given the large number of tests performed in GO analysis, correction is needed to avoid false positive results. At this stage, p-value correction was performed using the Benjamini-Hochberg test.

6. **Interpretation and Visualization of Results**. At this step, significant GO terms identified as enriched among the selected genes are evaluated and analyzed, relationships between different terms are analyzed, and network diagrams reflecting biological pathways or processes are created. Visualization of the results involves creating a network diagram of the most enriched GO terms.

7. **Formation of the List of Significant Genes Corresponding to the Most Significant GO Terms**. Formation of a subset of gene expression data containing significant genes as attributes for their further analysis and application in diagnostic systems of object states.

The following subsection presents the result of the practical implementation of this procedure on gene expression data using the functions and modules of the *topGO* and $org : Hs : eg : bd$ packages [5, 35] of the Bioconductor module [1].

2.4.1 Modeling the process of applying GO analysis to gene expression data to identify significant genes

The application of Gene Ontology (GO) analysis was simulated using gene expression data from patients undergoing evaluation for four distinct types of cancer: lung squamous cell carcinoma (LUSC) was identified in 502 patients, lung adenocarcinoma (LUAD) in 541, kidney renal clear cell carcinoma (KIRC) in 542, and brain lower grade glioma (LGG) in 534. The data, obtained through RNA sequencing (RNA-seq) within The Cancer Genome Atlas (TCGA) project, are freely available on the project's internet page [3]. Initially, the dataset included 2,119 samples and 19,947 genes. Following the exclusion of genes that were unexpressed across all samples (those with zero expression), the gene count decreased to 19,043. Annotation of gene identifiers by comparing with identifiers contained in databases corresponding to the human organism ising "org.Hs.eg.db" module [35] led to a reduction in the number of genes to 18,930. Genes not annotated in the database were removed.

In the next step, an ANOVA test was applied to the data to identify genes with high differential expression. The analysis of the test results showed that out of 18,930 genes, 17,803 have high differential expression (with a p-value criterion of less than 0.01). In the subsequent phase, a *topGOdata* object was constructed, encapsulating gene identifiers, respective scores, GO annotations, the hierarchical structure of GO, and other essential details needed for performing enrichment analysis on relevant genes.

The enrichment assessment utilizes statistical tests: Fisher's test, which relies on the gene count, and the Kolmogorov-Smirnov test (KS), focusing on gene score-based enrichment calculation. Within the scope of our research, both tests were applied to increase objectivity. Figure 2.9 depicts a distribution diagram of the ten most significant ontologies. On the X-axis, the ratio of the number of genes belonging to a specific category (GO) in the list of identified genes to the total number of genes in that category is plotted. A high gene ratio may indicate that a particular GO category is significant in the context of the research.

Figure 2.10 depicts a directed graph of the interactions among the 20 most significant ontologies (represented as rectangles, with the color saturation indicating the degree of significance). Inside the rectangles, the number of genes corresponding to each ontology is also indicated. The analysis of the diagram allows concluding about the complex nature of the relationships between ontologies and genes. Moreover, the results of the modeling showed differences in the outcomes when applying the Fisher and Kolmogorov-Smirnov tests. For this reason, the results of both tests were used

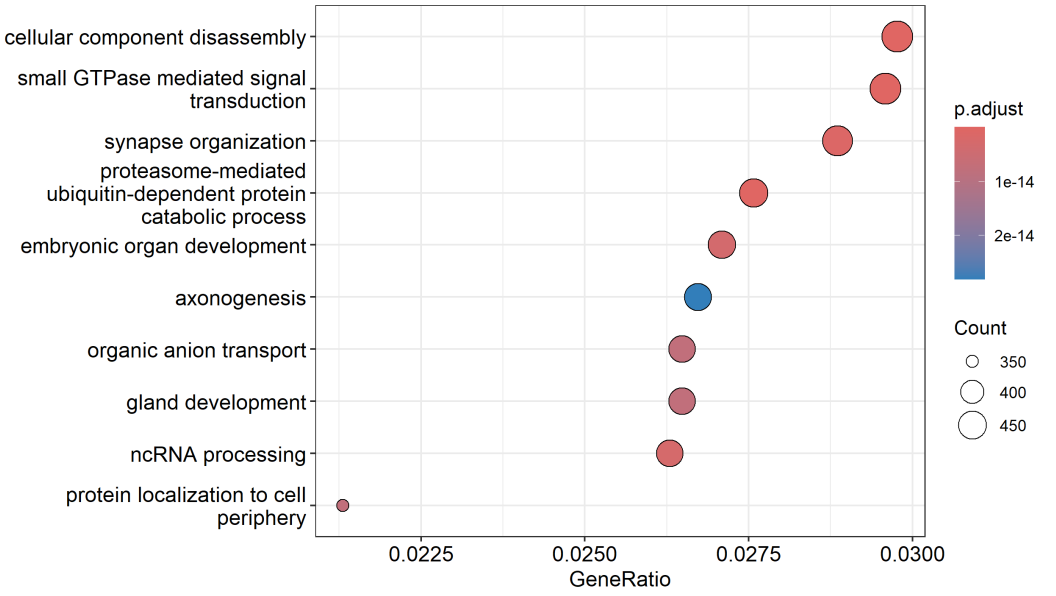

Figure 2.9: Distribution diagram of the ten most significant ontologies

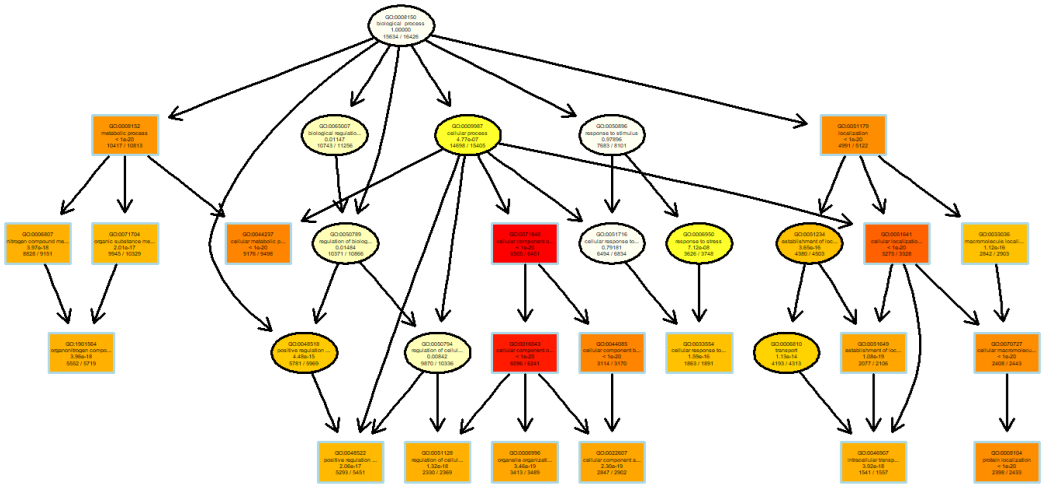

Figure 2.10: Network of interactions of the twenty most significant ontologies.

in forming the final list of significant genes. At the last step, a comprehensive list of significant genes according to both tests was created, unique genes were highlighted, and new data were formed, which included the identified significant genes as attributes. At this point, the gene count was diminished to 14,488.

The adequacy of the model was assessed by applying a classifier to the formed data. The classification results are presented in Table 2.6.

Table 2.6: Results of data classification based on the identification of significant genes via GO analysis application

Class	Prediction				Precision	Recall	F1-score	Accuracy
	kirc	lgg	luad	lusc				
kirc	162	1	1	0	0.988	1.000	0.994	97.9%
lgg	0	159	0	0	1.000	0.994	0.997	
luad	0	0	155	7	0.957	0.957	0.957	
lusc	0	0	6	143	0.960	0.953	0.957	

Analysis of the obtained results indicates high efficiency of the method based on GO analysis. Out of 619 samples that made up the test data subset, only 15 were identified incorrectly. The accuracy of the classification is 97.6%, which is quite high for this type of data. High values of precision, recall, and F1-score, which determine the quality of sample distribution into separate classes, are also quite high.

However, it should be noted that the number of genes remains quite large. Moreover, making decisions regarding the state of the object based on a large database has a high degree of subjectivity. Objectivity can be increased in this case by parallelizing the information processing flow through the use of cluster or bicluster analysis. At each level, the list of significant genes should be formed using GO analysis. This issue is addressed in the following sections of this thesis.

Chapter 3

Applying Cluster and Bicluster Analysis to Form Subsets of Co-Expressed Gene Expression Data

This chapter contains parts of the papers [19, 31, 20, 15, 21, 12, 11, 25, 24, 14].

3.1 Introduction

As shown by the simulation results presented in the previous chapter, the application of the gene expression profile reduction technique using statistical criteria and Shannon entropy can allow us to form a subset of the significant gene expression profiles. The number of genes, in this instance, is determined by the threshold value of a comprehensive criterion, the value of which is empirically defined during the simulation process, considering the target number of genes required for further research. However, it should be noted that the proposed technique only reduces the number of gene expression profiles by removing the non-informative ones. The next step is to form subsets of co-expressed gene expression profiles by applying gene expression profile clustering or/and biclustering algorithms and classification techniques to objects, which include allocated gene expression data as attributes at the model validation stage. The structural diagram of this procedure is shown in Figure 3.1.

As can be seen from Figure 3.1, the implementation of this process involves three stages. In the first stage, the model is configured by implementing the following steps: selecting the metric for assessing the closeness of gene expression profiles,

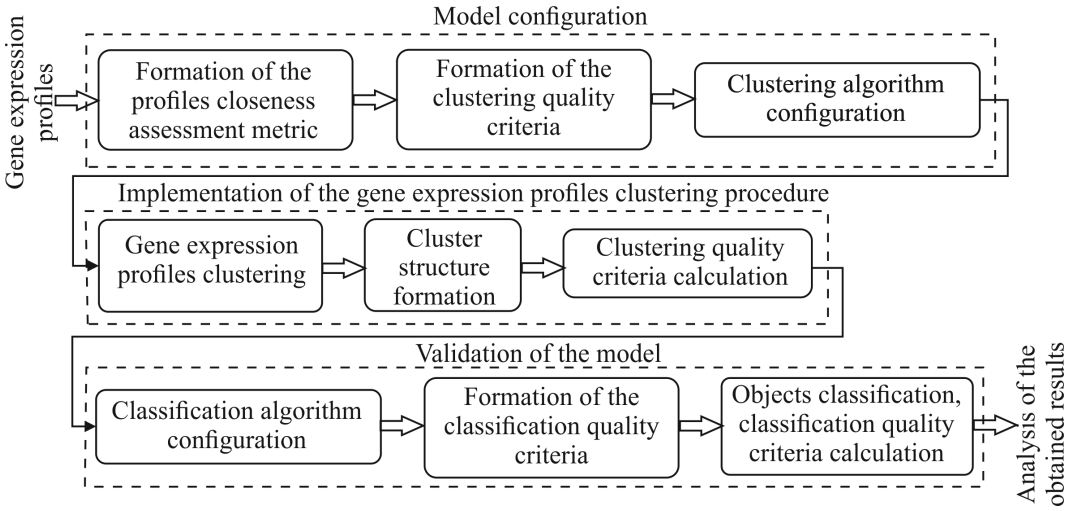

Figure 3.1: Structural diagram of the process for forming clusters of co-expressed gene expression profiles and model validation evaluation

clusters, and gene expression profiles and clusters; forming criteria for evaluating the quality of the cluster structure and applying the clustering algorithm, forming the structure of the clusters; and selecting and configuring the classification algorithm, considering the type of data being studied.

As noted in previous sections, a gene expression profile is a vector of expression values determined for various objects under study. A characteristic feature of this type of data is its high dimensionality, which makes standard proximity metrics such as Euclidean distance, Manhattan metric, Minkowski metric, etc., which underpin most clustering analysis algorithms, ineffective when applied to experimental data like gene expression profiles. In this case, correlation distance has a higher effectiveness in terms of discriminating ability, but studies presented in [27] have shown that it is inferior to proximity metrics based on mutual information assessment using various methods for estimating Shannon entropy. However, it should be noted that metrics based on mutual information analysis can also contradict each other, indicating that research in this area is not complete and requires further refinement.

The second step in the model adjustment phase for forming subsets of mutually expressed gene expression profiles involves formulating criteria for assessing the quality of clustering profiles. These criteria should consider both the grouping of gene expression profiles into separate clusters and the positioning of clusters relative to each other in feature space. Moreover, the number of clusters should also impact the value of the clustering quality criterion. Currently, there are many criteria that fully or partially include the aforementioned parameters as components. However, as studies presented in [25] have shown, applying the same clustering quality cri-

teria to different data can yield conflicting results, and different data may require different clustering criteria. A justified selection of the criterion or group of criteria for assessing the quality of the cluster structure, considering the type, is also one of the unresolved problems in modern data science.

The third step in the model adjustment procedure, presented in Figure 3.1, involves selecting a clustering algorithm, considering the features of gene expression profiles and tuning it by optimizing the algorithm's parameters. Most existing clustering algorithms are focused on low-dimensional data processing and are not effective for high-dimensional gene expression profiles. Furthermore, the parameter tuning model for the algorithm in the case of clustering gene expression profiles should be hybrid and include a classification procedure for objects that contain clusters of gene expression data as attributes. The optimal parameters of the clustering algorithm should correspond to the extrema of the relevant classification quality criteria for the objects under study.

The second stage in the process of forming subsets of mutually expressed gene expression profiles involves implementing the clustering procedure in the first step, forming the cluster structure in the second step, and calculating the quality criteria of the formed cluster structure in the third step. It should be noted that the complexity of the task of forming an optimal cluster structure is determined by the large number of gene expression profiles in the first stage on the one hand and the unknown target number of clusters on the other hand. Therefore, in this case, different cluster structures with varying numbers of clusters can be formed, which almost do not differ from each other in terms of clustering quality criteria. For this reason, the third stage of the procedure depicted in Figure 3.1 is very important: the validation of the model by applying a classifier to objects that contain gene expression data in the formed clusters as attributes.

The implementation of this stage involves, in the first step, selecting a classifier, considering the features of the data under study and tuning the classifier hyperparameters. In the second step, criteria for the quality of object classification, which are applied within the model, are formulated. In the third step, the classification procedure is performed with the calculation of classification quality criteria for the obtained cluster structures. Analyzing the obtained results can allow us to final form subsets of mutually expressed gene expression profiles for further research.

3.2 Forming a Metric for Assessing the Degree of Proximity of Gene Expression Profiles

As noted above, the formation of a metric for assessing the degree of proximity of gene expression profiles, clusters, and profiles within clusters is one of the essential stages, the successful implementation of which significantly influences the character

of the cluster structure formation and the assessment of its quality when applying the corresponding clustering quality criteria. Within the framework of the research, a proximity metric for gene expression profiles is used based on mutual information analysis with the application of various methods for calculating Shannon entropy [70]. This choice is determined by the following: as is known, the gene expression value is proportional to the quantity of appropriate gene type performing the corresponding functions in a biological organism. Thus, the gene expression value can be associated with the amount of information that determines the functional capabilities of the biological organism. Mutually expressed gene expression profiles in terms of information amount have a high level of similarity, i.e., minimal distance.

The formal definition of mutual information between two gene expression profiles E_i and E_j can be represented as follows:

$$MI(E_i, E_j) = \sum_{e_i \in E_i} \sum_{e_j \in E_j} P(e_i, e_j) \log \left(\frac{P(e_i, e_j)}{P(e_i)P(e_j)} \right) \tag{3.1}$$

where $P(e_i)$ and $P(e_j)$ are the marginal probability distributions of the gene expression values e_i and e_j respectively; $P(e_i, e_j)$ is the joint probability distribution of the expression values e_i and e_j.

Mutual information can be calculated using Shannon entropy formulas:

$$MI(E_i, E_j) = H(E_i) + H(E_j) - H(E_i, E_j) \tag{3.2}$$

where Shannon entropies $H(E_i)$, $H(E_j)$ and the joint entropy $H(E_i, E_j)$ can be calculated as follows:

$$H(E) = -\sum_{i=1}^{m} P(e_i) \log_2 P(e_i) \tag{3.3}$$

$$H(E_i, E_j) = -\sum_{i=1}^{m} \sum_{j=1}^{m} P(e_i, e_j) \log_2 P(e_i, e_j) \tag{3.4}$$

where m is the length of the gene expression profile or the number of samples constituting the experimental database.

It should be noted that Mutual information is a measure of shared information between two vectors of random variables, but it is not in itself a distance metric. Transforming the value of mutual information into a distance can be achieved in various ways. Within the scope of the research, a metric based on Shannon entropy is applied:

$$d(X, Y) = H(X) + H(Y) - 2MI(X, Y) \tag{3.5}$$

In this case, if considering two identical data distributions, then $H(X) = H(Y) = MI(X, Y)$ and $d(X, Y) = 0$. As the difference between data distributions increases,

the value of mutual information decreases, leading to an increase in the distance between these vectors.

Below is presented a list of methods for calculating Shannon entropy that were used withon the framework of the research for calculating distance based on mutual information:

- **Maximum Likelihood method (ML)**. This method is the simplest and is based on the assumption that each gene expression value can take on n values. Thus, in the first step, the gene expression profile is discretized, with the number of intervals n determined empirically, considering the type and size of the experimental data. The probability of a gene expression value e_i falling into the i-th interval is determined by the standard formula:

$$P(e_i) = \frac{n_i}{n}$$

where n_i is the frequency of occurrence of the i-th event or the number of gene expression values belonging to the i-th interval. Shannon entropy in this case is calculated by the standard formula:

$$H^{ML}(E) = -\sum_{i=1}^{n} \frac{n_i}{n} \log_2 \left(\frac{n_i}{n} \right) \tag{3.6}$$

- **James-Stein shrinkage estimator (JS)** [46]. This method is based on the combined use of two models: a high-dimensional model with high variance and low bias, and a low-dimensional model with low variance and high bias. The method involves dividing the gene expression profile into k equal cells. The probability of a gene expression value in the i-th cell is calculated by the formula:

$$P_i^{JS} = \lambda P_i + (1 - \lambda) P_i^{MP} \tag{3.7}$$

where $P_i = \frac{1}{n_i}$ is the target probability in the i-th cell (assuming that all gene expression values in this cell differ from each other); n_i is the number of gene expression values in the i-th cell; P_i^{MP} is the probability in the i-th cell calculated by the maximum likelihood method; λ is the shrinkage intensity parameter, whose value is constant for all cells and is calculated as:

$$\lambda = \frac{1 - \sum_{i=1}^{k} (P_i^{ML})^2}{(n-1) \sum_{i=1}^{k} (P_i - P_i^{ML})^2} \tag{3.8}$$

where n is the length of the gene expression profile (number of gene expression values). Shannon entropy in this case is also calculated by the standard

formula 3.3, using the probability value 3.7:

$$H^{JS}(E) = -\sum_{i=1}^{k} P_i^{JS} \log_2 \left(P_i^{JS} \right) \tag{3.9}$$

- **Jeffreys (JF), Laplace (LP), and Minimax (MM) methods** are based on the Bayesian method [8]. When applying the Bayesian method, the gene expression profile is also divided into k equal cells, followed by the probability estimation in each cell using the standard method ($\pi_i = \frac{n_i}{n}$) and forming the probability vector: $\pi = \{\pi_i\}, i = 1, \ldots, k, \sum_{i=1}^{k} \pi_i = 1$. It is assumed that the probability distribution character corresponds to the Dirichlet distribution and depends on the choice of the concentration parameter α:

$$p(\pi) \propto \prod_{i=1}^{k} \pi_i^{\alpha-1} \tag{3.10}$$

In [8], the authors obtained a formula for calculating the conditional probability considering the Dirichlet distribution for different concentration parameter values:

$$p_i(e|\alpha) = \frac{n_i + \alpha_i}{n + \sum_{i=1}^{k} \alpha_i} \tag{3.11}$$

The concentration parameters α are discrete and can take on specific values corresponding to different methods of estimating Shannon entropy. Thus, for $\alpha = \frac{1}{2}$, we get the Jeffreys method; for $\alpha = 1$, we have the Laplace method; the value $\alpha = \frac{\sqrt{n}}{k}$ corresponds to the minimax method of estimating Shannon entropy. In general, the formula for calculating Shannon entropy using the Bayesian method can be represented as follows:

$$H(e|\alpha) = -\sum_{i=1}^{k} p_i(e|\alpha) \log_2 \left(p_i(e|\alpha) \right) \tag{3.12}$$

3.3 Forming Criteria for Assessing the Quality of Cluster Structure

Forming quality criteria for assessing the cluster structure was carried out considering the principles of objective clustering inductive technology (OCIT) [64, 49, 19, 31, 20, 15, 21, 11], the application of which involves evaluating the cluster structure based on both internal and external clustering quality criteria. The final decision regarding the choice of the optimal cluster structure is made based on the analysis

of the balance criterion value, which includes both internal and external criteria as components. The application of OCIT involves dividing gene expression profiles in the first step into two equivalent subsets using proximity metrics for the profiles, as proposed in the previous subsection. These subsets consist of an equal number of pairwise close gene expression profiles.

Internal clustering quality criteria should take into account both the distribution of gene expression profiles within clusters relative to the median (since the average value of all expression values is an abstraction and does not correspond to the actual distribution of expression values in the profile) and the distribution of the clusters (medians of the respective clusters) in feature space. Within the framework of the current research, the first component of the internal criterion was calculated as the root mean square value of the distances from gene expression profiles to the cluster median where these profiles are located:

$$CW = \frac{1}{m}\sqrt{\sum_{k=1}^{K}\sum_{i=1}^{m_k} d(e_i, M_k)^2} \tag{3.13}$$

where $d(e_i, M_k)$ is the distance calculated based on mutual information assessment; e_i is the vector of expression values of the i-th gene; M_k is the median of the k-th cluster; m is the total number of gene expression profiles; m_k is the number of gene expression profiles in cluster k; K is the number of clusters.

Obviously, a smaller value of the internal criterion component (3.13) corresponds to a higher density of gene expression profiles in the clusters. The second component of the internal criterion was calculated as the root mean square value of the distances between all pairs of clusters' medians:

$$CB = \frac{2}{K(K-1)}\sqrt{\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} d(M_i, M_j)^2} \tag{3.14}$$

It should be noted that better clustering corresponds to a smaller distance between gene expression profiles within separated clusters and a larger distance between clusters (maximum value of criterion (3.14)). Considering the above, the formula for calculating the internal criterion can be represented as follows:

$$QC_{\text{int}} = \frac{CW}{CB} \tag{3.15}$$

Formula (3.15) defines the average density distribution of gene expression profiles and the medians of the respective clusters. A smaller value of criterion (3.15) corresponds to better clustering according to this criterion.

The external criterion involves the presence of two equivalent subsets of gene expression profiles. The relevance of applying this criterion is determined by the

reproducibility error, which is inherent in most existing data clustering algorithms. In other words, clustering results obtained on one dataset do not always repeat within an acceptable error range when using another equivalent subset of data. Obviously, if the distribution nature of gene expression profiles in equivalent data subsets is close, the values of the internal criteria (3.15) obtained on these subsets should not significantly differ from each other. As the reproducibility error increases, the discrepancy between the values of the internal criteria will grow. Considering the above, within the framework of the research, the value of the external criterion was calculated as the normalized difference of the internal criteria values obtained on equivalent data subsets A and B:

$$QC_{\text{ext}} = \frac{|\ QC^A_{\text{int}} - QC^B_{\text{int}}|}{QC^A_{\text{int}} + QC^B_{\text{int}}} \tag{3.16}$$

A smaller value of this criterion corresponds to a smaller discrepancy in clustering results obtained on equivalent data subsets. However, it should be noted that the values of the internal and external criteria may contradict each other. High values of the internal criteria (unsatisfactory clustering on equivalent subsets) can be close to each other, leading to a low value of the external criterion due to reproducibility error. In this case, applying the balance criterion, which includes both internal and external criteria as components, is appropriate.

The idea of using the external criterion was first proposed in [64, 49] and further developed in [19, 31, 20, 15, 21, 11]. The balance criterion was calculated based on the Harrington desirability function, presented in subsection 2.3 of the thesis (Figure 2.6). One of the significant advantages of the Harrington method is the absence of the need to normalize input data vectors while maintaining high objectivity in calculating the output parameter. Normalization of input data occurs automatically at the stage of transforming the scales of input parameters into the scale of the dimensionless parameter Y, whose range of values is $\langle -2, 5\rangle$. The algorithm for calculating the balance criterion includes the following stages:

1. Calculation of coefficients a and b in the linear equations for transforming the values of the respective criteria into the value of the parameter Y, considering the boundary values of the respective criteria (assuming that the scales of the criteria values are linear):

$$\begin{aligned} Y_{min} &= a - b \cdot QC_{max} \\ Y_{max} &= a - b \cdot QC_{min} \end{aligned} \tag{3.17}$$

   where Y_{min}, Y_{max}, QC_{min}, and QC_{max} are the minimum and maximum values of parameter Y, the internal and external criteria, respectively.

2. Transforming the values of the internal and external criteria into the values of the parameter Y:

$$\begin{aligned} Y_{int}^{A} &= a_{int}^{A} - b_{int}^{A} \cdot QC_{int}^{A} \\ Y_{int}^{B} &= a_{int}^{B} - b_{int}^{B} \cdot QC_{int}^{B} \\ Y_{ext} &= a_{ext} - b_{ext} \cdot QC_{ext} \end{aligned} \tag{3.18}$$

3. Calculation of the private desirability values for each criterion:

$$\begin{aligned} d_{int}^{A} &= \exp(-\exp(-Y_{int}^{A})) \\ d_{int}^{B} &= \exp(-\exp(-Y_{int}^{B})) \\ d_{ext} &= \exp(-\exp(-Y_{ext})) \end{aligned} \tag{3.19}$$

4. Calculation of the balance criterion as the geometric average of the private desirabilities:

$$QC_{bal} = \sqrt[3]{d_{int}^{A} \cdot d_{int}^{B} \cdot d_{ext}} \tag{3.20}$$

A higher value of the balance criterion corresponds to better clustering according to the group of criteria used.

3.4 Validation of the Gene Expression Profiles Clustering Model

As can be seen from the structural diagram of the process for forming clusters of mutually expressed gene expression profiles, shown in Figure 3.1, the final step involves evaluating the adequacy of the gene expression profiles clustering model by applying the classification procedure to objects that contain gene expression values of the formed clusters as attributes. The main idea is that clusters containing gene expression profiles with a higher level of mutual expression should correspond to higher classification results for objects containing gene expression values of these profiles as attributes. In the research, the quality of object classification was assessed using traditional methods based on Type I and Type II errors with the application of the confusion matrix, the components of which, in the case of diagnosing the presence or absence of a corresponding disease in patients based on expression data analysis, are presented in Table 3.1.

The following criteria were applied for the evaluation of object classification quality:

1. **Classification Accuracy (ACC)** - determines the probability of correct identification of the objects under study:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.21}$$

Table 3.1: Confusion matrix for diagnosing the presence or absence of a disease

Predicted Class	Actual Class	
	Healthy	Sick
Healthy	TP	FP
Sick	FN	TN

where:

- TP (True Positive) – true positive cases;
- TN (True Negative) – true negative cases;
- FN (False Negative) – false negative cases (Type II error);
- FP (False Positive) – false positive cases (Type I error).

2. **F1-measure (F1)** - defined as the harmonic mean of Precision (PR) and Sensitivity or Recal (RC):

$$F1 = \frac{2 \cdot PR \cdot RC}{PR + RC} \tag{3.22}$$

where:

- PR is defined as the ratio of correctly identified positive values to the total number of values identified as positive:

$$PR = \frac{TP}{TP + FP} \tag{3.23}$$

- RC or TPR (True Positive Rate) is defined as the ratio of correctly identified positive values to the total number of actual positive values:

$$RC = TPR = \frac{TP}{TP + FN} \tag{3.24}$$

3. **Matthews Correlation Coefficient (MCC)** - a quality criterion for evaluating classification performance based on Type I and Type II errors:

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{3.25}$$

Higher values of the criteria (3.21), (3.22) and (3.25) correspond to higher classification quality of the objects under study.

3.4.1 Modeling the Process of Forming Clusters of Mutually Expressed Gene Expression Profiles

Algorithm 3 presents a stepwise procedure to form subsets of co-expressed gene expression profiles based on based on mutual information analysis. The simulation of the formation process of subsets of mutually expressed gene expression profiles within the framework of objective clustering inductive technology based on mutual information evaluation using different Shannon entropy calculation methods was carried out using gene expression data from patients studied for lung cancer (GSE19188 [2, 47]).

Initially, a subset of significant genes was formed based on statistical criteria and Shannon entropy using a methodology based on the Harrington desirability function. In this process, 588 gene expression profiles were identified that correspond to a generalized desirability index value > 0.2 (gene expression profiles with values less than 0.2 correspond to undesirable desirability according to the above criteria).

In the second step, based on the set of gene expression profiles formed, two equivalent subsets were formed using the iterative algorithm. It should be noted that five mutual information evaluation metrics were used in the study (based on the estimation methods: Maximum Likelihood (ML), Jeffreys (JF), Minimax (MM), Laplace (LP), and James-Stein (JS)). Thus, five pairs of equivalent subsets of gene expression profiles were formed, with 294 profiles in each data subset.

Preliminary quality assessment of the formed data was carried out by applying the binary classifier "Random Forest" to objects containing the identified gene expression profiles as attributes and calculating the classification quality criteria (3.21) – (3.25). The classifier was applied to the full data set and to each of the equivalent subsets.

The set of objects under study was divided into two subsets: 60% of the objects were used to train the model, and 40% to test it.

The results of the simulation using the test data subsets are presented in Table 3.2. The analysis of the obtained results allows us to conclude that the classification accuracy criteria values when using the full set of gene expression profiles (588) are relatively high. Only one object was misidentified when using the test subset of objects (62 out of 156). This fact indicates the adequacy of the proposed model for forming informative gene expression profiles based on the comprehensive application of statistical criteria and Shannon entropy. It should be noted that the process of forming mutually expressed gene expression profiles has not yet been implemented.

The analysis of the classification results of objects based on gene expression data contained in equivalent subsets also allows us to conclude a small discrepancy in the results. It should be noted that when using the Shannon entropy calculation methods "ML", "MM", and "JS", the results are identical. In these cases, only 1 and 3 objects out of 62 were misidentified in the first and second subsets, respectively.

Algorithm 3: Algorithm for Forming Subsets of Mutually Expressed Gene Expression Profiles Based on Mutual Information Analysis

Data: Gene expression data $e = (e_{i,j}); i = 1, \ldots, n; j = 1, \ldots, m$
Stage I: Model Setup
begin
- **1.1** Form a vector of proximity metrics of gene expression profiles: $MI = (MI_k), k = 1, \ldots, r$;
- **1.2** Form functions to calculate clustering quality criteria;
- **1.3** Choose clustering algorithm, initialize range and step for parameter variation: $p = q_0, \ldots, q;\ dp.$;

Stage II: Clustering Gene Expression Profiles
begin
- **2.1** Choose the first proximity metric: $k = 1$;
- Form equivalent subsets A and B;
- **begin**
  - **2.1.1** Form triangular distance matrix between all pairs of vectors;
  - **2.1.2** Select the pair of gene expression profiles with the minimum distance: $d(e_p, e_s) = \min(d_{ij})$;
  - **2.1.3** Assign profile e_p to subset A and profile e_s to subset B;
  - **2.1.4** Continue the above procedure for other pairs of profiles;
  - **2.1.5** If the number of profiles is odd, assign the last profile to both subsets;
- **2.3** Initialize the first parameter of the clustering algorithm: $p = q_0$;
- **while** $p \leq q$ **do**
  - **2.4** Cluster the gene expression profiles allocated into subsets A and B;
  - **2.5** Calculate internal and external clustering quality criteria;
  - **2.6** Increment the clustering algorithm parameter by dp ($p = p + dp$);
- **2.7** Calculate the balance criterion;
- **2.8** Analyze the results, select the optimal clustering;

Stage III: Classification Procedure Implementation
begin
- **while** $k \leq r$ **do**
  - **3.1** Choose the data classification algorithm, set its parameters;
  - **3.2** Classify objects using gene expression data from identified clusters;
  - **3.3** Calculate classification quality criteria;
  - **3.4** Increment k and return to step 2.2;

Stage IV: Analysis of Results
begin
- **4.1** Analyze the results. Form the final decision.

Result: Clusters of co-expressed gene profiles

Table 3.2: Classification Results of Objects Based on Full Gene Expression Data and Gene Expression Values in Equivalent Subsets

Experemental data	Classification quality criteria				
	ACC	SE	SC (Specificity)	F1-measure	MCC
init_data	0.984	1.000	0.973	0.992	0.967
ML_data_1	0.984	0.963	1.000	0.973	0.968
ML_data_2	0.952	0.926	0.971	0.939	0.902
MM_data_1	0.984	0.963	1.000	0.973	0.968
MM_data_2	0.952	0.926	0.971	0.939	0.902
JF_data_1	0.968	0.962	0.972	0.965	0.934
JF_data_2	0.952	0.926	0.971	0.939	0.902
JS_data_1	0.984	0.963	1.000	0.973	0.968
JS_data_2	0.952	0.926	0.971	0.939	0.902
LP_data_1	0.968	0.962	0.972	0.965	0.934
LP_data_2	0.952	0.926	0.971	0.939	0.902

The results are also identical when using the criteria based on the "JF" and "LP" methods, with 2 and 3 objects out of 62 misidentified in the first and second subsets of gene expression profiles, respectively.

The difference in the number of misidentified objects when using equivalent subsets of gene expression profiles can be explained by the fact that in the iterative process, the degree of proximity of the profiles decreases with an increase in the iteration number. This fact may lead to discrepancies in the classification results of objects containing mutually non-expressed gene expression profiles as attributes. However, it should be noted that in all cases, the quality of the object classification process according to the criteria used is quite high.

3.4.2 Inductive Model for the Formation of Clusters of Mutually Expressed Gene Expression Profiles Based on the Spectral Clustering Algorithm

The choice of the spectral clustering (SC) algorithm is determined by the fact that the SC algorithm is currently one of the modern clustering algorithms that allows the identification of clusters of various shapes. The practical implementation of the SC algorithm can be effectively implemented using standard computational mathematics and linear algebra methods. The results of the practical implementation of the spectral clustering algorithm within the inductive model, presented by Algorithm 3, using different proximity metrics of gene expression profiles are presented in Tables 3.3 - 3.6. It should be noted that increasing the number of clusters led to the appearance of a larger number of small clusters, which significantly worsened the clustering results according to the criteria used. For this reason, the clustering

results for a larger number of clusters are not provided.

Table 3.3: Results of the simulation on the application of a inductive model for forming mutually expressed gene expression profiles based on the spectral clustering algorithm (ML and JS proximity metrics)

Number of clusters	Number of genes		Internal criterion		External criterion	Balance criterion
2	A	B	A	B	0.036	0.993
	201	107	0.069	0.074		
	93	187				
3	A	B	A	B	0.394	0.001
	15	89	0.191	0.083		
	93	18				
	186	187				

Table 3.4: Results of the simulation on the application of a inductive model for forming mutually expressed gene expression profiles based on the spectral clustering algorithm (MM proximity metrics)

Number of clusters	Number of genes		Internal criterion		External criterion	Balance criterion
2	A	B	A	B	0.043	0.993
	206	106	0.068	0.074		
	88	188				
3	A	B	A	B	0.395	0.001
	19	88	0.194	0.084		
	88	18				
	187	188				

Tables 3.7 - 3.9 present the classification results of objects based on gene expression data in the identified clusters using different proximity metrics of gene expression profiles.

The analysis of the obtained results allows us to conclude that models based on the use of ML, MM, and JS metrics show almost identical results (a slight deviation in the criteria values (0.001) in the table is not considered), with the classification results of objects according to all criteria in the three clusters being quite high. Out of 62 objects, only two were incorrectly identified based on gene expression data in the first cluster of the first equivalent subset of gene expression profiles and the second cluster of the second equivalent subset of data, and one based on gene expression data in the first cluster of the second equivalent subset of data.

When using JF and LP metrics, the results differ slightly, but it should be noted

Table 3.5: Results of the simulation on the application of a inductive model for forming mutually expressed gene expression profiles based on the spectral clustering algorithm (JF proximity metrics)

Number of clusters	Number of genes		Internal criterion		External criterion	Balance criterion
2	A	B	A	B	0.038	0.993
	206	107	0.068	0.073		
	88	187				
3	A	B	A	B	0.388	0.001
	20	89	0.190	0.084		
	88	18				
	186	187				

Table 3.6: Results of the simulation on the application of a inductive model for forming mutually expressed gene expression profiles based on the spectral clustering algorithm (JF proximity metrics)

Number of clusters	Number of genes		Internal criterion		External criterion	Balance criterion
2	A	B	A	B	0.036	0.993
	204	106	0.068	0.073		
	90	188				
3	A	B	A	B	0.375	0.001
	16	88	0.189	0.086		
	90	18				
	188	188				

Table 3.7: Classification results of objects based on gene expression data in the corresponding clusters using ML, MM, and JS data

Experemental data	Classification quality criteria				
	ACC	SE	SC (Specificity)	F1-measure	MCC
Data_1, CL_1	0.974	0.969	0.978	0.972	0.947
Data_1, CL_2	0.949	0.938	0.956	0.944	0.895
Data_2, CL_1	0.987	0.985	0.989	0.986	0.974
Data_2, CL_2	0.974	0.969	0.978	0.972	0.947

that in all cases, the four identified clusters contain different gene expression profiles, with the classification results being quite high. This fact indicates the adequacy of the model for forming a set of informative gene expression profiles at the data preprocessing stage. It should also be noted that forming a conclusion about the patient's condition (sick, healthy) based on both the full set of gene expression data

Table 3.8: Classification results of objects based on gene expression data in the corresponding clusters using JF data

Experemental data	Classification quality criteria				
	ACC	SE	SC (Specificity)	F1-measure	MCC
Data_1, CL_1	0.974	0.969	0.978	0.972	0.947
Data_1, CL_2	0.942	0.924	0.956	0.933	0.882
Data_2, CL_1	0.936	0.923	0.945	0.929	0.868
Data_2, CL_2	0.974	0.955	0.989	0.965	0.948

Table 3.9: Classification results of objects based on gene expression data in the corresponding clusters using LP data

Experemental data	Classification quality criteria				
	ACC	SE	SC (Specificity)	F1-measure	MCC
Data_1, CL_1	0.974	0.969	0.978	0.972	0.947
Data_1, CL_2	0.942	0.924	0.956	0.933	0.882
Data_2, CL_1	0.981	0.970	0.989	0.974	0.961
Data_2, CL_2	0.981	0.970	0.989	0.974	0.961

and the gene expression data in one of the clusters is quite subjective. Increasing objectivity, in this case, can be achieved through the comprehensive application of object classification results to analyze the classification results of different groups of gene expression profiles. Within the framework of the research, the final decision on the patient's state was made based on the analysis of the classification results of patients using an odd number of clusters corresponding to the highest values of all three object classification quality criteria (ACC, F1-measure, MCC). In the current simulation, three clusters of gene expression profiles were identified, and the final decision was made based on the alternative voting; that is, if the current object was identified as healthy (sick) based on the use of three or two groups of gene expression profiles, it was identified as healthy (sick).

The simulation results showed that in all cases (when using different Shannon entropy calculation methods), the classification results of objects based on gene expression data are the same: classification accuracy is 0.974 (2 objects out of 62 are incorrectly identified), with a classification error of 3.22%; the F1-index and MCC criteria values are 0.972 and 0.947, respectively. It should be noted that the choice of the type of mutual information evaluation metric in the process of making a collegial decision using the clustering results of gene expression profiles is not decisive.

3.5 Application of Bicluster Analysis for the Formation of Subsets of Coherent Gene Expression Data

As noted in the introduction, bicluster analysis is a method that allows simultaneous clustering of rows and columns of a data matrix. The difference from traditional clustering is that ordinary clustering groups only rows (or columns) based on their similarity. In contrast, bicluster analysis allows for the identification of mutually correlated subsets of rows and columns. The features of bicluster analysis include:

- Unlike global clustering, where groups of rows or columns of the data matrix are formed and are similar across the entire data set, bicluster analysis forms local patterns in data subsets, i.e., it identifies subsets of mutually similar rows and columns based on a specific similarity metric.
- Biclusters can have different sizes and shapes, which allows for the discovery of various relationships within the data being studied.
- Different biclusters can overlap, meaning that one row or column can belong to multiple biclusters simultaneously. This can reveal more complex dependencies within the data being studied.

The above features suggest the following applications for bicluster analysis in processing gene expression data:

1. **Identification of genetic modules:** It is possible to identify subsets of genes that are co-regulated under certain conditions. This may indicate shared regulatory mechanisms or participation in common biological processes.
2. **Condition analysis:** Allows for the identification of specific conditions (e.g., certain disease states or treatment responses) under which a particular group of genes shows co-activity.
3. **Noise analysis:** Gene expression data are often noisy. Bicluster analysis can be less sensitive to noise as it focuses on local patterns.
4. **Variability analysis:** Since genes can be regulated differently under various conditions, bicluster analysis allows for the examination of this variability, which may be overlooked when using classical clustering algorithms.

It should be noted, however, that the results of existing biclustering algorithms are determined by a set of hyperparameters, the optimization of which is a necessary step for the successful application of biclustering algorithms to analyze and process gene expression data. This step, in turn, requires the formation of biclustering quality criteria, which can be used as the objective function to form the optimal vector of hyperparameters.

3.5.1 Forming Quality Criteria for Biclustering of Gene Expression Data

In general, the quality assessment criteria for bicluster structure can be divided into two groups:

1. *Internal criteria.* These criteria allow for the assessment of individual biclusters in the cluster structure without comparing them to reference biclusters, the formation of which, in the case of gene expression data, is problematic. The most common internal criteria include the following metrics:

   - Mean Squared Residue (MSR) [40, 84]. Measures the coherence of a bicluster or the degree of consistency or homogeneity of values within the bicluster. Ideally, a bicluster should contain very similar values that form a certain pattern or structure. This homogeneity can be defined as constancy, additivity, or multiplicative homogeneity. Constant coherence implies that all values in the bicluster are approximately the same. Additive coherence means the differences between values in rows or columns are approximately the same. Multiplicative coherence implies that the values in rows and/or columns are approximately the same after multiplication by a certain factor. The formula for calculating the MSR criterion for a bicluster with I rows and J columns is:

   $$MSR = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} (x_{ij} - \bar{x}_i - \bar{x}_j + \bar{x})^2 \tag{3.26}$$

   where: x_{ij} is the matrix element; $\bar{x}_i$, $\bar{x}_j$, and $\bar{x}$ are the average values of row i, column j, and the overall bicluster mean, respectively.

   A lower value of this criterion indicates increased bicluster coherence.

   - Volume. Calculated as the product of the number of rows and columns. A larger value of this criterion indicates that the bicluster reflects a more general data structure.
   - Variability. Measures the spread or variability of values within the bicluster and can be calculated using the classical formula for variance:

   $$V = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} (x_{ij} - \bar{x})^2 \tag{3.27}$$

   A lower value of variability usually indicates a more stable pattern in the bicluster.

- Connectivity. Measures the number of connections or relationships between elements in the bicluster and can be calculated by the formula:

$$Connectivity = \frac{Number\ of\ connections\ between\ elements\ in\ the\ bicl.}{Maximum\ possible\ number\ of\ connections} \tag{3.28}$$

It should be noted that the calculation of this criterion value assumes the reconstruction of the gene regulatory network at a previous step by applying a corresponding algorithm, where one of the key hyperparameters is the thresholding coefficient, which significantly affects the number of real connections between bicluster elements. Higher connectivity indicates stronger relationships between elements in the bicluster.

- Robustness. Assesses how stable the biclusters remain with minor changes in the input data, for example, when adding Gaussian "white" noise.

2. **External criteria.** External quality criteria for biclustering are based on comparing the biclustering results with a certain standard or external reference structure. Typically, such criteria are used in cases where additional (external) data about the data structure or the true distribution of biclusters are available. The most widely used external quality criteria for biclustering currently are the Rand Index (RI) and the Jaccard Index (JI).

- Rand Index. Determines the similarity between two biclusterings based on type I and II errors [42]:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.29}$$

where: TP is the number of object pairs correctly assigned to the same bicluster; TN is the number of object pairs correctly assigned to different biclusters; FP is the number of object pairs wrongly assigned to the same bicluster; FN is the number of object pairs wrongly considered to belong to different biclusters.

- Jaccard Index. Calculated as the ratio of common objects in two biclusters to the total number of objects in the biclusters (their union):

$$JI = \frac{|B1 \cap B2|}{|B1 \cup B2|} \tag{3.30}$$

where $B1$ and $B2$ are two biclusters. In the presence of more than two biclusters in the biclusterings and in the absence of overlap between biclusters, the formula for calculating the Jaccard Index becomes:

$$JI(BC1, BC2) = \frac{1}{n_1} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left(\frac{|B_i(BC1) \cap B_j(BC2)|}{|B_i(BC1) \cup B_j(BC2)|} \right) \tag{3.31}$$

where: n_1 and n_2 are the number of biclusters in biclusterings $BC1$ and $BC2$ respectively. In the presence of overlap between biclusters, formula (6) is refined as follows:

$$JI_{corrected}(BC1, BC2) = \frac{JI(BC1, BC2)}{\max(JI(BC1, BC1), JI(BC2, BC2))} \quad (3.32)$$

It should be noted that in the absence of overlap between biclusters, formula (7) transforms into formula (6), as the denominator in formula (7) will equal one. External criteria, in contrast to internal ones, necessitate extra insights into the data's architecture or the precise (reference) distribution of biclusters. This requirement presents challenges when gene expression data are used as experimental material. Typically, these criteria are employed to assess the efficacy of different biclustering techniques.

3.5.2 The Internal Criterion of Biclustering Quality Based on the Assessment of Mutual Information

As noted above, biclustering is the process of simultaneously clustering rows and columns of a matrix. In the context of gene expression data analysis, the experimental data is represented as a matrix where the rows are genes, and the columns are experimental conditions, or vice versa, and the values in the matrix reflect the expression level of a gene under a certain condition, i.e., its expression. In this case, a bicluster defines a subset of genes that have similar expression profiles under a subset of conditions. One way to assess the quality of a bicluster is to apply mutual information (MI) analysis between the rows and columns. MI can indicate how much information in the rows and columns depends on each other, and therefore, a high MI value may indicate a high coherence of the bicluster.

As mentioned hereinbefore, mutual information is a measure of shared information between two vectors of random variables, but it is not inherently a distance metric. Transforming the MI value into a distance can be done in various ways. Within the framework of this research, a metric based on Shannon entropy is applied:

$$d(X, Y) = H(X) + H(Y) - 2 \cdot MI(X; Y) \quad (3.33)$$

Here, $H(X)$ and $H(Y)$ denote the Shannon entropy for vectors X and Y respectively, and $MI(X; Y)$ represents the mutual information between X and Y.

In scenarios where two data distributions are identical, it follows that $H(X) = H(Y) = MI(X; Y)$, rendering $d_{MI}(X, Y) = 0$. Conversely, as the divergence between the data distributions widens, the mutual information diminishes, which in turn increases the calculated distance d_{MI} between the vectors.

Estimating the internal measure for bicluster coherence involves computing the mean distance among both rows and columns within the bicluster. The methodology for deriving this measure encompasses the steps below:

1. Determine the mean distance across all row pairs within the bicluster:

$$QC_{rows} = \frac{2}{n_{rows} \times (n_{rows} - 1)} \sum_{i=1}^{n_{rows}-1} \sum_{j=i+1}^{n_{rows}} d(R_i, R_j) \tag{3.34}$$

2. Compute the mean distance across all column pairs within the bicluster:

$$QC_{columns} = \frac{2}{n_{columns} \times (n_{columns} - 1)} \sum_{i=1}^{n_{columns}-1} \sum_{j=i+1}^{n_{columns}} d(C_i, C_j) \tag{3.35}$$

3. Calculate the overall average of the criteria:

$$QC = \frac{QC_{rows} + QC_{columns}}{2} \tag{3.36}$$

The minimum value of the criterion 3.36 corresponds to the maximum coherence level of the bicluster. It should be noted that when applying any clustering algorithm to gene expression data, which is characterized by a large volume of data, a significant number of biclusters with low coherence values may emerge. These low-coherence biclusters do not allow for unambiguous identification of the class of samples under study. Moreover, the architecture of biclustering is largely determined by the parameters of the corresponding algorithm used to form the cluster structure. Therefore, there is also the problem of optimizing the algorithm parameters, for which the Bayesian optimization algorithm is used within the current research. The application of this algorithm involves the following steps:

1. Selection of the biclustering algorithm. Determination of the range of algorithm hyperparameters.

2. Selection of the Bayesian optimization algorithm model. A model based on Gaussian processes was used in the research.

3. Application of the Bayesian optimization algorithm using the selected model. Formation of the best combination of hyperparameters based on the formulated objective function.

4. Application of the biclustering algorithm with the optimal hyperparameter values to gene expression data. Formation of the bicluster structure.

5. Assessment of the coherence of the identified biclusters and formation of a subset of biclusters with high coherence values for further research.

3.5.3 Evaluation of the Effectiveness of Internal Criteria for Biclustering Quality Using Artificial Biclusters

The evaluation of the effectiveness of biclustering internal quality criteria was carried out using artificial data, which contained five non-overlapping coherent biclusters of the same size but with different degrees of coherence (Figure 3.2). As can be seen

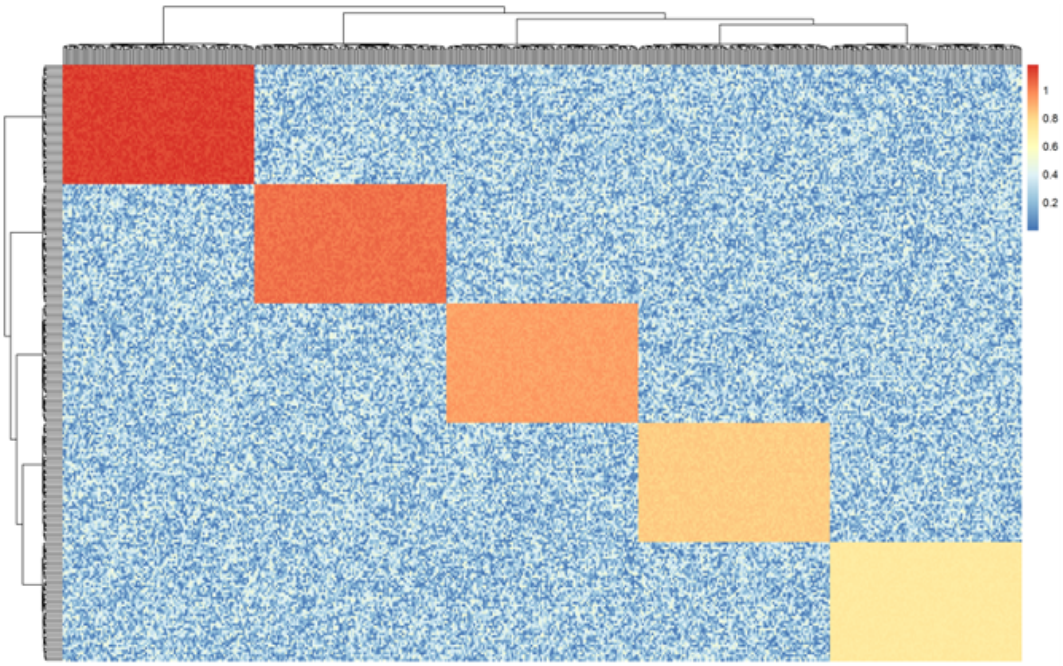

Figure 3.2: Heatmap of Bicluster Distribution in Synthetic Data

from the figure, the synthetic data contains five biclusters that can be identified by applying the appropriate biclustering algorithm. However, it should be noted that comparing the biclusters obtained from the biclustering algorithm with the perfect biclustering by calculating the respective biclustering external quality criteria is not objective in this case. This is because a large number of small biclusters may be identified (as confirmed by the simulation results), which can significantly affect the value of the external criterion.

Considering the above, the evaluation of the effectiveness of the corresponding biclustering internal quality criteria when applying synthetic data, presented in Figure 3.2, was carried out by comparing the values of the criteria calculated for the first five biclusters with the values of these criteria calculated for the perfect biclusters. The relative deviations of the corresponding criteria values were calculated as

follows:

$$QC_{rel} = \frac{|QC_{exp} - QC_{perf}|}{QC_{perf}} \tag{3.37}$$

where:

- QC_{exp} is the value of the corresponding biclustering quality criterion (MSR or MI) obtained during the application of the biclustering algorithm to the first five biclusters.
- QC_{perf} is the value of the criteria calculated for the perfect biclustering shown in Figure 3.2.

The simulation process was carried out in the R programming environment using the `biclust` package [53], which contains functions for applying various biclustering algorithms. Considering the research presented in [24], the current simulation process used the `ensemble` algorithm [53], the efficiency of which, based on the simulation results presented in [24], is significantly higher compared to other biclustering algorithms. The result of the `ensemble` algorithm is determined by two parameters: the thresholding coefficient (`thr`) and the approximate ratio of the number of rows and columns in the biclusters (`simthr`). The simulation process involved changing the values of these parameters within a predetermined range with a certain step size, calculating the absolute values of the MSR and MI criteria and their relative values, determined by formula 3.37. The effectiveness of the criteria was evaluated by analyzing the convergence between absolute and relative values.

Algorithm 4 presents a stepwise procedure for determining the optimal hyperparameter of the ensemble biclustering algorithm using ordered grid search method. Figure 3.3 shows the simulation results of applying Algorithm 4 to synthetic data to determine the optimal value of the *thr* parameter of the ensemble biclustering algorithm. The analysis of the obtained results allows concluding that the values of the relative and absolute biclustering quality criteria change consistently, indicating the adequacy of MSR and MI-based metrics for forming the bicluster structure. The optimal value of the thr parameter, corresponding to the minimum criteria values, is 0.37.

Further simulation results led to the conclusion that for synthetic data, the value of the simthr parameter, when the thr value is fixed, does not affect the biclustering result. Figure 3.4 shows the result of applying the ensemble biclustering algorithm with optimal parameters (thr = 0.37, simthr = 0.3) to the synthetic data depicted in Figure 3.2. As can be seen, the cluster structure completely corresponds to that depicted in Figure 3.2. This fact indicates the adequacy of the used biclustering quality criteria.

Algorithm 4: Determination of Optimal Parameters for the Ensemble Biclustering Algorithm using Ordered Grid Search Method

Input: Initial values and ranges for hyperparameters thr and simthr

Stage I: Model Setup

1. Initialize intervals and step sizes for thr and simthr.
2. Calculate MSR and MI criteria values for perfect biclustering.

Stage II: Determine Optimal thr

1. Fix simthr value (initially 0.3).
2. Initialize thr with $thr_{\min}$.
3. **while** $thr < thr_{\max}$ **do**

   Apply ensemble biclustering method to synthetic data;

   Extract first five biclusters and calculate coherence;

   Calculate average coherence for each criterion;

   Calculate relative criterion for each metric;

   Increment thr by step size;

   end
4. Analyze results and fix optimal thr corresponding to the minimum quality criteria values.

Stage III: Determine Optimal simthr

1. Initialize simthr with $simthr_{\min}$.
2. **while** $simthr < simthr_{\max}$ **do**

   Apply steps 2.3 to 2.5;

   Increment simthr by step size;

   end
3. Analyze results and fix optimal simthr corresponding to the minimum quality criteria values.

Stage IV: Formation of Bicluster Structure and Result Analysis

1. Apply ensemble biclustering with optimal parameters to synthetic data.
2. Form bicluster structure and analyze results.

Output: Optimal values for thr and simthr

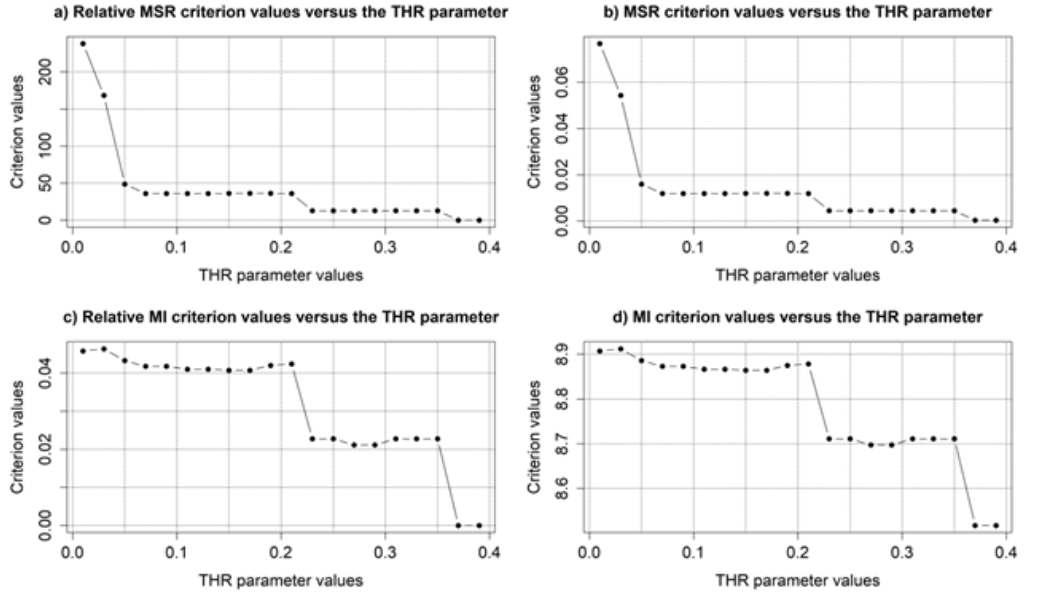

Figure 3.3: Simulation results for determining the optimal value of the *thr* parameter in the *ensemble* biclustering algorithm

3.5.4 Modeling to Determine the Optimal Parameters of the Biclustering Algorithm Using the Bayesian Optimization Algorithm

At this stage of simulation, gene expression data from patients studied for various types of cancer were used as experimental data. The data are freely available on the website of The Cancer Genome Atlas (TCGA) project and contain nine classes of samples, eight of which correspond to different types of cancer, and the ninth group of gene expression data corresponds to subjects for whom cancer was not detected. Overall, the initial data contained 3269 samples and 19947 genes. After removing non-expressed and weakly expressed genes for all samples, the number of genes was reduced to 19265. In the next step, co-expressed gene expression profiles were extracted from the data by applying the inductive spectral clustering algorithm, highlighting 3444 genes contained in the third cluster of the four-cluster structure (corresponding to the highest classification accuracy of the samples). Thus, the experimental data had the form (3269×3444).

During the modeling process, five first biclusters were identified at each iteration, and the value of the corresponding criterion was calculated for each of them. The biclustering evaluation was performed based on the average of all components of the

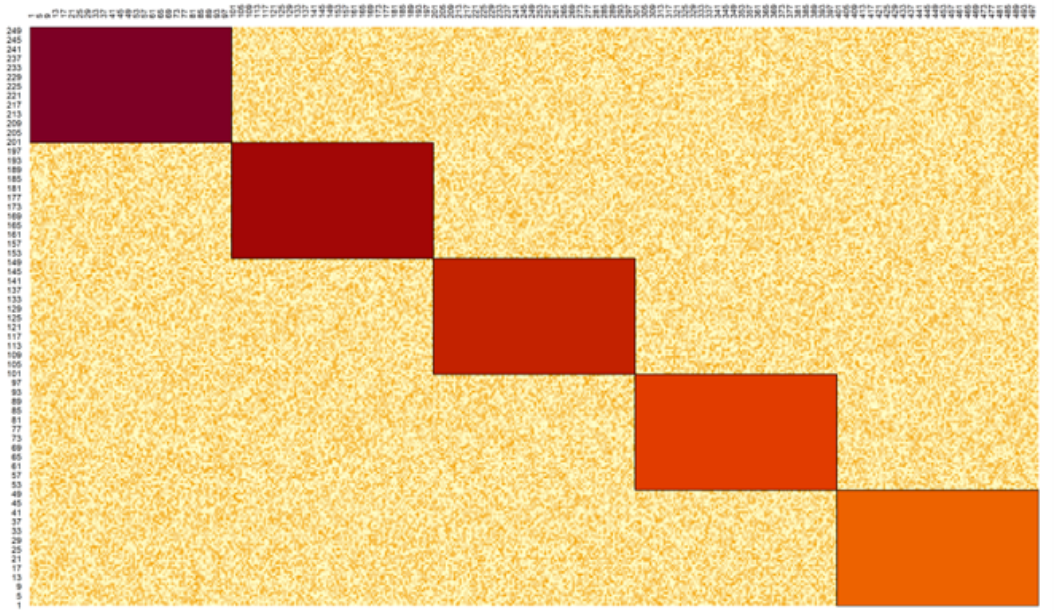

Figure 3.4: Result of applying the *ensemble* biclustering algorithm with optimal parameters to synthetic data

corresponding criterion, which determines each identified bicluster's coherence level. Analysis of the obtained results allows us to conclude that when applying both biclustering quality criteria, the minimum value is reached at the 25th iteration when using the MSR criterion and at the 10th iteration when using the mutual information (MI) based criterion. The values of the optimal parameters of the "ensemble" biclustering algorithm corresponding to the minima of the respective criteria are shown in Table 3.10.

Table 3.10: Optimal parameters of the "ensemble" biclustering algorithm according to MSR and MI criteria when applying the Bayesian optimization algorithm

Biclustering quality criteria	Parameters of ensemble biclustering algorithm	
	thr	simthr
MSR	0.263	0.395
MI_dist	0.549	0.151

The next step is the assessment of the adequacy of the relevant quality criteria for biclustering of GED through the analysis of biclusters obtained using the "ensemble" BC algorithm. Figures 3.5 and 3.6 show the distribution diagrams of the number of samples in biclusters, the number of genes, and the values of the re-

spective criteria, with 31 biclusters being studied when applying the MSR criterion and 18 biclusters when applying the criterion based on mutual information analysis (the number of biclusters corresponded to the maximum number when applying the biclustering algorithm with respective parameters). The analysis of the obtained

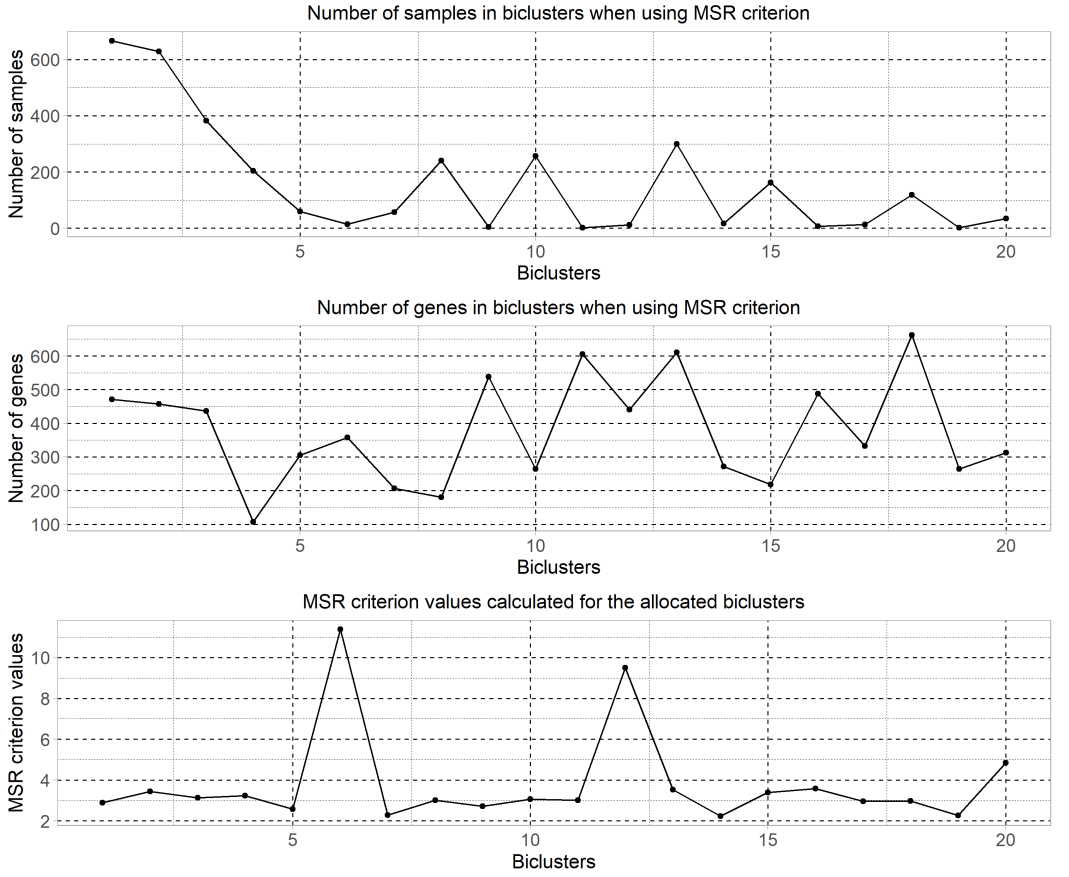

Figure 3.5: Modeling results regarding the application of the "ensemble" BC algorithm with optimal parameters according to the MSR criterion

diagrams allows concluding that in each case, a certain number of biclusters can be identified, containing a very small number of samples. The number of genes in the identified biclusters varies in a sufficiently high range, approximately from 100 to 700 genes per bicluster when applying both criteria. This fact indicates a high level of informativeness of the identified biclusters by the number of genes. When analyzing the distribution diagrams of the quality criteria values that define the coherence of biclusters, the conclusions are not so unequivocal. Statistical analysis showed that the values of the MSR criterion almost do not correlate with the numbers of samples

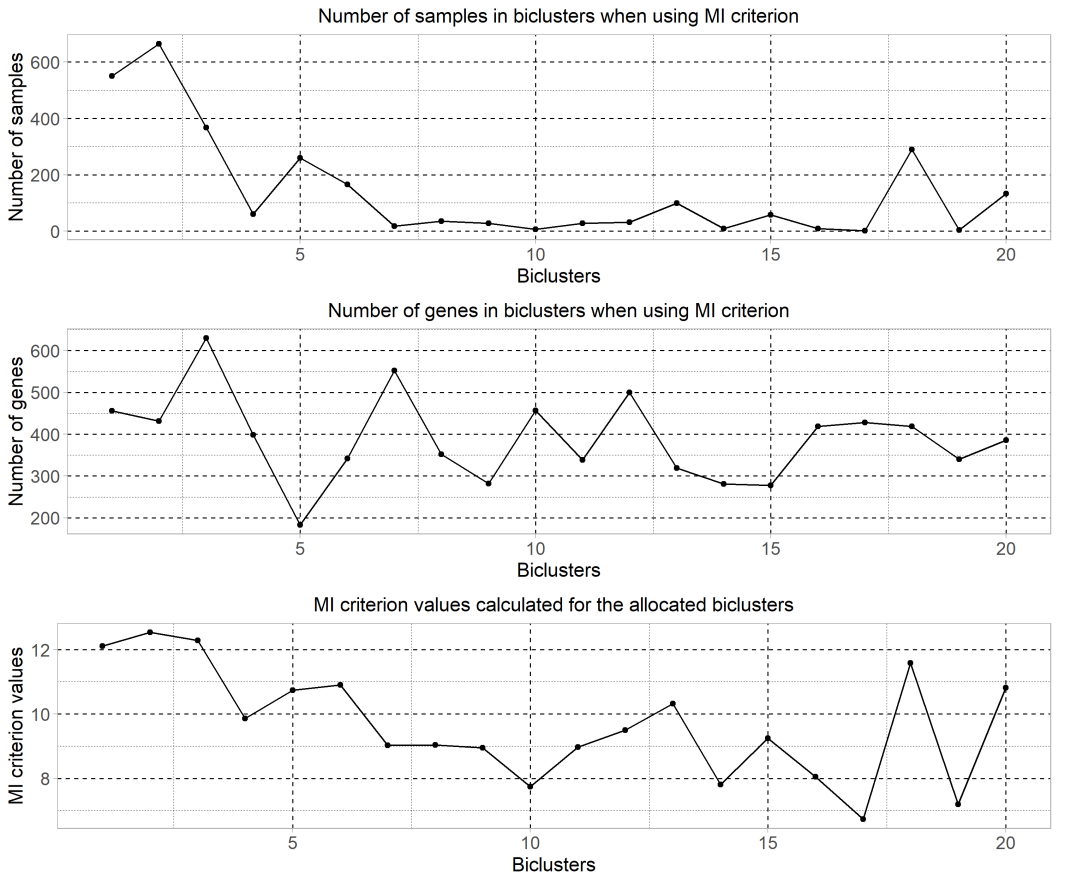

Figure 3.6: Modeling results on the application of the "ensemble" BC algorithm with optimal parameters according to the MI-dist criterion

and genes in the biclusters. In most cases, the value of this criterion varies within a relatively narrow range, indicating that the level of coherence of most biclusters according to this criterion is quite high. Figure 3.7 shows the result of the correlation analysis of the data table, which contains the values of the calculated criteria and the numbers of samples and genes in the respective biclusters. The evaluation of the results leads to the conclusion that the value of the MSR criterion shows little to no correlation with the gene count and exhibits a slight negative correlation with the sample size in the bicluster. This fact indicates the appropriateness of using this criterion for forming the bicluster structure. Biclusters informative according to this criterion correspond to its minimum value regardless of the number of samples and genes in the bicluster. An alternate inference emerges from examining the modeling outcomes using the mutual information-based criterion. The value of this criterion

Figure 3.7: The result of the correlation analysis of the BC quality criteria values and the numbers of samples and genes in biclusters

has a high positive correlation with the number of samples in the bicluster (0.85) and a minor positive correlation with the number of genes (0.2), indicating the dependence of this criterion's value on the size of the bicluster. This observation leads to the speculation that the criterion might lack reliability in generating a meaningful bicluster configuration. This conjecture could be validated or disproven through model verification, which would involve scrutinizing the biclusters and employing GOA on the data within these biclusters.

The results of the bicluster analysis of gene expression data with identifying sample types in the corresponding biclusters using criteria based on the MSR and MI metrics are presented in Figures 3.8 and 3.9. The analysis of the obtained results indicates a greater attractiveness of the method based on applying the MI criterion. In both cases, the samples in the biclusters are formed identically (luad, lusc, stad; acc, sarc; gbm, lgg; kirc). It should be noted that in both cases, a small number of samples corresponding to the normal state of patients (norm) were identified. This fact can be explained by the lack of correlation between the samples and genes due to the absence of the diseases being studied, and the gene expression values in a biological organism can vary significantly due to different biological processes. However, as modeling results show, when applying the MSR criterion in forming

BC 1				BC 2				BC 3			BC 4		
Gene	Sample			Gene	Sample			Gene	Sample		Gene	Sample	
465	gbm	lgg	nrm	461	gbm	lgg	nrm	399	luad	stad	734	kirc	
	139	526	5		108	514	5		3	49		136	
BC 5				BC 6				BC 7			BC 9		
Gene	Sample			Gene	Sample			Gene	Sample		Gene	Sample	
124	luad	lusc	stad	632	kirc	nrm		183	luad	lusc	stad	516	lusc
	99	29	22		342	1			118	49	57		24
BC 8				BC 10				BC 14			BC 16		
Gene	Sample			Gene	Sample			Gene	Sample		Gene	Sample	
306	luad	lusc	stad	326	luad	lusc	stad	332	acc	sarc	535	gbm	
	7	6	48		106	27	18		5	16		2	
BC 11				BC 12				BC 13			BC 17		
Gene	Sample			Gene	Sample			Gene	Sample		Gene	Sample	
324	luad	lusc	stad	339	luad	lusc		456	luad	lusc	stad	397	stad
	69	64	1		5	23			71	58	1		5
BC 15			BC 18			BC 19			BC 20			BC 25	
Gene	Sample		Gene	Sample		Gene	Sample		Gene	Sample		Gene	Sample
434	luad	stad	238	acc	sarc	266	lusc		272	luad	stad	498	lusc
	7	1		4	18		4			19	1		11
BC 21			BC 22			BC 23				BC 24			
Gene	Sample		Gene	Sample		Gene	Sample			Gene	Sample		
351	acc	sarc	569	kirc	norm	256	luad	lusc	stad	402	acc	sarc	
	5	21		360	1		155	69	65		3	2	

BC 26		BC 27			BC 28			BC 29		BC 30		
Gene	Sample	Gene	Sample		Gene	Sample		Gene	Sample	Gene	Sample	
435	luad	273	luad	lusc	292	acc	sarc	382	stad	177	acc	sarc
	15		1	16		6	10		3		11	17

BC 31			
Gene	Sample		
104	luad	lusc	stad
	104	44	51

Figure 3.8: The result of the bicluster analysis of gene expression data using the criteriuon based on MSR metric

the bicluster structure, the number of biclusters is significantly higher compared to the case of using the criterion based on mutual information assessment (31 versus 18). Moreover, in the first case, a significantly larger number of small biclusters were identified, which, in terms of the number of samples, have low informativeness for further forming a subset of informative genes for disease diagnosis based on the identified gene expression data. However, this hypothesis can be confirmed or refuted by implementing a classification procedure for objects containing the selected gene expression profiles as attributes when implementing the corresponding type of biclusters allocation.

BC 1				BC 2				BC 3			
Gene	Sample			Gene	Sample			Gene	Sample		
456	gbm	lgg	norm	430	gbm	lgg	norm	399	luad	lusc	stad
	53	494	4		137	524	5		7	5	49

BC 4			BC 5				BC 8		BC 9		
Gene	Sample		Gene	Sample			Gene	Sample	Gene	Sample	
339	luad	lusc	123	luad	lusc	stad	484	lusc	782	kirc	norm
	5	23		105	32	22		4		197	1

BC 6				BC 7				BC 11			
Gene	Sample			Gene	Sample			Gene	Sample		
189	luad	lusc	stad	329	luad	lusc	stad	463	luad	lusc	stad
	127	56	60		104	33	23		68	62	1

BC 10		BC 12			BC 14				BC 16		
Gene	Sample	Gene	Sample		Gene	Sample			Gene	Sample	
461	sarc	505	acc	sarc	348	luad	lusc	norm	612	kirc	norm
	13		5	2		14	1	4		299	1

BC 13		BC 15				BC 17			BC 18		
Gene	Sample	Gene	Sample			Gene	Sample		Gene	Sample	
450	lusc	315	luad	lusc	stad	346	acc	sarc	231	acc	sarc
	14		6	3	69		6	5		8	30

Figure 3.9: The result of the bicluster analysis of gene expression data using the criteriuon based on MI metric

3.5.5 Assessment of the Bicluster Structure Adequacy Through Gene Ontology Analysis

In the context of information technology and bioinformatics, ontology is a formalized representation of knowledge that uses a controlled vocabulary and a set of relationships between terms to describe the considered domain [77, 9]. Such an ontology can be used for modeling a subject area and serves for information exchange, data integration, and the development of various computer applications, including artificial intelligence. In bioinformatics, ontologies are used to structure and standardize information about biological processes, protein functions, cellular components, etc.

The Gene Ontology (GO) is an example of such a system, allowing genes and protein products to be annotated in a unified form, ensuring consistency and compatibility of biological databases.

Biclustering and data analysis based on gene ontology are linked through their common goal: understanding the biological mechanisms and functional characteristics of genes revealed in experimental gene expression data. While biclustering allows the identification of groups of genes that show similar expression patterns under different conditions or in different sample types (in the presence of various disease types), which is essential for understanding which genes are co-regulated in

certain physiological states or respond to specific external stimuli, the analysis of data in biclusters based on gene ontology allows determining the possible role of identified genes in the cell or organism being studied. In other words, gene ontology provides functional annotation of genes. By integrating biclustering results with GO-based analysis, it becomes possible to gain a deeper understanding of the biological context of gene expression patterns and to identify groups of genes that are co-expressed in the presence of a particular type of disease.

Thus, biclustering and GO-based analysis complement each other, providing a mechanism for identifying and functionally understanding biological modules in large gene expression datasets. The procedure for identifying significant genes based on gene ontology analysis was carried out in the R programming environment using packages and functions from the Bioconductor module [1]. The practical implementation of this procedure includes the following steps:

1. **Loading the necessary packages in R.** During the simulation process, the following packages were used for gene ontology analysis and the selection of informative genes: GO.db [34], org.Hs.eg.db [35], biomaRt [39], and topGO [5].

2. **Data preparation.** Forming a list of vectors of gene identifiers (ENTREZ ID) contained in the identified biclusters.

3. **Mapping genes to GO terms using functions from the org.Hs.eg.db package.** Obtaining GO terms for all genes contained in the bicluster.

4. **Statistical analysis of gene expression values** to assess the probability (p-value) that differences between gene expression values corresponding to different classes of the studied samples could have arisen by chance. At this stage, the ANOVA (Analysis of Variance) statistical method was used, which allows comparing the mean values of three or more groups. In the context of gene expression analysis, ANOVA is used to determine whether there is a statistically significant difference in gene expression levels between different sample classes. The obtained p-values, in this case, indicate the probability that the observed differences could have arisen by chance. The Benjamini-Hochberg (BH) method was used to adjust p-values (calculate p-adjust) to control for Type I errors when performing multiple comparisons.

5. **Creating a topGOdata object**, which contains all gene identifiers and their scores, GO annotations, the hierarchical structure of GO, and all other information necessary for performing enrichment analysis of the studied genes.

6. **Performing enrichment tests.** Two types of statistical tests were applied in the dissertation research: the Fisher test, which is based on counting the num-

ber of genes corresponding to each sample class, and the Kolmogorov-Smirnov test, which calculates enrichment based on gene expression values. Each of these tests provides an assessment of how differentially expressed the corresponding gene is, allowing genes to be categorized by their level of differential expression.

7. **Forming a matrix of gene ontology analysis results** with gene identifiers corresponding to significant gene ontologies based on the analysis results.

8. **Forming a vector of significant genes for the corresponding bicluster** by matching the gene identifiers contained in the bicluster with the gene identifiers identified as a result of gene ontology analysis.

The simulation process regarding the gene ontology method application to form a vector of significant genes, considering the sample type, was carried out using gene expression data from the first bicluster identified using the MSR criterion. The data included 465 genes and 670 samples. Figure 3.10 shows the result of applying the ANOVA statistical test to gene expression data (Volcano plot).

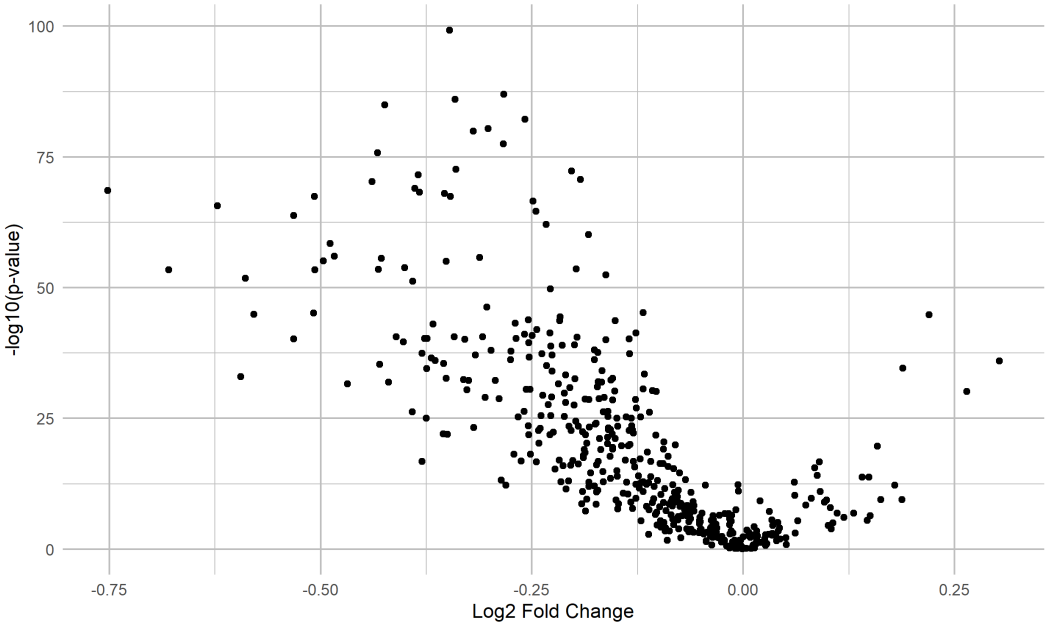

Figure 3.10: Visualization of the gene p-values distribution by their significance level (Volcano Plot)

The horizontal axis (Log2 Fold Change) on the chart displays the level of gene expression value of one group of genes compared to the expression of genes in an-

other group. Genes to the left of the center have lower expression in the first group compared to the second group. Genes to the right of the center have higher expression in the first group. It is evident that the further a gene is from the center, the greater its level of differential expression. The vertical axis displays p-values (p-adjust) in a logarithmic scale (-log10(p-adjust)). Genes allocated higher on the graph have lower p-values, indicating greater statistical significance of the difference in expression.

The analysis of the obtained results allows us to conclude that a relatively large number of genes contained in the bicluster can be identified as insignificant (located at the bottom center of the diagram), which confirms the need for further analysis to remove them.

The next step is to implement enrichment tests with the calculation of p-values, which determine the significance level of genes according to the corresponding test. As mentioned above, the Fisher and Kolmogorov-Smirnov tests were used during the simulation. The simulation results are shown in Figure 3.11. In both tests, 388

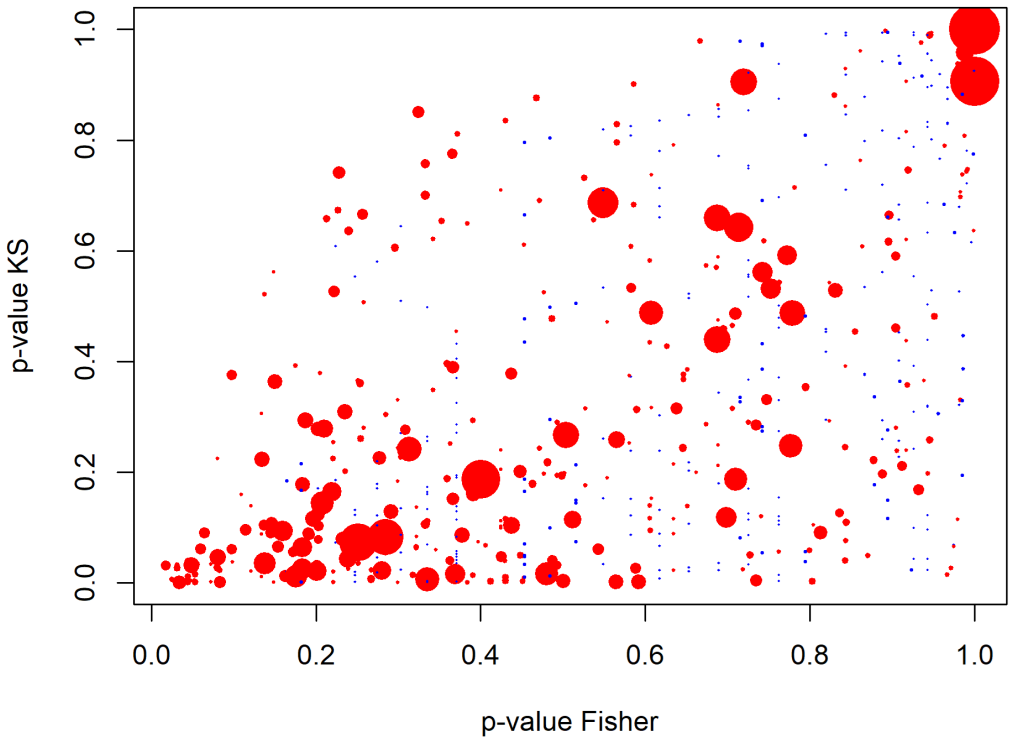

Figure 3.11: Scatter plot of p-values distribution calculated using the classical Fisher test (x-axis) and the Kolmogorov-Smirnov method (y-axis)

significant GO terms out of 704 were identified. In the depicted diagram, the size of the dot is proportional to the number of annotated genes for the corresponding GO term, and its color reflects the number of significantly differentially expressed genes. The threshold parameter, which separates genes into significant and insignificant, was chosen at the median level of the gene significance vector. As can be seen, the red dots contain significantly more genes than the blue ones. The analysis of the diagram presented in Figure 3.11 also allows us to conclude that the results of applying the Fisher and Kolmogorov-Smirnov tests differ from each other. Some GO terms identified as significant using the Fisher test are less significant when using the Kolmogorov-Smirnov test. However, in some cases, it is possible to visually identify several GO terms for which the p-values from both tests are almost identical. The obtained results also indicate that despite the same number of significant genes when using both tests, the application of a single test to form a subset of significant genes based on GO analysis is not objective. In this case, increasing the objectivity of the analysis can be achieved by using both tests with the formation of intermediate decisions, followed by their combination to select unique identifiers of significant genes.

Figures 3.12 and 3.13 present the results of GO analysis, highlighting the ten significant GO terms when using the Fisher and Kolmogorov-Smirnov tests, respectively. Significant nodes are represented as rectangles. The color of the node represents the relative significance, ranging from dark red (most significant) to bright yellow (least significant). The analysis of the obtained graphs confirms the conclu-

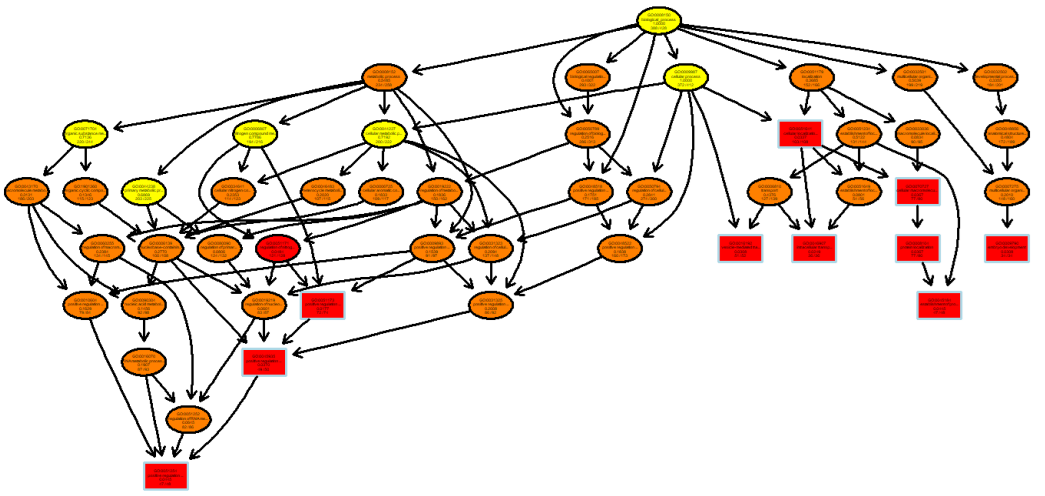

Figure 3.12: The result of applying GO analysis, highlighting ten significant GO terms using the Fisher test

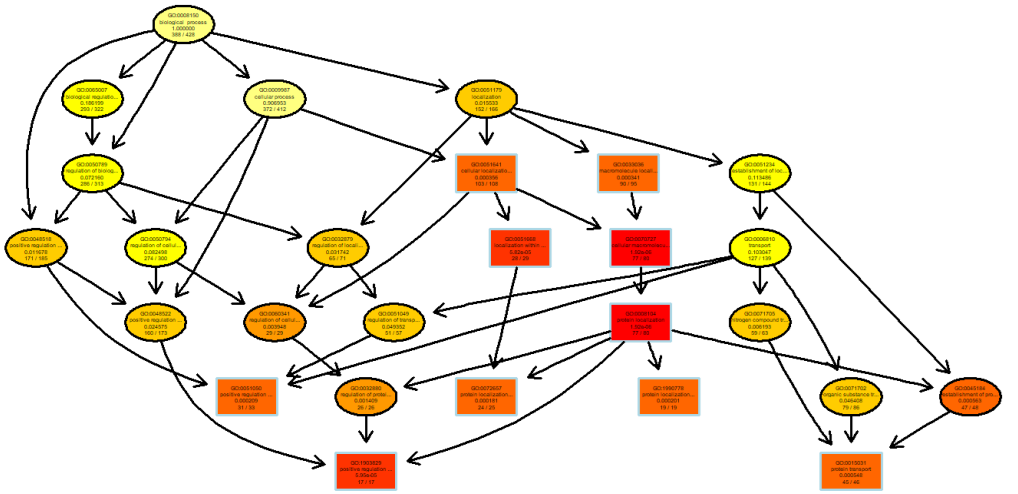

Figure 3.13: The result of applying GO analysis, highlighting ten significant GO terms using the Kolmogorov-Smirnov test

sion regarding the inconsistency of results when using different tests in GO analysis to form a subset of significant genes. As seen from the figures, when highlighting the ten most significant GO terms, the results differ both in the graph topology and in the significance level of the GO terms that are the nodes of the graph. This fact confirms the hypothesis regarding the advisability of using both tests to form a subset of significant genes.

The simulation results showed that the application of GO analysis results allows us to form a table of GO terms and gene identifiers corresponding to the relevant terms. The simulation results corresponding to the top ten matches for the most significant GO term using both tests are shown in Figure 3.14. As can be seen from the table, a large number of genes may correspond to a single GO term. For instance, when applying Fisher's test, the total number of genes corresponding to 388 significant GO terms was 26,092, while in the case of the Kolmogorov-Smirnov test, it was 24,456. In line with the stated objective, the final step involved associating the gene identifiers contained in the bicluster with the gene identifiers highlighted by the GO analysis. This highlighted 270 genes using Fisher's test and 254 genes using the Kolmogorov-Smirnov test. The total number of genes in the bicluster was 465. When combining the results of the two tests and highlighting unique gene identifiers, the total number of significant genes amounted to 296.

However, it should be noted that the above type of GO analysis is effective when applied to data containing at least two classes of samples, with a sufficiently large number of samples in each class. If these conditions are invalid, the ANOVA test may

№	Fisher test				Kolmogorov-Smirnov test			
	GO	Evidence	Ontology	EntrezID	GO	Evidence	Ontology	EntrezID
1	GO:0051173	IGI	BP	3458	GO:0008104	IDA	BP	190
2	GO:0051173	ISS	BP	7124	GO:0008104	IGI	BP	287
3	GO:0046907	TAS	BP	348	GO:0008104	ISS	BP	583
4	GO:0046907	NAS	BP	1176	GO:0008104	IEA	BP	604
5	GO:0046907	NAS	BP	3257	GO:0008104	ISS	BP	857
6	GO:0046907	IMP	BP	7879	GO:0008104	ISS	BP	859
7	GO:0046907	NAS	BP	8120	GO:0008104	IBA	BP	987
8	GO:0046907	IBA	BP	8195	GO:0008104	IEA	BP	998
9	GO:0046907	ISS	BP	8195	GO:0008104	IBA	BP	1130
10	GO:0046907	NAS	BP	8546	GO:0008104	IEA	BP	1454
…	…	…	…	…	…	…	…	…

Figure 3.14: Modeling results using GO analysis based on Fisher and Kolmogorov-Smirnov tests (10 correspondences to the first most significant GO terms are presented)

either not work or yield unreliable results. For this reason, this type of GO analysis is appropriate when performing cluster analysis of gene expression profiles, where each cluster corresponds to a complete set of sample classes with a sufficiently large number of samples in each class. When applying bicluster analysis, the condition for using the ANOVA test may not be met, as biclusters may include only one class of samples, or the number of samples corresponding to one of the classes may be relatively small, reducing the reliability of the test results. In this case, it is advisable to apply a statistical test based on assessing whether the number of genes associated with a specific GO term in the list of genes comprising the bicluster differs from what is expected by chance. In other words, the statistical test compares the number of genes in the selected GO category of genes contained in the bicluster with their total number in the genome of the studied object. In the context of current research, the statistical test was implemented in the R software environment using the *enrichGO()* function from the *clusterProfiler* package [79, 85]. The application of the statistical test using the *enrichGO()* function involves two steps:

- **Implementation of the hypergeometric test** by comparing the number of genes associated with a specific GO term with what is expected by chance. It should be noted that the GO term database must correspond to the type of biological object being studied. In the current research, the GO terms database homo sapiens “org.Hs.eg.db” was used.

- **p-value correction.** The necessity of this step is determined by the large number of GO terms analyzed. Therefore, it is necessary to adjust the p-values to control for multiple comparisons. The application of the Benjamini-Hochberg (BH) method helps reduce Type I error.

The result of the GO analysis applied based on the *enrichGO()* function is a table with GO terms, which also contains p-values, adjusted p-values, and the number of

genes in each term. Table 3.11 presents the results of the GO analysis of gene expression data from the first bicluster obtained using the MSR criterion (the first ten rows are shown). The threshold value separating significant and non-significant

Table 3.11: Results of the GO analysis using the statistical test based on the *enrichGO()* function applied to gene expression data from the first bicluster

No	ID	GeneRatio	p-value	p.adjust	Count
1	GO:0007409	42/428	6.361398e-15	2.358170e-11	42
2	GO:0010975	41/428	4.477555e-14	8.299147e-11	41
3	GO:0050771	13/428	2.408776e-09	2.813517e-06	13
4	GO:0050770	19/428	3.035896e-09	2.813517e-06	19
5	GO:0050890	26/428	1.789554e-08	9.847434e-06	26
6	GO:0007411	22/428	1.859510e-08	9.847434e-06	22
7	GO:0097485	22/428	1.859510e-08	9.847434e-06	22
8	GO:0031345	19/428	8.314768e-08	3.852856e-05	19
9	GO:0048675	15/428	1.020838e-07	4.204719e-05	15
10	GO:0010977	16/428	1.213019e-07	4.496662e-05	16
			...		

GO terms was set at 0.05. At this value, 118 significant GO terms were identified. Figure 3.15 shows a dot plot of the distribution of the 20 most significant GO terms, with the size of the dots representing the number of genes and the color indicating the adjusted p-value. Figure 3.16 shows a network of the connections of the five most significant GO terms and their corresponding genes. As can be seen, similar to the previous simulation results, each GO term corresponds to a relatively large number of genes, confirming the necessity of filtering gene identifiers at a certain stage of data processing.

The simulation results created the conditions for developing a hybrid model for identifying significant genes from gene expression data based on the comprehensive application of clustering or biclustering analysis and the method based on GO analysis. The structural diagram of the step-by-step procedure implemented within the framework of the model is shown in Figure 3.17. Its implementation involves the following stages:

Stage I. Data preparation and implementation of cluster or bicluster analysis

1.1. Formation of gene expression data in the form of a matrix, where rows are samples and columns are genes, with expression values defining the state of the corresponding samples.

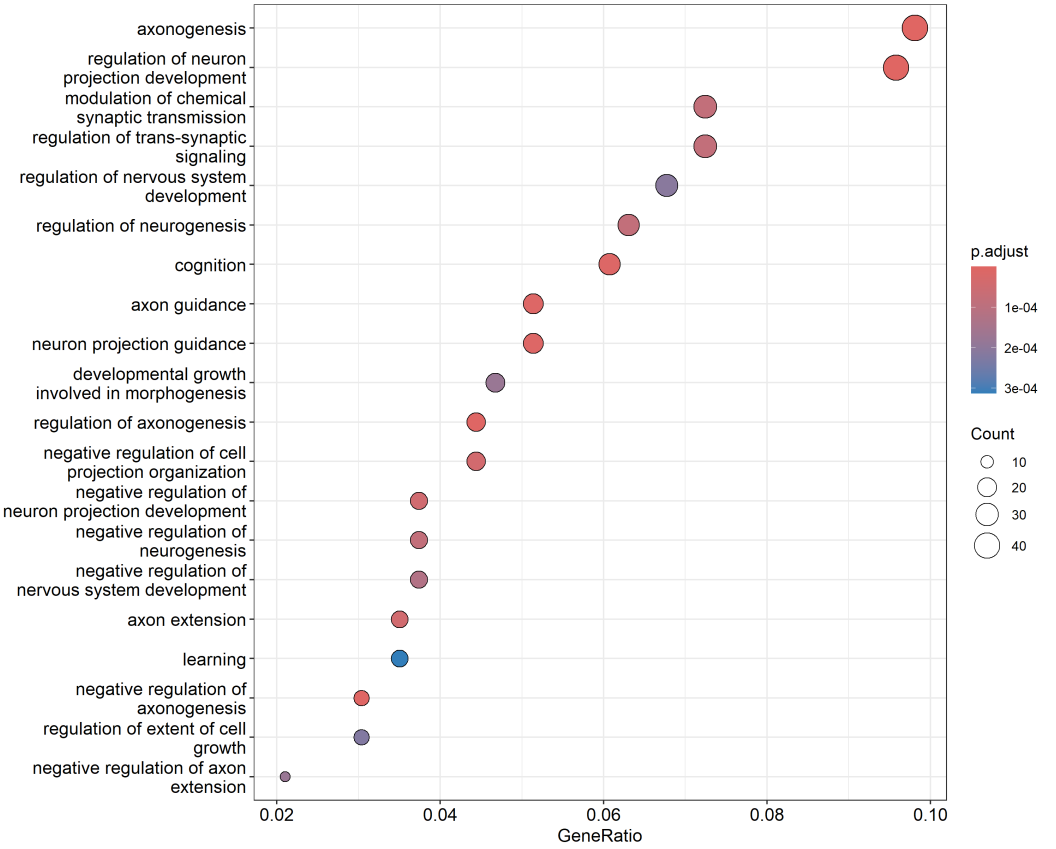

Figure 3.15: Scatter plot of the 20-th significant GO terms obtained using the *enrichGO()* function

1.2. Setting the parameters of the clustering or biclustering algorithm using the Bayesian optimization method.

1.3. Clustering or biclustering of gene expression data. Formation of subsets of clusters or biclusters.

Stage II. Application of GO analysis to the selected subsets of gene expression data

2.1. When using cluster analysis:

2.1.1. Application of the ANOVA test to compare the mean expression values of genes corresponding to different classes of samples. Formation of the p-

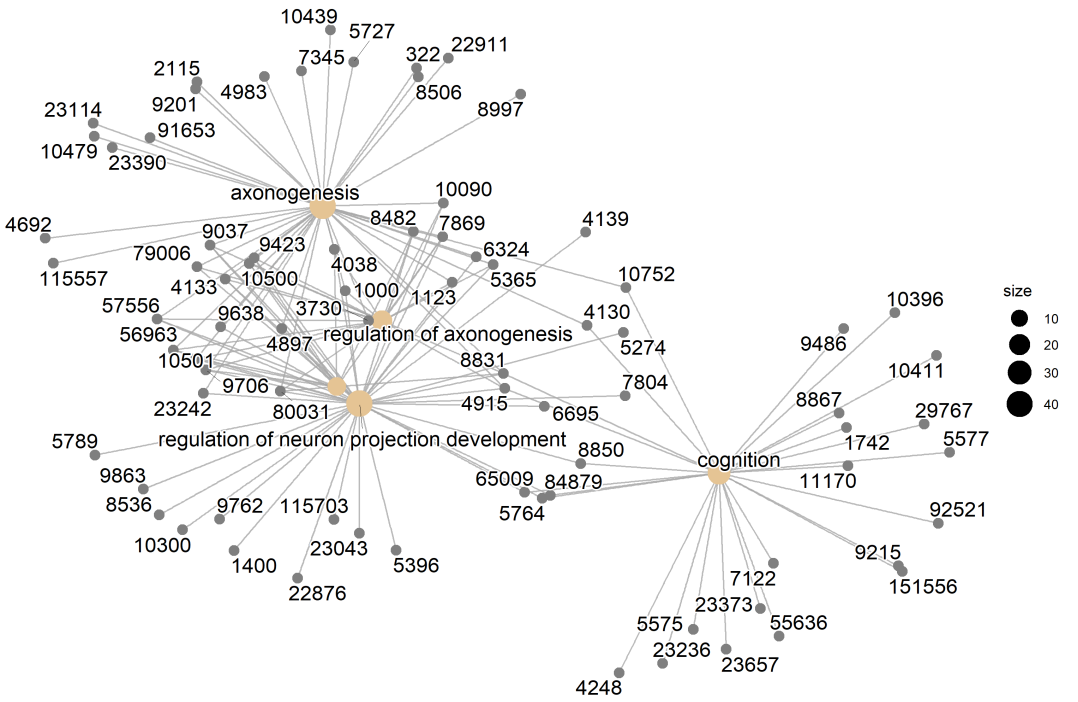

Figure 3.16: Graph of connections of the five most significant GO terms with their corresponding genes

value vector, which determines the probability that the mean expression value of the corresponding gene for all groups is the same, i.e., this gene is insignificant in its discriminative ability.

2.1.2. Identification of genes showing significant variation ($p < 0.05$) in their expression between classes.

2.1.3. Application of GO analysis to selected genes to determine their biological functions.

2.1.4. Application of Fisher's and Kolmogorov-Smirnov tests, where Fisher's test allows determining the statistical significance of differences in category frequencies, i.e., the number of genes expressed in each class of samples, and the Kolmogorov-Smirnov test determines statistical significance when comparing gene expression distributions between classes.

2.1.5. Integration of results and determination of significant genes. At this step, results obtained from both tests are compared and genes that are significant by both tests and associated with key biological processes are identified through GO analysis.

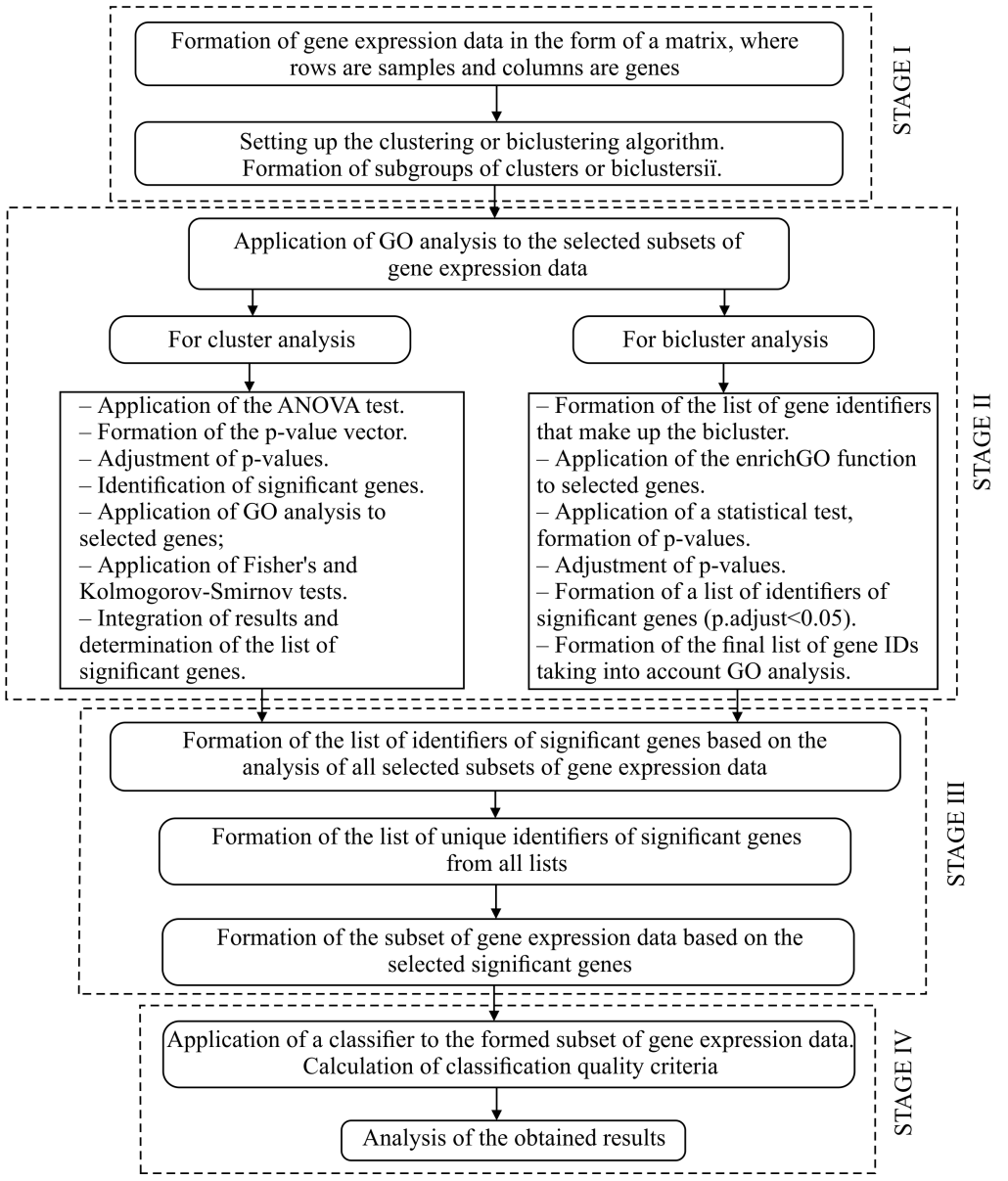

Figure 3.17: Structural diagram of the model for forming subsets of significant genes based on cluster-bicluster analysis and GO analysis

2.1.6. Identification of significant genes that make up the cluster being analyzed and are in the list of significant genes formed in the previous step.

2.2. When applying bicluster analysis:

2.2.1. Identification of the list of genes that make up the bicluster and application of the *enrichGO()* function to the selected genes.

2.2.2. Application of a statistical test to compare the number of genes associated with specific GO terms with what is expected by chance.

2.2.3. Correction of p-values to control for multiple comparisons using the Benjamini-Hochberg (BH) method.

2.2.4. Interpretation of results, identification of the list of genes that are significant by the adjusted p-value ($p.adjust < 0.05$).

2.2.6. Formation of the final list based on the results of the statistical test and GO analysis.

Stage III. Formation of the list of identifiers of significant genes based on the analysis of all selected subsets of gene expression data

3.1. Implementation of Stage II for all subsets of gene expression data using cluster or bicluster analysis. Formation of intermediate solutions, i.e., lists of significant genes corresponding to each subset of gene expression data.

3.2. Consolidation of results. Formation of the list of unique gene identifiers that are significant across all lists.

3.3. Formation of the subset of gene expression data, where rows are samples and columns are the expression values of significant genes identified in step 3.2 of this procedure.

Stage IV. Evaluation of the effectiveness of the above procedure through the implementation of samples classification procedure, including selected gene expression data as attributes

4.1. Formation of gene expression data subsets for model training, validation, and testing.

1. Formation of functions for evaluating the quality of the classifier during model training (classification accuracy and loss function value) and during testing (accuracy, F1-score).

4.2. Set up the classifier, determining the optimal values of hyperparameters of the model.

4.3. Model training and validation.

4.4. Model testing. Calculation of classification quality criteria.

4.5. Analysis of the obtained results.

The simulation results regarding the application of the proposed step-wise procedure to the bicluster structure obtained by applying a subset of gene expression data formed as a result of using an inductive clustering algorithm (3444 genes), showed that the application of a metric based on the MSR criterion identified 1447 significant genes, and the utilizing a mutual information-based metric led to the identification of 1780 significant genes. Consequently, through the implementation of GO analysis on the respective cluster configurations, two data matrices were generated, measuring (3269×1447) and (3269×1780) for the MSR and MI metrics, respectively..

To assess the efficacy of the introduced approach, a 1D two-layer GRU recurrent neural network (RNN) was deployed on the data gathered. The Bayesian optimization algorithm was utilized to ascertain the optimal count of neurons for each layer, with the neuron range set between 20 and 100. The findings indicated that an increase in neuron count generally led to network overfitting, marked by a growing gap in classification performance between the training and validation datasets. Through Bayesian optimization, the optimal neuron configuration for the network analyzing the first dataset (derived from the MSR metric) was established at 83 and 87 for the first and second layers, respectively. Conversely, for the dataset generated via the MI metric, the neuron counts were set at 98 and 75 for the initial and subsequent layers. In a conventional approach to classifier application, the initial step involved splitting the data (samples) into training and testing subsets at a 0.7/0.3 ratio. Subsequently, the training subset was further divided at a 0.8/0.2 ratio, with the latter portion serving to validate the model during training.

Figures 3.18 and 3.19 illustrate the results of training and validation of RNNs using two datasets. The analysis of the obtained diagrams indicates the absence of overfitting in the models, as the discrepancy between the accuracy values calculated for the training data and the validation data, and the corresponding loss function values, increases slightly with an increase in training epochs, without exceeding the permissible norm. Tables 3.12 and 3.13 showcase the classification performance of the test subset from the significant genes expression dataset. From reviewing these outcomes, it's evident that the bicluster structure developed via mutual information criteria offers superior classification quality compared to that formed by the MSR metric. This difference may stem from the presence of numerous small biclusters whose data are insufficient for accurately compiling a list of significant gene identifiers through GO analysis. The modest classification accuracy observed in both instances could be attributed to the constraints of the dataset utilized for the modeling. Out of 19260 genes, merely 3444 were included, constituting the third cluster identified by spectral clustering. This selection process inherently restricts the amount of relevant information available for identifying the studied samples.

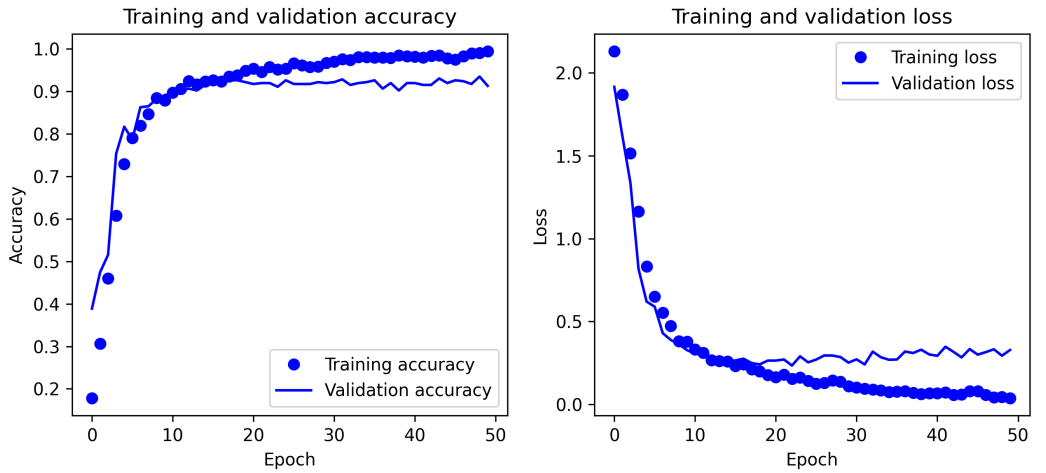

Figure 3.18: Accuracy distribution diagrams of sample classification and loss function at different stages of network training, calculated during the training and validation of the model when applying data obtained using the MSR criterion

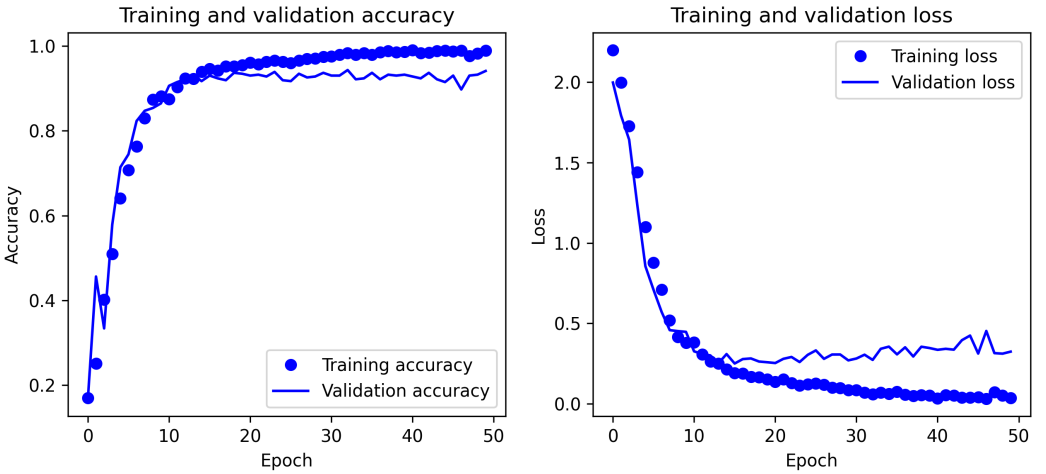

Figure 3.19: Accuracy distribution diagrams of sample classification and loss function at different stages of network training, calculated during the training and validation of the model when applying data obtained using the MI criterion

3.6 Hybrid Model for Identifying Gene Expression Data Samples Based on GO Analysis, Spectral Clustering Algorithm, Bicluster Analysis, and Convolutional Neural Network

According to the flowchart presented in Figure 3.17, the formation of subsets of significant and co-expressed gene expression data can be carried out using both

Table 3.12: Optimal parameters of the "ensemble" biclustering algorithm according to MSR and MI criteria when applying the Bayesian optimization algorithm

Cancer type	Classification metrics				Total samples	Correctly classified
	Precision	Recall	F1-score	Overal accuracy		
acc	0.963	0.929	0.945		28	26
gbm	0.821	0.780	0.8		59	46
kirc	0.964	0.947	0.955		169	160
luad	0.918	0.940	0.929		166	156
lgg	0.901	0.919	0.910	92.4%	149	137
lusc	0.887	0.926	0.906		135	125
normal	0.906	0.935	0.921		62	58
sarc	0.933	0.886	0.909		79	70
stad	0.985	0.955	0.970		134	128

Table 3.13: Optimal parameters of the "ensemble" biclustering algorithm according to MSR and MI criteria when applying the Bayesian optimization algorithm

Cancer type	Classification metrics				Total samples	Correctly classified
	Precision	Recall	F1-score	Overal accuracy		
acc	0.893	0.893	0.893		28	25
gbm	0.915	0.915	0.915		59	54
kirc	0.982	0.964	0.973		169	163
luad	0.982	0.975	0.979		166	162
lgg	0.873	0.926	0.899	94.3%	149	138
lusc	0.923	0.889	0.906		135	120
normal	0.908	0.952	0.929		62	59
sarc	0.962	0.949	0.955		79	75
stad	0.977	0.963	0.970		134	129

cluster and bicluster analysis at the data preprocessing stage. At the same time, the comprehensive application of bicluster analysis and gene ontology analysis provides a deeper and more thorough examination. While bicluster analysis allows for the identification of coherent subsets of genes and samples considering the experimental conditions, gene ontology analysis provides for the formation of a list of significant gene identifiers considering the type and condition of the biological object being studied. This subsection examines the effectiveness and feasibility of applying a step-by-step procedure for clustering and biclustering gene expression data using gene ontology analysis both at the gene expression data preprocessing level and after the biclustering process is implemented. The feasibility of implementing the cluster-bicluster analysis procedure is determined by the following reasons: the clustering stage allows for the formation of subsets of gene expression profiles based on their similarity according to the chosen metric. Within the framework of the current research, the formation and evaluation of the cluster structure were performed using a metric based on the mutual information evaluation of gene expression profiles. Implementing the biclustering procedure on the formed clusters allows for the

identification of subsets of genes exhibiting high expression in certain samples or under certain experimental conditions. Thus, combining these two methods ensures a deeper and more comprehensive analysis. Clustering reveals general trends in the data, while biclustering uncovers specific interrelations that may be crucial for identifying the state of the object in diagnostic systems.

At the stage of implementing bicluster analysis, gene expression data obtained using the spectral clustering algorithm were applied. The simulation results analysis presented in this section allowed the formation of two clusters of gene expression data, containing 6,150 and 8,466 genes, respectively. Each cluster contained 3,021 samples corresponding to six types of cancer. One group of samples corresponded to objects in which cancer was not detected. According to the methodology presented in Figure, the first stage involved determining the optimal values of the ensemble bicluster algorithm hyperparameters (thr and simthr), which play a key role in the formation of biclusters, using the Bayesian optimization algorithm. The first parameter determines the threshold value at which elements (genes or samples) are included in the bicluster, while the second parameter sets the bicluster similarity threshold. A higher value of this parameter means that only very similar biclusters will be combined, while a lower threshold allows for greater diversity in the ensemble. When applying the Bayesian optimization algorithm, the target objective function was the average distance value based on the mutual information assessment, calculated for the rows and columns of the first five biclusters. Based on the simulation results, the following parameters of the ensemble bicluster algorithm were determined:

- For gene expression data of the first cluster: thr = 0.529, simthr = 0.248.
- For gene expression data of the second cluster: thr = 0.518, simthr = 0.267.

The simulation results regarding the distribution of samples and genes in the identified biclusters are shown in Figure 3.20.

As can be seen, with almost identical bicluster algorithm parameters, the biclustering results differ significantly. For instance, applying gene expression data from the first cluster (6,150 genes) resulted in 65 biclusters, with a large number of small biclusters containing relatively few samples and genes. The biclustering results on the second cluster's data (8,466 genes) are more appealing both in the number of biclusters (9) and the bicluster contents.

In the next stage, a list of unique gene identifiers was formed by applying gene ontology analysis to the data in the identified biclusters of each cluster in the first step and combining the identified subsets of significant genes with the subsequent formation of subsets of significant gene expression profiles based on the data of each cluster in the second step. As a result of this step implementation, new subsets

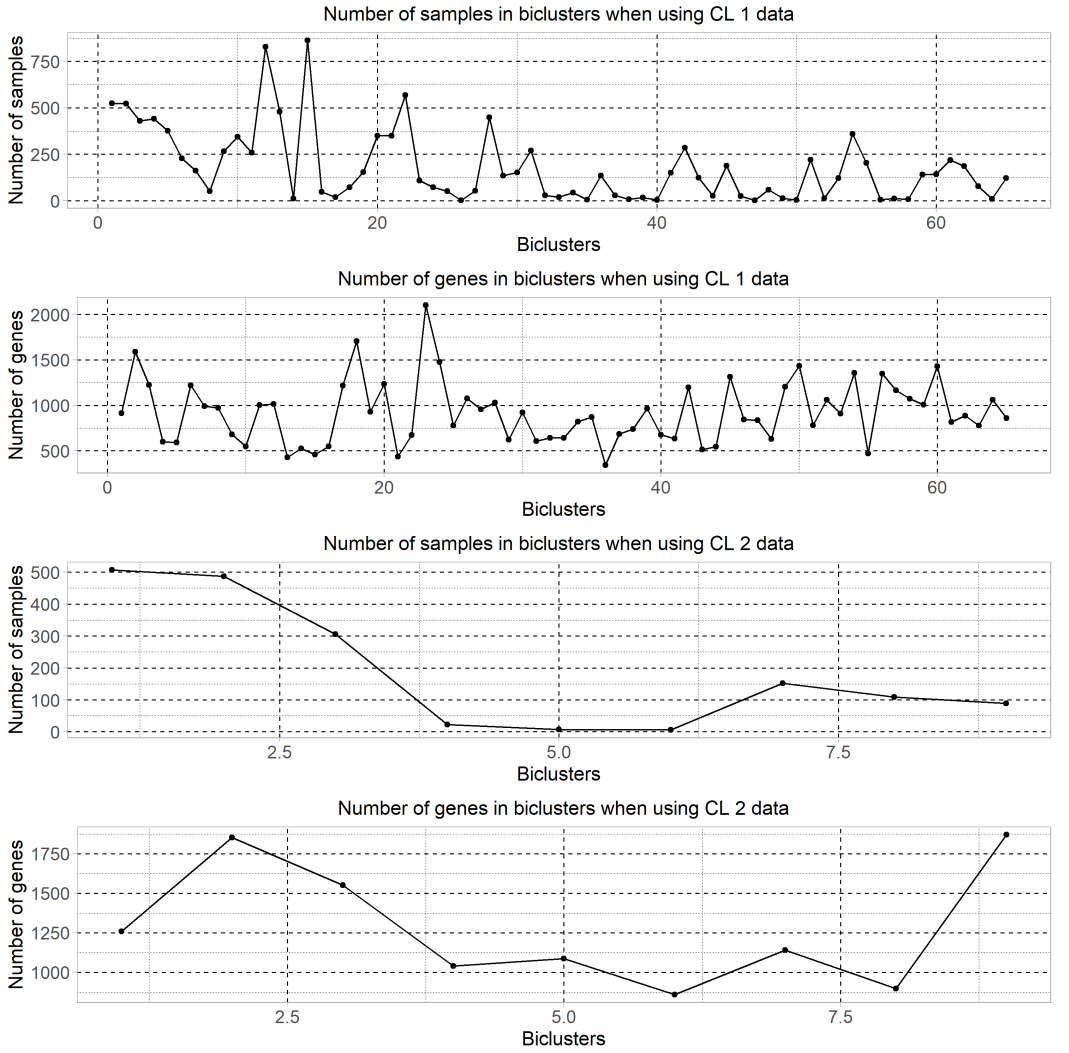

Figure 3.20: Simulation results on the application of bicluster analysis to gene expression data in the identified clusters

of gene expression data were formed, containing 4,442 and 4,520 significant genes corresponding to the first and second cluster data, respectively.

The next step is to apply the CNN to the formed data. Figure 3.21 shows the results of applying the Bayesian optimization algorithm to determine the optimal hyperparameters of the CNN when using the formed gene expression data as a result of cluster-bicluster analysis. The results of training and validating the models of the formed neural networks using 5-Fold cross-validation are shown in Figures 3.22

Optimal Hyperparameters of CNN using data from the first cluster							
The first convolutional layer			The second convolutional layer			Dropout value	Number of neurons in the density layer
Number of filters	Size of the filtering window	Conv. window size	Number of filters	Size of the filtering window	Conv. window size		
52	6	4	15	3	3	0.47	159
Optimal Hyperparameters of CNN using data from the second cluster							
The first convolutional layer			The second convolutional layer			Dropout value	Number of neurons in the density layer
Number of filters	Size of the filtering window	Conv. window size	Number of filters	Size of the filtering window	Conv. window size		
39	7	2	59	10	3	0.29	64

Figure 3.21: Simulation results on determining the optimal hyperparameters of CNN based on data obtained through cluster-bicluster analysis

and 3.23. Analyzing the obtained diagrams allows us to conclude that, as in the case of using gene expression data obtained by applying only cluster analysis, overfitting of the model is not observed when using data obtained through cluster-bicluster analysis since the values of classification accuracy and loss function obtained on the data for training and validating the model change consistently within the permissible error range.

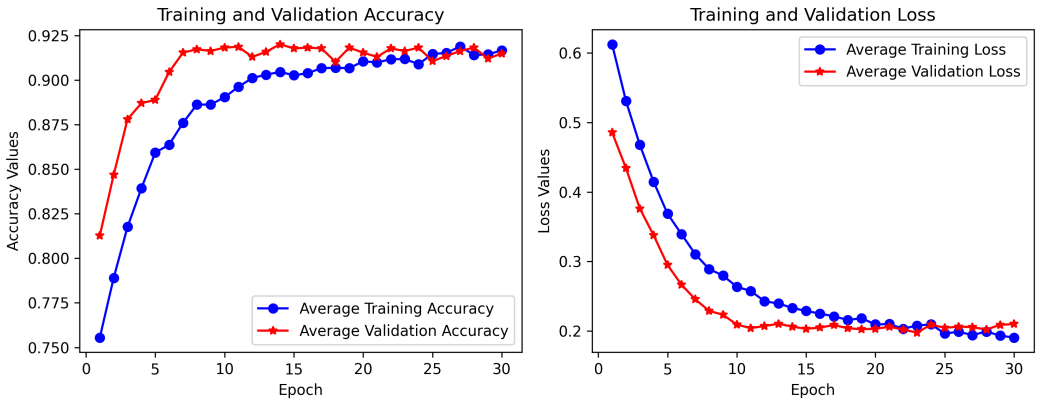

Figure 3.22: Simulation results on training and validating CNN using cluster-biclustering analysis with gene expression data from the first cluster

Figures 3.24 and 3.25 show the results of applying the trained CNN models to the test gene expression data. The analysis of the obtained results indicates that the models are more efficient when using gene expression data obtained through cluster-biclustering analysis. This fact confirms the feasibility of the comprehensive application of cluster analysis, biclustering data from identified clusters, and gene

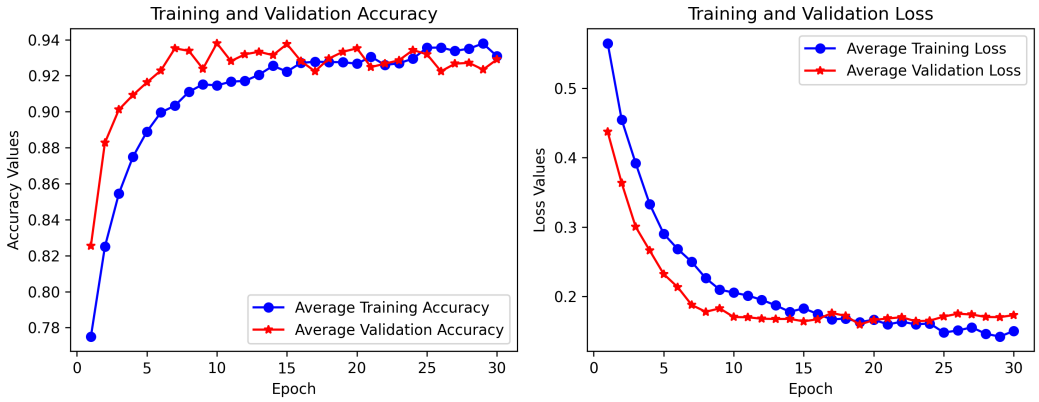

Figure 3.23: Simulation results on training and validating CNN using cluster-biclustering analysis with gene expression data from the second cluster

ontology analysis to form subsets of significant genes, considering the type of the object being studied.

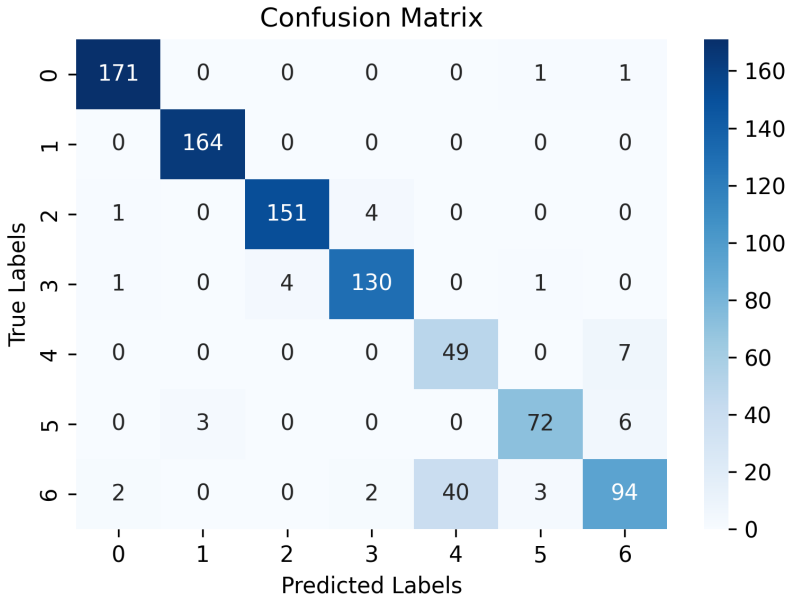

Figure 3.24: The confusion matrix formed as a result of applying the CNN model to the gene expression data of the first cluster, obtained through cluster-bicluster and gene ontology analysis

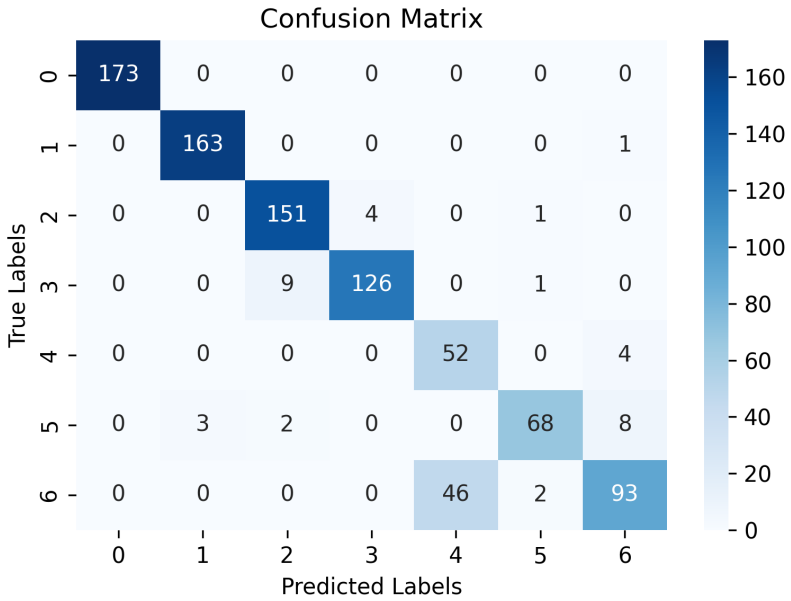

Figure 3.25: The confusion matrix formed as a result of applying the CNN model to the gene expression data of the second cluster, obtained through cluster-bicluster and gene ontology analysis

The application of bicluster analysis to the gene expression data of the respective clusters in the first step and gene ontology analysis in the second step resulted in a reduction in the number of significant genes from 6,150 to 4,442 when using the data from the first cluster, and from 8,466 to 4,526 when using the data from the second cluster. Moreover, the accuracy of cancer type identification in many cases increases compared to the results obtained in the previous modeling stage using data derived solely from cluster analysis. The results of the comparative analysis of sample classification accuracy using gene expression data obtained through cluster analysis and cluster-bicluster analysis are shown in Figure 3.26. The analysis of the obtained results confirms the earlier conclusion regarding the low classification accuracy of samples from patients without cancer. However, the classification accuracy of these samples is also higher when using the second type of data compared to the data obtained solely from cluster analysis. The classification accuracy of cancer type in most cases is also higher when using gene expression data obtained through cluster-bicluster analysis and gene ontology analysis, indicating the effectiveness of the proposed step-by-step procedure for processing gene expression data and the feasibility of its use in diagnostic systems based on gene expression data.

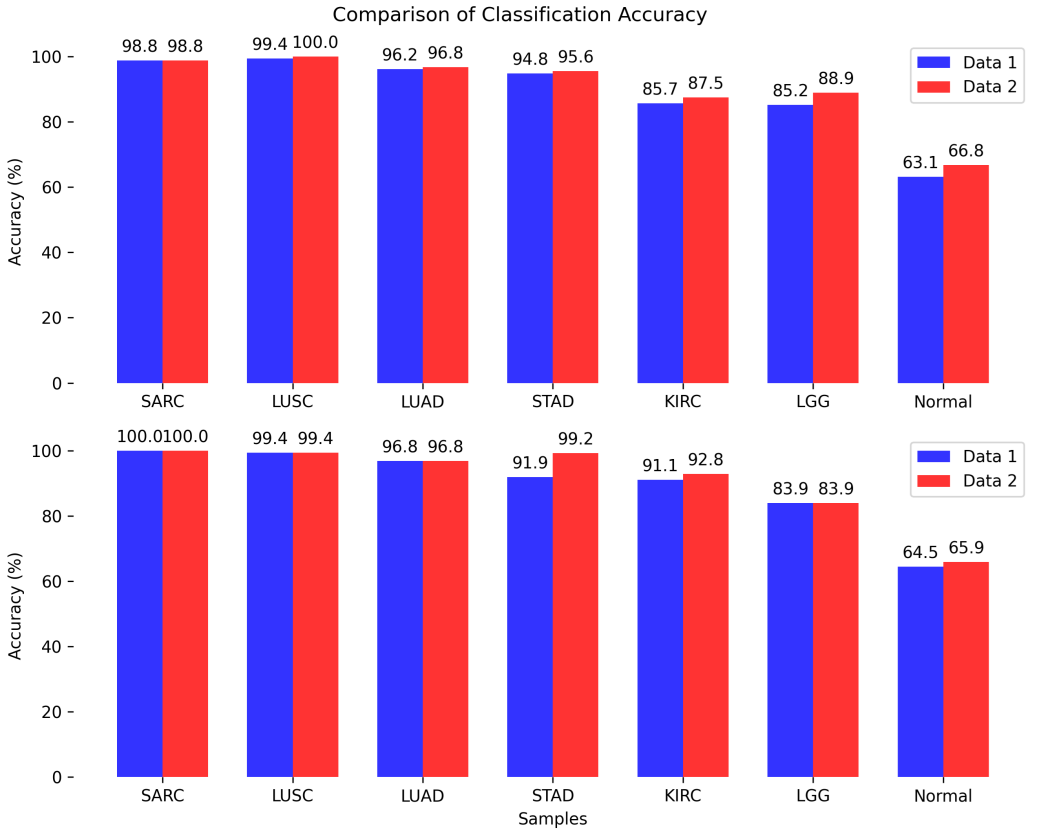

Figure 3.26: Comparative analysis of the samples' classification accuracy based on gene expression data obtained through cluster analysis and cluster-bicluster analysis: the top row presents the analysis results for the first cluster data, and the bottom row presents the analysis results for the second cluster data.

Chapter 4

The Application of Deep Learning Methods in Hybrid Models of Disease Diagnosis Based on Gene Expression Data

This chapter contains parts of the papers [17, 16, 26, 18, 80].

4.1 Introduction

This section presents theoretical and experimental research on applying deep learning methods in hybrid models for processing gene expression data. The appropriateness of using deep learning methods for processing gene expression data is determined by the structure of the experimental data and their large volume. Typically, experimental data contain thousands of objects and more than ten thousand attributes. The advantages of deep learning methods include the ability to process complex and unstructured data. Deep learning algorithms can identify specific patterns in the hierarchical representation of data and form certain functions that allow for high-accuracy identification of the objects being studied.

The second significant advantage of models based on deep learning methods is their high accuracy and performance. Moreover, models based on deep learning methods can derive relevant functions directly from raw data, allowing for discovering hidden patterns and complex relationships in the data, which is problematic with traditional methods. Deep learning-based models are inherently scalable. This means that these models can efficiently scale to process large volumes of data, leveraging parallel or distributed computing architectures, significantly accelerating the

training and inference processes.

In the case of using gene expression data, the correct application of deep learning methods can improve the efficiency of diagnostic systems for complex objects by increasing the accuracy of object identification, on the one hand, and enhancing the objectivity of determining the object's state through parallel processing of information, on the other hand. All of the above highlights the relevance of current research.

4.2 Comparative Analysis of Deep Learning Methods and Models for Objects Identification Based on Gene Expression Data

Currently, there are several deep learning (DL) methods that can be applied to gene expression data to identify hidden patterns and make predictions about the state of the respective object [69]. Figure 4.1 shows a block diagram of the most common deep learning methods aimed at processing gene expression data and analyzing genomic sequences, as well as possible directions for their application.

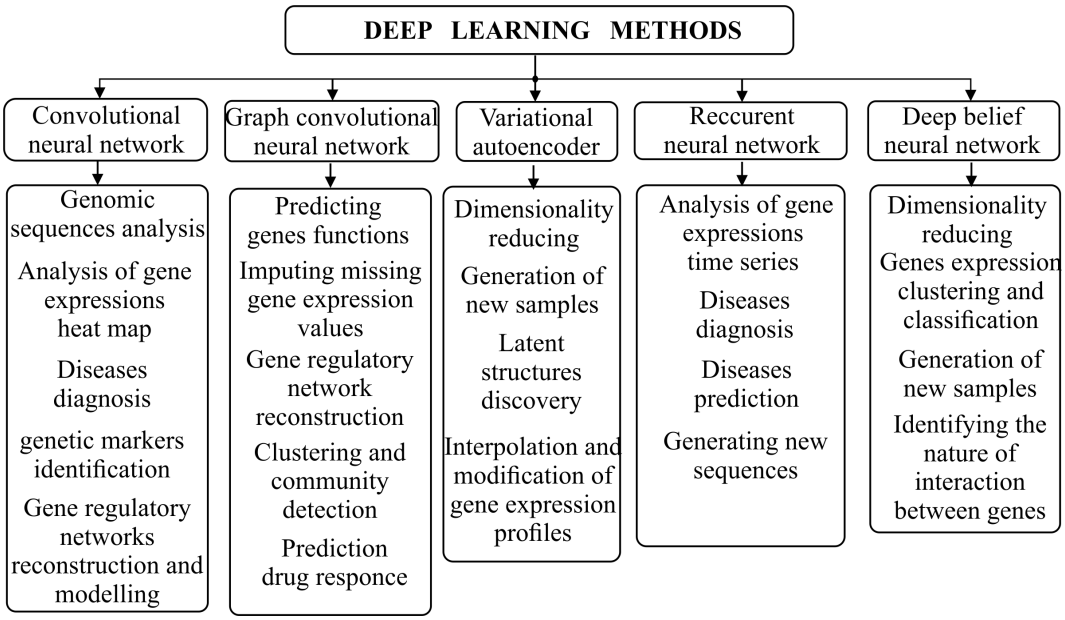

Figure 4.1: Block diagram of existing deep learning methods and their application directions for analyzing gene expression data and genomic sequences

As shown in Figure 4.1, the main DL methods include the following:

1. **Convolutional Neural Networks (CNNs)**. These are used for analyzing gene expression data, which can be represented as vectors (one-dimensional CNNs) or in the form of images or heatmaps (two-dimensional CNNs). The advantages of CNNs include their ability to detect hidden dependencies and form a vector of useful features from genomic data. Depending on the task, the following applications of CNNs can be identified:

    - *Genomic Sequence Analysis:* CNNs can be used to analyze genomic sequences, such as DNA or RNA sequences. They can detect motifs, common binding regions, and other structures in the genome that may indicate functional elements, regulatory mechanisms (transcription factors), or gene types.
    - *Gene Expression Heatmap Analysis:* Gene expression heatmaps represent the distribution of gene expression levels in a two-dimensional space across different samples or experimental conditions. CNNs can analyze these heatmaps, detect correlations between genes, identify groups of genes with similar profiles, and uncover biological processes regulated by these genes.
    - *Disease Diagnosis:* CNNs can be applied to classify gene expression samples based on their expression profiles. They can distinguish between healthy and diseased samples, identify disease subtypes, or predict clinical outcomes based on gene expression data.
    - *Genetic Marker Identification:* CNNs can identify genetic markers associated with specific diseases or phenotypes. They can also detect genes or combinations of genes that are discriminative for different groups of samples and are used for diagnosis, prediction, or other medical applications.
    - *Reconstruction and Modeling of Gene Regulatory Networks (GRNs):* CNNs can help reveal the nature of interactions within GRNs, including interactions between transcription factors and their targets. They can identify regulatory modules, important regulators, and establish mechanisms of gene regulation.

2. **Recurrent Neural Networks (RNNs)** are a powerful tool for analyzing and processing gene expression data, including time series of gene expression values. Typically, when applying gene expression data, RNNs are used to solve the following tasks:

    - *Time Series Analysis:* RNNs, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), can detect dependencies in gene expression time series, model the dynamics of genetic processes, identify

recurring motifs, determine trends, and predict future gene expression values.

- *Disease Prediction:* RNNs can be utilized for predicting diseases or clinical outcomes based on gene expression data. They can identify significant genes or genetic markers correlating with diseases and use this information to predict risks or diagnose diseases.
- *Sample Classification:* Similar to CNNs, RNNs can also be applied to classify gene expression samples based on their expression profiles, identify healthy or diseased samples, determine disease subtypes, or predict responses to specific treatments.
- *Sequence Generation:* RNNs can be employed to generate new gene expression sequences based on trained models. This can be useful for studying variations in expression profiles, generating synthetic data for training models, or exploring potential genetic states.

3. **Graph Convolutional Networks (GCNs)** are a method for processing gene expression data represented as graphs, where genes are nodes and the connections between genes (such as expression correlations or regulatory interactions) are edges. Possible applications of GCNs include:

   - *Gene Function Prediction:* GCNs can use the graph structure of genetic data to predict gene functions. By aggregating information from neighboring genes, GCNs can explore representations of genes that account for their connections and use these representations to predict the functions of unknown genes.
   - *Filling Missing Gene Expression Values:* GCNs can be used to fill in missing values in gene expression data. By propagating information through the graph, GCNs can explore complex interactions between genes and use these interactions to predict missing gene expression values.
   - *Reconstruction of Gene Regulatory Networks:* GCNs can reconstruct gene regulatory networks from gene expression data. They can model the connections between genes as edges in a graph and use this information to investigate regulatory dependencies between genes and identify key regulatory genes.
   - *Clustering and Community Detection:* GCNs can perform clustering and community detection in gene expression graphs. They can reduce the dimensionality of gene expression data, considering their connections, and use the transformed data to group genes with similar expression profiles.
   - *Drug Response Prediction:* GCNs can also be used to predict drug responses based on gene expression data. They can model the connections

between genes and drug targets as edges in a graph and use this information to predict drug efficacy or identify potential drug targets based on gene expression profiles.

4. **Variational Autoencoders (VAEs)** are generative models capable of identifying gene interaction patterns based on low-dimensional representations of gene expression data and generating new samples with similar expression profiles. In the context of processing gene expression data, VAEs can be applied to solve the following tasks:

   - *Dimensionality Reduction:* VAEs can derive a compact representation of gene expression data that retains the essential information about gene expression, allowing for dimensionality reduction and simplification of subsequent analyses.
   - *Generation of New Samples:* VAEs can simulate the data distribution and generate new samples that fit this distribution. This can be useful for expanding the available data volume and conducting virtual experiments.
   - *Detection of Latent Structures:* VAEs can derive latent representations of genes with complex interaction topologies. This can help uncover hidden patterns in gene expression and identify groups of genes with similar functional properties.
   - *Interpolation and Modification of Expression Profiles:* VAEs can form interpolative dependencies between different gene expression profiles for further modification. VAEs can find intermediate expression profiles between two data groups through the latent space and manipulate latent representations to adjust gene expression values.

5. **Deep Belief Networks (DBNs):** DBNs consist of multiple layers of Restricted Boltzmann Machines (RBMs) and can represent the distribution of gene expression data in a hierarchical structure. The applications of DBNs in this context can include:

   - *Dimensionality Reduction:* DBNs can identify important features from a large amount of gene expression data and create a compact representation without significant loss of useful information.
   - *Clustering and Classification:* DBNs can learn to identify complex interrelationships between genes and group them by similarity or classify genes based on their functional properties.
   - *Generation of New Samples:* DBNs can use learned features and relationships between genes to create new samples with similar expression profiles.

- *Detection of Gene Interactions:* DBNs can model complex interactions between genes and identify genetic mechanisms that determine gene expression.

It should be noted that each deep learning method can be applied to different tasks, and the type of experimental data, the objective, and the limitations of the research determine the choice of method. The following factors determine the main differences between existing DL methods:

1. **Network Architecture:** Convolutional Neural Networks (CNNs) specialize in processing both two-dimensional data (images) and one-dimensional data. Recurrent Neural Networks (RNNs) are more suitable for modeling sequential data such as text or time series. Variational Autoencoders (VAEs) are mainly focused on generating new samples based on existing (learned) latent representations. Graph Convolutional Networks (GCNs) specialize in processing data represented as graphs, which requires the reconstruction of the gene network from existing data using certain algorithms at a preliminary data processing stage, complicating the information processing. Each approach has its advantages and disadvantages, creating the potential for improving gene expression data processing by justified hybridization of existing deep learning models to enhance the quality of gene expression data processing.

2. **Types and Size of Input Data:** CNNs typically require a large number of input data samples to achieve high accuracy. Although increasing the number of training epochs is possible, it carries a high risk of overfitting the network, which is unacceptable. RNNs, unlike CNNs, can work with smaller datasets. GCNs require input data to be represented as a graph, necessitating additional research to optimize the graph structure.

3. **Application Tasks:** Each of the hereinbefore described DL methods can be applied to different tasks such as classification, clustering, generation and recommendations, reconstruction, etc. The choice of method is determined by the specific task of gene expression data analysis, such as biomarker identification, health status prediction or disease type identification, gene interaction characterization, and gene regulatory network reconstruction.

Within the framework of current research, the problem of improving the efficiency of a disease diagnosis system based on gene expression data is addressed. The solution involves identifying co-expressed and significant genes in the first stage and classifying objects based on the formed subsets of gene expression profiles in the second stage. This fact limits the number of deep learning methods that can be applied to solve the problem.

The task of classifying objects based on gene expression data can be solved using convolutional or recurrent neural networks. In this case, there is a challenge of determining the optimal network structure and the vector of hyperparameters that determine the network's performance. Identifying subsets of co-expressed gene expression profiles can be achieved using a deep belief network. However, in addition to determining the optimal structure and hyperparameters of the network, there is a challenge of proving its advantage over classical clustering algorithms for gene expression profiles that are currently used in this field.

Graph convolutional neural networks can also be used in classification systems, but their application requires the reconstruction of the gene regulatory network in a preliminary stage to represent it as a graph. This, in turn, necessitates the identification of subsets of co-expressed gene expression profiles by applying clustering procedures to the gene expression data. This process can be implemented by hybridizing the model through the application of different deep learning methods at the respective data processing stages, which, in turn, requires thorough research to evaluate the effectiveness of the corresponding method and determine the optimal network structure and vector of hyperparameters.

4.3 Applying Convolutional Neural Network (CNN) for Gene Expression Data Classification

The general architecture of the multilayer CNN is depicted in Figure 4.2 [89]. Usu-

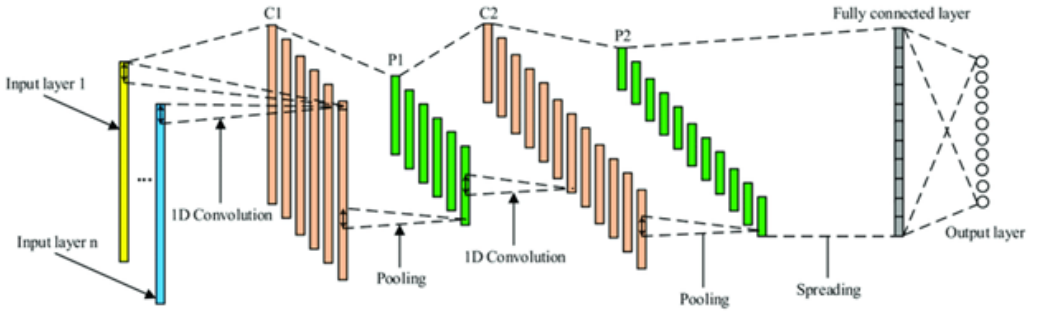

Figure 4.2: The general architecture of the multilayer CNN

ally, it includes the following main components:

1. *Input layer:* Accepts input data, which can be represented as a one-dimensional data vector (a vector of gene expression values that define the state of the object) or a two-dimensional matrix (a heatmap of gene expression values,

images, etc.). Depending on the type of input data, one-dimensional (1D) or two-dimensional (2D) convolutional layers are formed.

2. *Convolutional layers:* Used to detect local features in the input data. Each convolutional layer consists of a set of filters that perform the convolution operation on the input data. Convolution is the basic operation in CNN. Typically, in the convolutional layer, the feature map of the previous layer is a convolution using convolutional kernels, and the nonlinear activation function creates the output feature map. The computational process in this case can be expressed as follows [89]:

$$X_j^l = f(\sum_{i \in M_i} X_i^{l-1} * \omega_{ij}^l + b_j^l) \tag{4.1}$$

where: X_j^l and $X_i^(l-1))$ are the $j-th$ and $i-th$ features of the data at levels l and $l-1$. respectively; M_i is the set of input feature maps (determined by the filter applied to the input data at the corresponding convolutional level); ω_{ij}^l is the convolutional kernel connecting the $i-th$ feature map of input data with the $j-th$ feature map at the convolution level l; b_j^l is the bias; $f(\cdot)$ is a nonlinear activation function, $*$ stands for the convolution operation.

3. *Pooling layers:* These are used to reduce the spatial dimensions of the feature vector or matrix to decrease the number of parameters. The max pooling layer transforms the data vector or matrix into a single value equal to the maximum value from that region.

4. *Fully Connected Layers:* The data is passed to the fully connected layers after several convolutional and pooling layers. Every neuron in a fully connected layer is connected to every neuron of the previous layer. The fully connected layers are used for classification or regression based on the features obtained. They take the features from the flattened layers and generate an output vector that can be presented as the model's output.

5. *Activation Functions:* After each convolutional layer, an activation function is applied. In most cases, these are nonlinear, allowing the network to detect complex dependencies in the data during the learning process.

6. *Loss Function:* It determines the difference between the predicted and expected values. The derivatives of the loss function are used to update the weights and biases in the network during the backpropagation of the error. This allows the model to assess its accuracy and adjust its weights during training.

As mentioned above, the hyperparameters of CNN determine the network's architecture and training parameters. They are set during the initialization of the network and affect its learning and generalization capabilities. Some of the critical hyperparameters of a CNN include:

- *Number of Convolutional Layers:* It defines the number of layers where convolutional filters detect features in the input data. Having more layers can help the model learn more complex dependencies, but it can also lead to greater complexity, longer training times and overfitting of the network. Overfitting can be determined by evaluating the convergence of accuracy values and the loss function calculated on the training and validation data during network training.
- *Size of Convolutional Filters:* It determines the size (width and height) of the filters that move over the input data to perform convolution. Larger filters can detect larger patterns but may also lead to increased computational load.
- *Number of Filters in Convolutional Layer:* It determines the number of filters applied to the input data in each convolutional layer. Each filter generates a feature map corresponding to a specific feature. Typically, the number of filters increases with each subsequent convolutional layer.
- *Size of Pooling Window:* This refers to the window size (width and height) that moves across the feature map to perform pooling operations.
- *Activation Function:* This is the function used to introduce non-linearity in the network after each layer.
- *Number of Fully Connected Layers:* This determines the number of fully connected layers that should be added after the convolutional and pooling layers. These layers connect every neuron in one layer to every neuron in the next layer. They are typically used for classification or regression based on the features extracted by the preceding layers.

Within the framework of the current research, the optimal combination of CNN hyperparameters was determined using the ordered empirical grid search method by evaluating all possible combinations of hyperparameter values within predefined ranges. The implementation of this procedure involves the following stages:

1. Definition of the range of hyperparameter values variation that are subject to optimization.
2. Determination of the metric for evaluating the efficiency of a particular combination of hyperparameter values during their sequential enumeration. Since

the current research involved classifying objects based on gene expression data, metrics based on the assessment of type I and type II errors were applied [14]:

- *Classification Accuracy* – determines the proportion of the total number of samples that are correctly identified (Formula 2.13).
- *F1-score* is a measure to identify the correctness of the samples distribution into the relevant class and is calculated as the harmonic mean of precision (PR) and recall (RC) (Formula 2.14).

3. Creation of a grid of all possible combinations of hyperparameters within the range specified in item 1. Each cell of this grid structure represents a unique combination of the model's hyperparameters.

4. For each combination of hyperparameters:

   4.1. Construct a neural network model, the architecture and parameters of which correspond to the current combination of hyperparameters.

   4.2. Training, validation, and testing of the model.

   4.3. Calculation of the quality criteria for sample identification according to formulas (2.13) and (2.14).

   5. Analysis of the values of the obtained quality criteria for sample classification. Selection of the combination of hyperparameters that corresponds to the maximum values of the sample classification quality criteria.

It should be noted that a drawback of the empirical grid search method is the significant computational time it requires. However, it ensures systematic exploration of the hyperparameter space. It helps to choose the optimal combination for the neural network model, considering both the research objective and the type of experimental data.

4.3.1 Experimental Studies on Optimizing Hyperparameter Values of CNN Using Gene Expression Data

The modeling process was carried out using gene expression data from patients who were studied for various types of cancer diseases. The data is freely available in The Cancer Genome Atlas (TCGA) [3]. Gene expression data obtained on the Illumina platform was used by applying the method of RNA molecules genomic sequencing, and for each sample, the number of respective genes determining the state of the sample under study was identified. In the initial state, the experimental data contained 3269 samples and 19947 genes. Table 4.1 presents the classification of

Table 4.1: Classification of experimental gene expression data used in the modeling process

No	Type of Cancer	Number of Samples
1	Adrenocortical carcinoma (ACC)	79
2	Glioblastoma multiforme (GBM)	169
3	Sarcoma (SARC)	263
4	Lung squamous cell carcinoma (LUSC)	502
5	Lung adenocarcinoma (LUAD)	541
6	Stomach adenocarcinoma (STAD)	415
7	Kidney renal clear cell carcinoma (KIRC)	542
8	Brain Lower Grade Glioma (LGG)	534
9	Normal (No cancer detected)	224

experimental data, including the type of disease and the number of samples corresponding to each type of disease. The data also includes the number of samples for which no cancerous tumor was detected (healthy patients). The gene expression values in the data presented in Table 4.1 determine the level of its activity (the intensity of the protein synthesis process corresponding to this type of gene) and are proportional to the number of genes of the corresponding type. In the first stage, the absolute values of the number of genes were transformed into a more convenient range for further processing (Count Per Million – CPM) according to the formula:

$$CPM_{ij} = \frac{count_{ij}}{\sum_{j=1}^{m} count_{ij}} \cdot 10^6 \tag{4.2}$$

where: $count_{ij}$ is the count of genes of the $j-th$ type corresponding for the $i-th$ sample; m is the total number of different types of genes studied during the experiment performing.

The implementation of this step significantly reduced the range of variation in absolute values, determining the expression (activity level) of respective genes. In the second stage, data normalization was performed by applying the $log_2(CPM)$ function to all values. In the third stage, non-expressed genes were removed according to the condition $log_2(CPM) \leq 0$ for all samples under study. The number of genes at this stage was reduced by 682, and the matrix of experimental gene expression data took the form: $E = (3269 \times 19265)$. In the final stage, negative gene expression values were replaced with zeros, representing non-expressed genes for some samples. For proper initialization of CNN filters, the number of gene expression profiles was increased to 19,300 by supplementing with profiles with zero expression.

At this stage, the reduction of the number of genes based on statistical and entropy criteria according to the methodology presented in the second chapter of this work was not performed, since the main objective of the current study is to

optimize the hyperparameters of CNN and to compare different types of neural network models. This can be achieved by using the full set of gene expression data.

4.3.2 Simulation of 1-D Convolutional Neural Network

Figure 4.3 depicts the flowchart of a 1-D single-layer CNN with relevant hyperparameters at different stages of the neural network's operation.

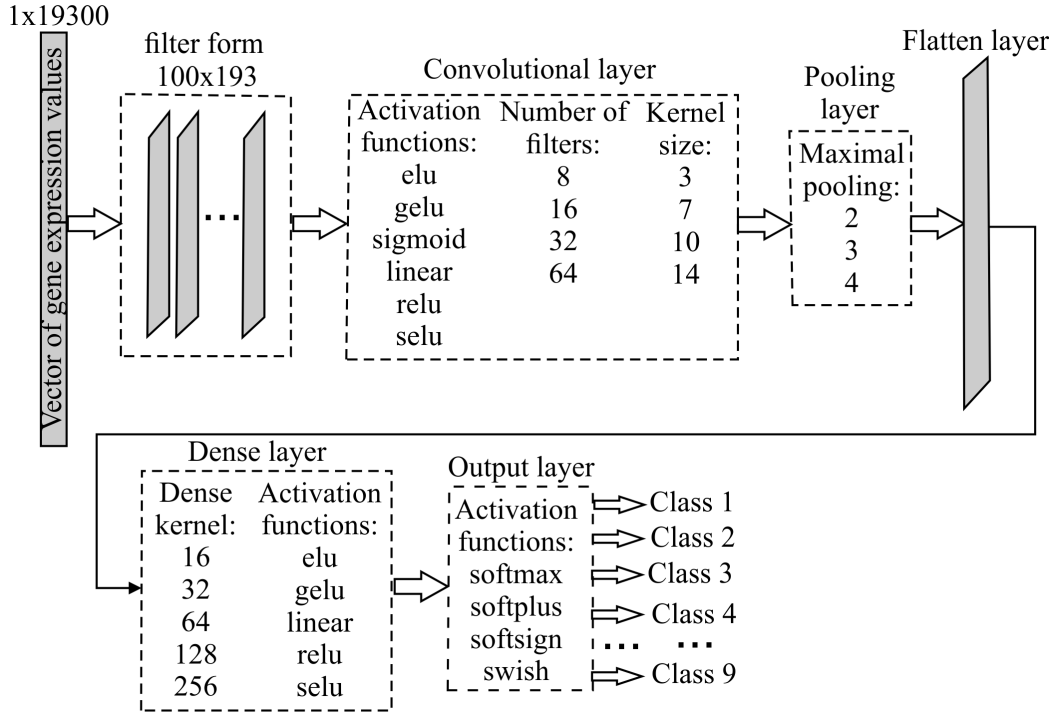

Figure 4.3: Flowchart of a 1-D single-layer CNN for determining the optimal hyperparameter vector of the neural network

Figure 4.4 presents the activation functions investigated during the modeling process implementation. To determine the optimal combination of hyperparameters within the grid search concept, a heuristic search algorithm is proposed, the pseudocode of which is given below (Algorithm 5).

As heuristic functions, the overall classification accuracy across all classes (coarse evaluation) and the accuracy of sample distribution across single classes by calculating the F1-score for each class (detailed analysis) were used. Considering that with a large number of classes, analyzing the F1-score values for the respective classes to choose the optimal alternative from the hyperparameter list can be problematic, the integral F1-score value was calculated based on the values obtained in the previous

Algorithm 5: Optimization of Hyperparameters for Neural Networks Using Gene Expression Data

Data: Gene expression data matrix
Start;
Form the matrix of gene expression data;
Form the list of hyperparameters $g = 1, \ldots, g_{\max}$;
Form the vector of values for each hyperparameter $k = 1, \ldots, k_{\max}$;
Divide the gene expression data into subsets:

- for training the model;
- for validating the model;
- for testing the model.

while $g \leq g_{max}$ **do**
  while $k \leq k_{max}$ **do**
    Train the neural network on the training data;
    Validate and test the model;
    Calculate classification quality criteria:
      - accuracy on test data;
      - loss function values;
      - F1-score for each class.
  end
  Calculate the integral F1-score;
  Analyze results and fixation of the optimal hyperparameter value;
end
Fixation of the list of optimal hyperparameter values;
Result: Optimal hyperparameter values for neural networks

Exponential linear unit (elu): $elu(x, \alpha = 1.0) = \begin{cases} x, if\ x > 0 \\ \alpha \cdot (\exp(x) - 1), if\ x < 0 \end{cases}$	Gaussian error linear unit (GELU) : $gelu(x) = x \cdot P(X \leq x), where\ P(X) \sim N(0,1)$	Rectified linear unit (relu): $relu(x) = \max(x, 0)$
Linear activation function: $linear(x): f(x) = x$	Scaled exponential linear unit (selu): $selu(x) == \begin{cases} scale \cdot x, if\ x > 0 \\ scale \cdot \alpha \cdot (\exp(x) - 1), if\ x < 0 \end{cases}$, where: *scale = 1.05070098*; $\alpha = 1.67326324$	
Sigmoid activation function: $sigmoid(x) = \frac{1}{1 + \exp(-x)}$	Softmax activation function: $softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}$	Softplus activation function: $softplus(x) = \log(\exp(x) + 1)$
Softsign activation function: $softsign(x) = \frac{x}{abs(x)+1}$	Swish activation function: $swish(x) = x \cdot sigmoid(x)$	Hyperbolic tangent: $\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$

x is the input tensor

Figure 4.4: The activation functions investigated during the simulation process implementation

step using Harrington's desirability function. The algorithm for implementing this procedure includes the following steps:

I. **Initialization.**

1.1. Represent the F1-score values as a matrix, where the rows are classes, and the columns are the hyperparameter values being investigated at this stage.

2. **Calculation of private desirabilities.**

2.1. Determine the minimum and maximum F1-score values at the respective stage of the neural network's operation (when using the corresponding hyperparameter values combination).

2.2. Transform the F1-score values to a linear scale of a dimensionless parameter Y considering the boundary F1-score values determined in the previous step (the Y parameter values according to the desirability method is varied within the range from $Y_{min} = -2$ to $Y_{max} = 5$). At the first step,

the coefficients of the linear equation are calculated:

$$\begin{aligned} Y_{min} &= a + b \cdot F1_{min} \\ Y_{max} &= a + b \cdot F1_{max} \end{aligned} \tag{4.3}$$

2.3. At the second step, the F1-score values are directly transformed into Y values:

$$Y = a + b \cdot F1 \tag{4.4}$$

2.4. Calculate the private desirabilities for each F1-score value:

$$d = \exp\left(-\exp\left(-Y\right)\right) \tag{4.5}$$

3. **Calculation of the integrated F1-score value.**

3.1. For each column of the matrix obtained in step 2, calculate the integrated F1-score value as the geometric mean of all private desirabilities:

$$F1_{int}^{j} = \sqrt[9]{\prod_{i=1}^{9} d_{ij}} \tag{4.6}$$

where j denotes the respective column of the private desirabilities matrix.

4. **Analysis of the obtained results.**

4.1. Create a diagram showing the dependence of the integrated F1-score value on the corresponding hyperparameter values. Choose the optimal hyperparameter value that corresponds to the maximum value of the integrated F1-score.

For the output layer of neurons, the following activation functions were used: softmax, softplus, softsign, and swish. For the fully connected hidden layer of neurons, the following activation functions were sequentially applied: elu, gelu, linear, relu, and selu. Other activation functions applied to this layer showed unsatisfactory results. For the hidden convolutional layer, the following activation functions were used: elu, gelu, sigmoid, linear, relu, and selu.

In the initial data preprocessing stage, the data was split into two subsets in a 0.7/0.3 ratio (2,288/981 samples). The first subset (2,288 samples) was further divided into two subsets in a 0.8/0.2 ratio (1830/458). 1,830 samples were used for training the network, 458 for validating the model during its training, and 981 samples were used for testing the model. The model's quality assessment was based on the analysis of the loss function value calculated during the model's validation, the classification accuracy, and the F1-score value calculated when applying the test

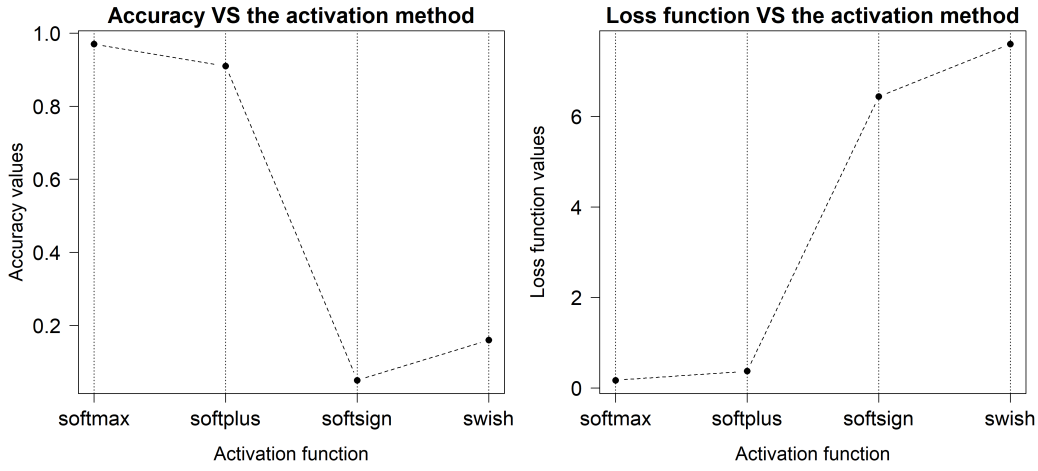

Figure 4.5: Distribution diagrams of classification quality criteria when determining the optimal activation function for the output layer of neurons in the neural network model (CNN)

data. The simulation results for determining the optimal activation function of the neurons' output layer are shown in Figure 4.5.

As seen from Figure 4.5, the use of the *softmax* function allows obtaining the best classification results for samples in terms of accuracy, which was calculated on the test data subset, and in terms of the loss function, which was calculated on the validation data. When using the *softplus* function, the classification results are slightly worse. When using other functions, the classification results are unsatisfactory. This conclusion is confirmed by the analysis of F1-score values calculated for each of the nine classes. Due to the clear results obtained, the diagrams showing the dependence of the F1-score values on the type of activation function used are not shown.

Figure 4.6, the simulation results concerning determining the optimal activation function for the CNN's neurons for the dense layer are presented. The analysis of the simulation results allows concluding that in terms of sample classification accuracy (Figure 4.6a) and loss function value (Figure 4.6b), the optimal activation functions are *elu* and *selu*, which to some extent does not match the results based on the analysis of F1-score values (Figure 4.6c,d). The analysis of the integrated F1-score criterion values (Figure 4.6d) allows concluding that the highest values of this criterion correspond to *relu* and *gelu* functions. Slightly lower values are achieved when using the *selu* function. The analysis of the distribution character of the F1-score values for individual clusters (Figure 4.6c) confirms this conclusion. Thus,

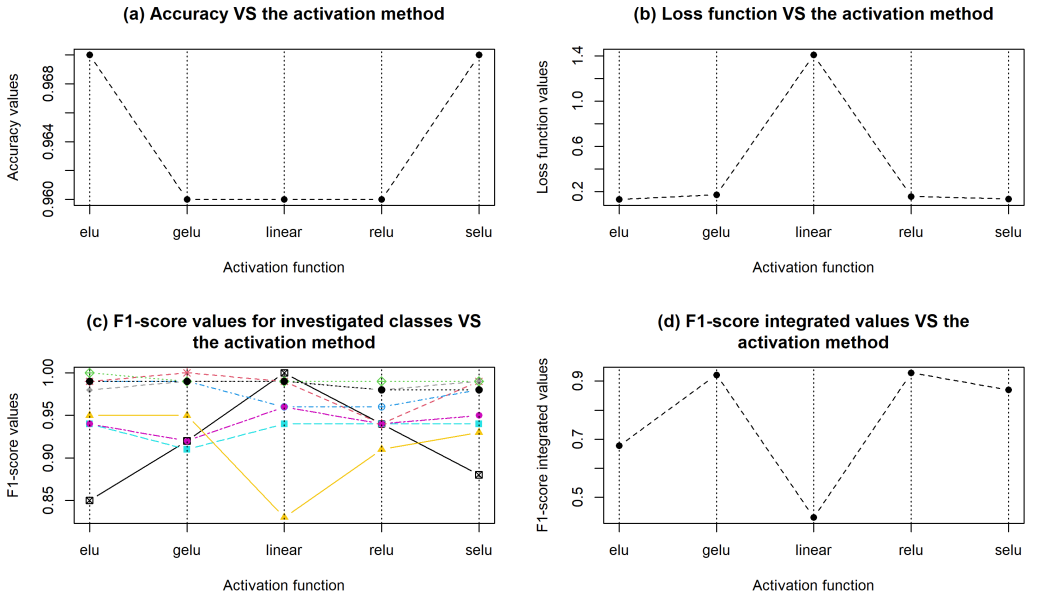

Figure 4.6: Simulation results for determining the optimal activation function of the dense layer neurons: (a) – classification accuracy of samples calculated on the test data subset; (b) – loss function value calculated on the validation data subset; (c) – F1-score value calculated for each class on the test data subset; (d) – integrated F1-score value

based on the analysis of the values of all the criteria, the *selu* activation function was determined as optimal one at this stage of research.

In Figure 4.7, the simulation results for the selection of the optimal activation function for the neurons of the convolutional layer are shown. As can be seen, in terms of sample classification accuracy and the integrated value of the F1-score, the *sigmoid* and *linear* activation functions are optimal ones. However, in terms of the loss function value, the *sigmoidal* function is more preferable.

In Figures 4.8 - 4.11, similar results are depicted for determining other types of CNN's optimal hyperparameters. The analysis of the obtained results suggests that in terms of the classification accuracy of the samples (Figure 4.8a) and the integrated value of the F1-measure (Figure 4.8d), the optimal value of the hyperparameter maximal pooling could be 2 or 3. However, in terms of the loss function value, 2 value corresponds to better results.

Analysis of the dependency diagrams for classification quality criteria of samples on the dense layer neuron kernel size (dense kernel), shown in Figure 4.9, indicates that selecting the optimal kernel size based on the F1-score is problematic, as the results for sizes 32, 64, 128, and 256 are almost indistinguishable (Figure 4.9d).

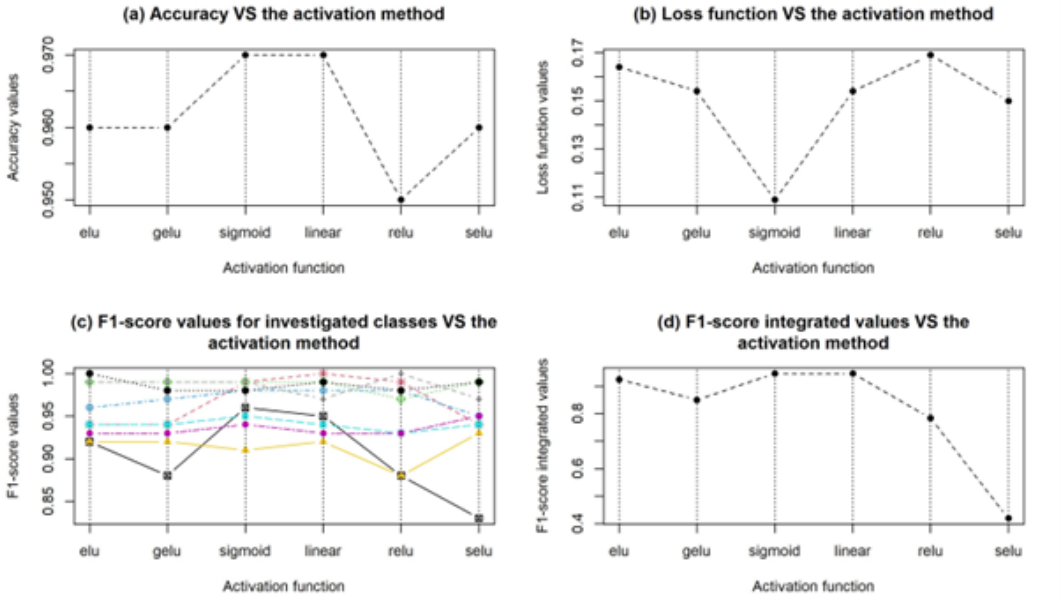

Figure 4.7: Simulation results for determining the optimal activation function for the neurons of the convolutional layer of the CNN model

According to the classification accuracy criterion, the optimal values are 64 and 128 (Figure 4.9a). Based on the loss function value, 64 is more preferable in this case (Figure 4.9b).

The analysis of the simulation results shown in Figure 4.10 allows us to conclude that, according to all classification quality criteria, the optimal kernel size for the convolutional layer neurons is 3.

The analysis of the distribution diagrams for classification quality criteria at different numbers of convolutional layer filters (Figure 4.11) shows that, based on the classification accuracy criterion, 8 and 32 filters are optimal. Regarding the loss function value, 32 filters result in slightly lower losses. The integrated F1-score value indicates a slightly higher attractiveness when using eight filters. In this case, a compromise decision was made to use 32 filters, as reducing the number of filters could lead to decreased sensitivity of the CNN, which is unacceptable within the framework of the dissertation research.

The obtained simulation results allowed for forming a list of optimal hyperparameters for a 1D single-layer CNN, the values of which are presented in Table 4.2.

The next step of the simulation is to compare the efficiency of 1D single-layer, double-layer, and triple-layer CNNs using the hyperparameters determined in the

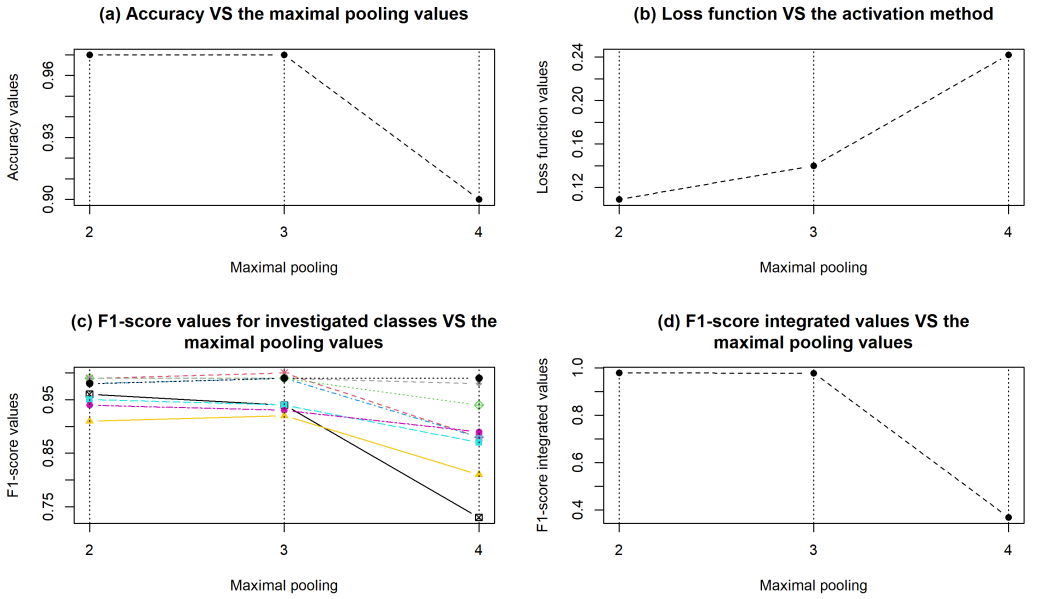

Figure 4.8: Results of simulation to determine the optimal value of maximal pooling for neurons of the convolutional layer

Table 4.2: Optimal hyperparameters values for a 1D single-layer CNN

Number of filters	Kernel size	Dense kernel	Maximal pooling	Activation function of convolutional layer	Activation function of dense layer	Activation function of output layer
32	3	64	2	sigmoid	selu	softmax

previous modeling stage. In this process, a filter of size $(100 - times193)$ was applied to the gene expression value vector in the first convolutional layer, (50×386) in the second, and (25×772) in the third. The simulation results are presented in Figure 4.12. The training time for the model was the same in all cases – 83 seconds. Figure 4.12a also shows the total number of samples that comprised the test data subset and the number of samples correctly identified in each case. Analysis of the obtained results allows us to conclude that, according to all criteria, the single-layer neural network has higher efficiency for this type of data. The number of correctly identified samples is 955 out of 981. The classification accuracy is 97.3%. The F1-score value, calculated for all classes using the single-layer network, is also higher compared to the double-layer and triple-layer networks. The loss function value, calculated using the validation data in this case, is also minimal.

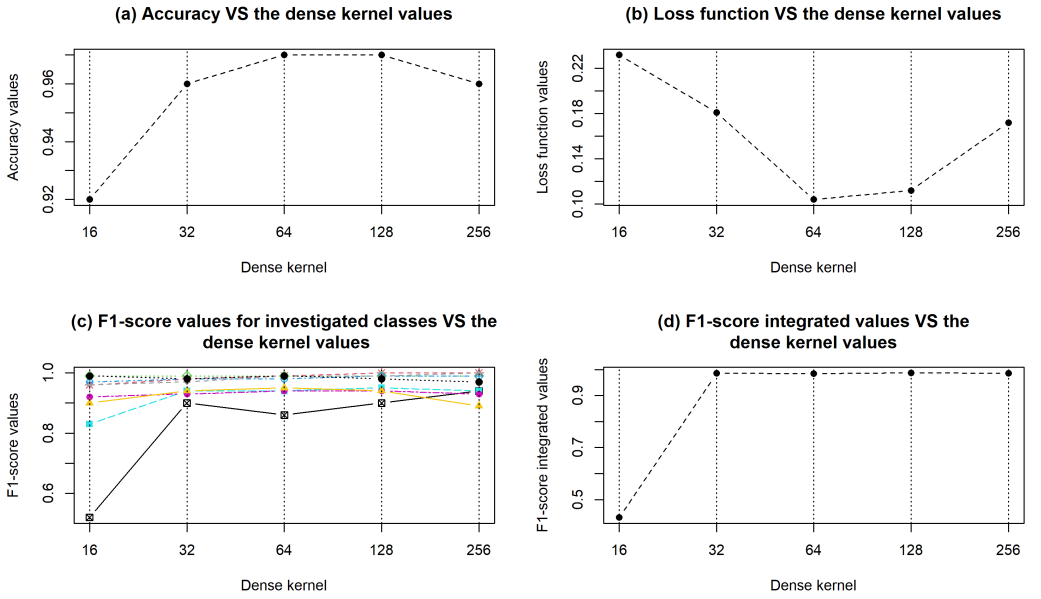

Figure 4.9: Results of simulation to determine the optimal dense dense kernel value

4.4 Applying Recurrent Neural Network (RNN) for Gene Expression Data Classification

The Recurrent Neural Network (RNN) represents a neural network architecture designed to handle sequential data, including text, time series, and speech. The foundational concept of an RNN involves its ability to maintain connections to prior states, thereby enabling the model to preserve information about preceding sequence elements [90, 7, 43]. In general, the RNN architecture is composed of several pivotal components that facilitate the processing of sequential data and the uncovering of hidden dependencies in sequences, ultimately aiming to enhance the precision of identifying objects, the attributes of which are vectors of input data presented to the network:

- **Input Layer**: The Recurrent Neural Network (RNN) receives a sequence of input data, which can be represented as a vector or a matrix.
- **Recurrent Layer**: The recurrent layer, being a pivotal component of an RNN, processes sequential data while retaining and updating a hidden state at each data processing step. It's noteworthy that a distinctive feature of RNNs is that the hidden state at step t encompasses information from both the preceding step t-1 and the current input signal. The recurrent layer applies

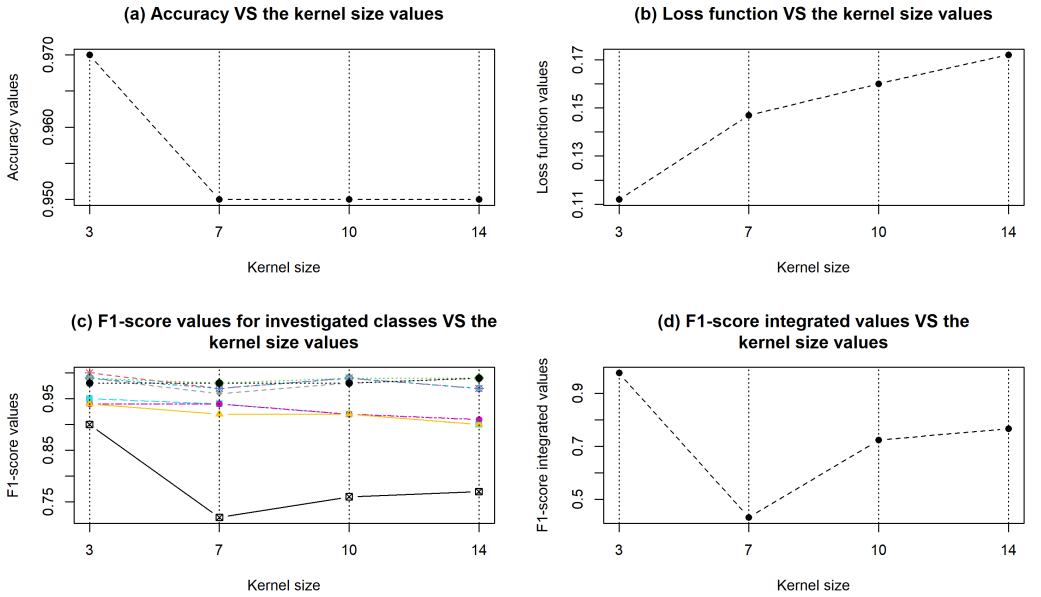

Figure 4.10: Results of simulation to determine the optimal kernel size for the convolutional layer neurons

an appropriate activation function to transform the combined input at each step.

- **Output Layer**: After the input sequence is processed through the recurrent layer, the final hidden state conveys the formulated information to the output layer to produce the desired outcome. Depending on the task, the output layer can have various structures.
- **Feedback among Neurons of Hidden Layers**: Upon obtaining the output, losses are computed by applying a loss function to compare the predicted output with the actual result. The error is then backpropagated to update the weight coefficients of the recurrent layer and optimize the model.

Generally, the mathematical model of an RNN can be depicted as follows:

$$\begin{aligned} h(t) &= f_{act}^{h}(w_{ih}x(t) + w_{hh}f(t-1)) \\ y(t) &= f_{act}^{0}(w_{h0}f(t)) \end{aligned} \tag{4.7}$$

where: $x(t)$ is the vector of input data; $y(t)$ is the vector of output data (classes); f_{act}^{h} and f_{act}^{0} are the activation functions for the hidden and output layers, respectively;

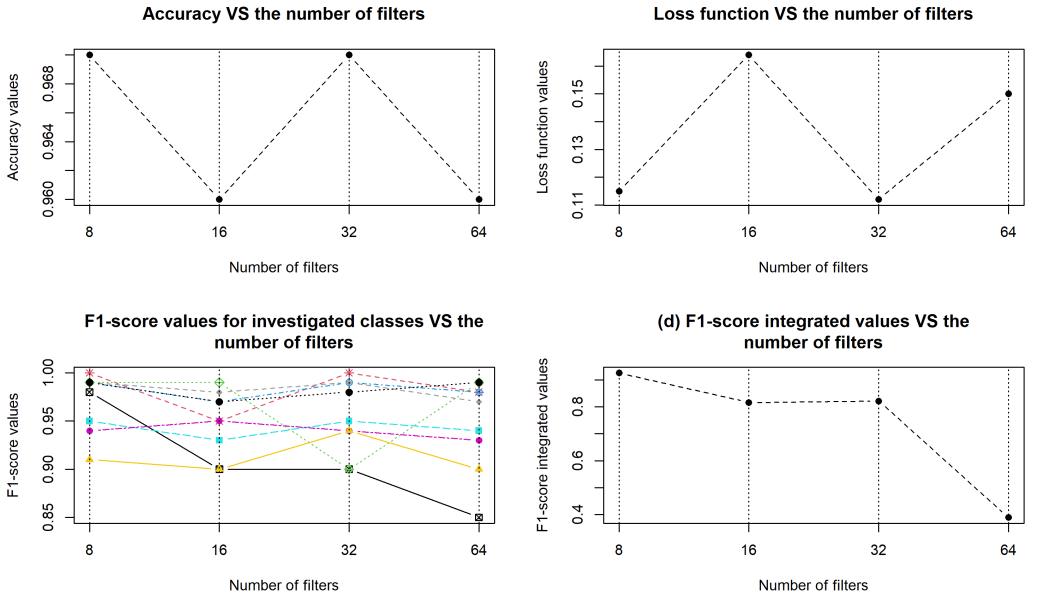

Figure 4.11: Results of simulation to determine the optimal number of filters for the convolutional layer neurons

w_{ih}, w_{hh} *and* w_{h0} are the weight coefficient matrices for the input to first hidden, between hidden, and last hidden to output layers, respectively; $h(t-1)$ and $h(t)$ are the output values of the neurons in the hidden layers at steps $t-1$ and t, respectively.

The primary drawback of a simple Recurrent Neural Network (RNN) is the presence of the vanishing gradient problem, which complicates the processing of high-dimensional gene expression profiles for uncovering hidden patterns. To address this issue, more complex variants of RNNs have been developed, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, which incorporate specialized mechanisms for tackling the vanishing gradient problem and efficiently detecting hidden dependencies. Consequently, within the scope of the current research, LSTM and GRU RNNs are explored.

It should be noted that compared to another type of neural network, likely a Convolutional Neural Network (CNN), an RNN has a shorter list of hyperparameters, simplifying the formation of a list of optimal hyperparameters through grid search. The primary hyperparameters that determine the performance efficiency of RNNs include:

- The number of recurrent (hidden) layers.
- The number of neurons in the recurrent layers.

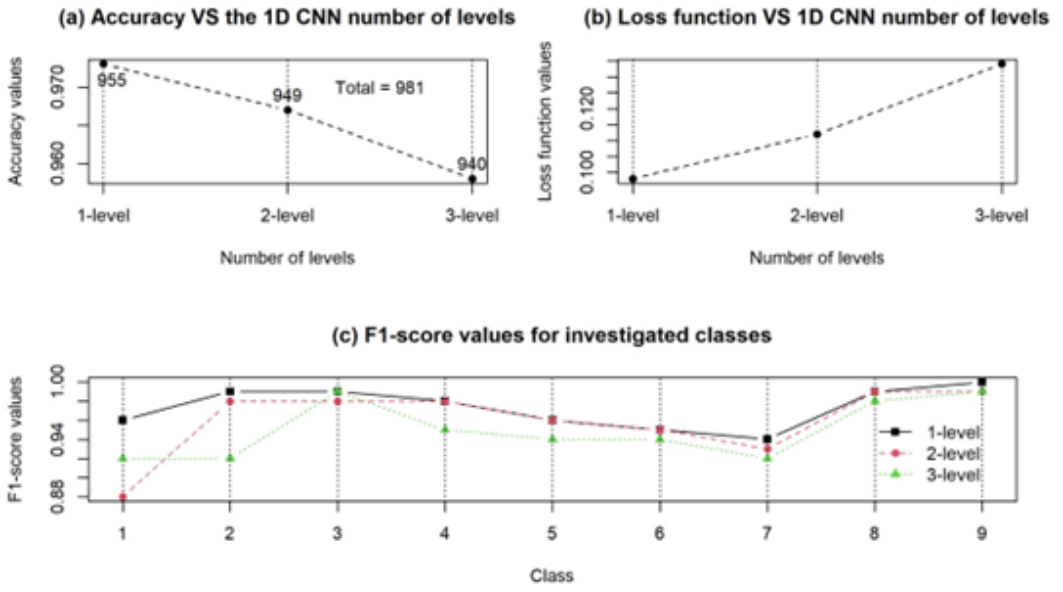

Figure 4.12: Simulation results for determining the number of convolutional layers in 1D CNN

- Activation functions for the recurrent and output layers. Typically, as with [the previous type of network mentioned], *softmax* activation function is used for the neurons in the output layer. For the neurons in the recurrent layers, *sigmoid*, *tanh*, and *relu* activation functions might be utilized.

Results from preliminary modeling indicated that when using both models of RNNs (LSTM and GRU), the hyperbolic tangent (*tanh*) activation function is substantially more effective than *relu* and *sigmoid* activation functions, based on sample classification criteria that comprise the experimental database. Therefore, the modeling process envisaged optimizing two RNN hyperparameters: the number of neurons in the recurrent layers and the number of recurrent layers. The procedure for forming the RNN optimal hyperparameter vector was carried out according to an algorithm, the implementation of which involves the following stages:

Stage I. Data Formation and Algorithm Parameter Adjustment.

1.1. Presenting the gene expression data as a matrix $E = (e_{ij})_{n\times m}$, where n is a number of rows or samples under investigation; m is a number of genes, the expression values of which determines the state of the respective samples.

1.2. Formation the list of hyperparameters for optimization, their range, and step of change during the algorithm operation: $layers = \overline{1,3}$ (number of neural layers in the recurrent layer of RNN); $k = 30, \ldots, 80,\ dk = 5$ (range and step change of the number of neurons in recurrent layers).

1.3. Dividing the set of gene expression data samples into two subsets in a 0.7/0.3 ratio, where the first subset E_{train} is used for model training and the second E_{test} - for testing.

1.4. Splitting the training subset E_{train} further into two subsets in a 0.8/0.2 ratio, where the first subset E'_{train} is used directly for training and the second E_{valid} - for model validation during training. Ensuring the model does not overfit is controlled by monitoring the convergence nature of classification accuracy and the loss function values, calculated on training and validation subsets during the model training.

Stage II. Algorithm Operation within the Hyperparameters Adjustment Range.

2.1. Initializing the number of neural layers in the recurrent layer: $levels = 1$.

2.2. Initializing the starting value of the number of neurons in the recurrent layers: $k = 30$.

2.3. Model training. At each training step, calculating the classification accuracy and the loss function value on the data subsets for training and validation.

2.4. Testing the model on the test data subset. Calculating the samples' classification accuracy, F1-score for each class.

2.5. If $k < k_{max}$, increase the number of neurons in the recurrent layers by 5 (k = k + 5) and escape to step 2.3 of this procedure. Otherwise, calculate the integrated F1-score value, analyze the obtained results and forming the optimal decision regarding the number of neurons in the recurrent layers at this stage.

2.6. If the number of recurrent layers is less than the maximum number ($layers < layers_{max}$), increase the number of layers by 1 and and go to step 2.2 of this algorithm. Otherwise, proceed to Stage III.

Stage III. Analysis of the Obtained Results and Formulating an Optimal Solution.

3.1. Comparative analysis of the solutions obtained in the previous algorithm operation stage. Forming the optimal decision regarding the hyperparameter vector for the corresponding type of RNN.

4.4.1 Modeling of LSTM Recurrent Neural Network

In Figures 4.13 - 4.15, the simulation results to determine the optimal hyperparameters of the LSTM recurrent neural network are depicted. Single-layer, two-layer,

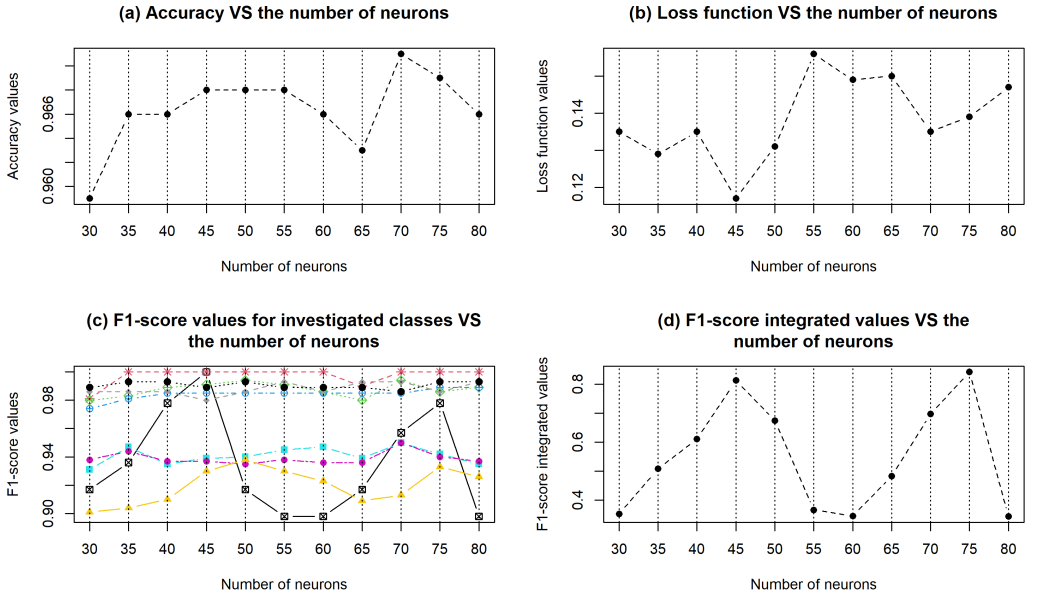

Figure 4.13: Results of the modeling when applying a single-layer LSTM recurrent neural network

and three-layer neural networks were investigated during the simulation process. As the simulation results showed, increasing the number of layers when applying gene expression data is not advisable since the network's performance quality decreased according to the used criteria, while its propensity for overfitting increased due to enhanced complexity. To reduce the likelihood of network overfitting, 20% of neurons were zeroed out after each layer.

Analyzing the modeling results allows concluding that in all cases, the accuracy of classifying samples comprising the test data subset varies within a quite narrow range: from 95% to 97%. This indicates the high quality of the RNN's performance in classifying gene expression data. A more detailed analysis of the obtained diagrams indicates a higher efficiency of a two-layer LSTM recurrent neural network with 35 neurons in the recurrent layers, according to all utilised quality criteria. This RNN model was used in subsequent studies.

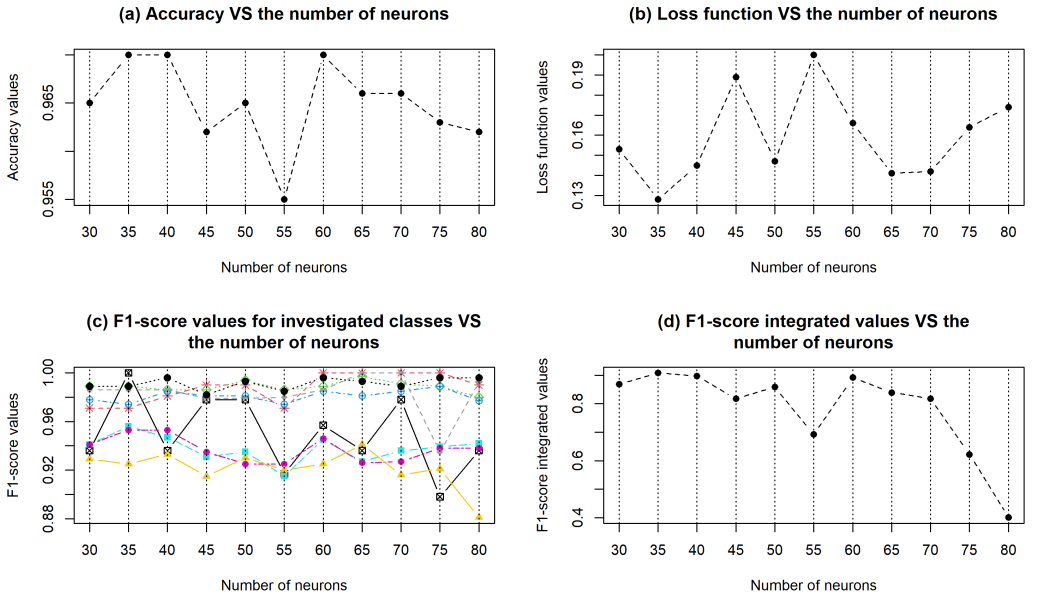

Figure 4.14: Results of the modeling when applying a two-layer LSTM recurrent neural network

4.4.2 Modeling of GRU Recurrent Neural Network

In Figures 4.16 - 4.18, the modeling results using a GRU recurrent neural network are depicted. Analysis of the obtained results also indicates the high effectiveness of this type of RNN for classifying data based on gene expression. However, compared to the LSTM network, a single-layer GRU neural network is more appealing both in terms of stability and quality. With 55 neurons in the recurrent layer, utilizing this type of network allows for achieving a classification accuracy of 96.9% for the samples of the test data subset, with a loss function value of 0.138 and a relatively high density of variation in the F1-measure values across individual classes of the test data subset (ranging from 0.922 to 1). The integrated F1-measure value was 0.944 in this case.

4.4.3 Calculating the Comprehensive Quality Criterion for the Classification of Gene Expression Data

Analysis of the simulation results, presented hereinbefore, indicates challenges in determining the optimal architecture and hyperparameters of the neural network based on the combination of classification quality criteria used during the simulation process. The values of these criteria can be contradictory. Moreover, even a

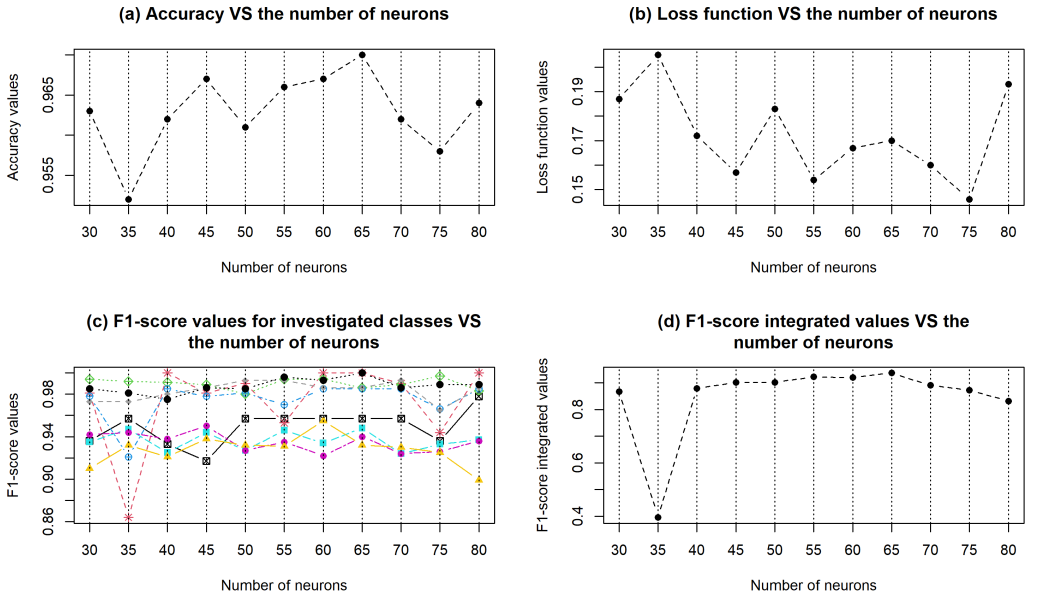

Figure 4.15: Results of the modeling when applying a three-layer LSTM recurrent neural network

small difference in values can somewhat complicate selecting a list of optimal neural network hyperparameters. In this case, it is advisable to calculate a comprehensive quality criterion based on calculated individual criteria, such as sample classification accuracy, loss function value, and integrated F1-score value. Notably, higher accuracy and F1-score values and a lower loss function value correspond to a higher quality level of the model, i.e., an optimal network type and list of its hyperparameters. The calculation of the comprehensive quality criterion was performed using the weighted average method:

$$QC_w = \sum_{k=1}^{n} weight_k \cdot QC_k \tag{4.8}$$

Here: $weight_k$ denote the weight of the corresponding k-th quality criterion (QC_k).

The algorithm for calculating the criterion (4.8) within the current research entails the following steps:

1. Inverting the loss function value into a vector of values that increase with the model's attractiveness:

$$loss'_k = max(loss) - loss_k \tag{4.9}$$

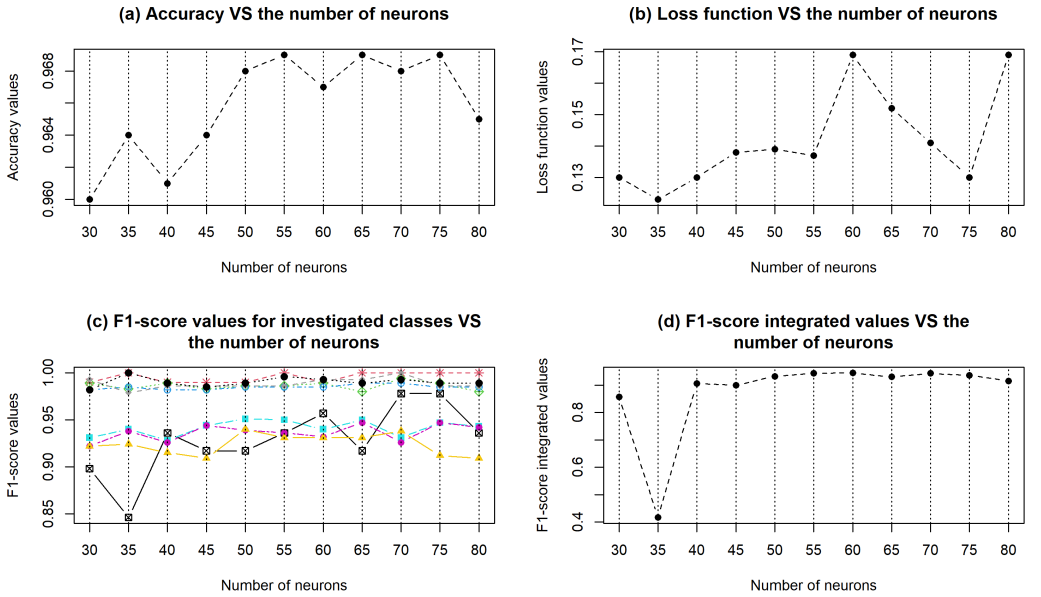

Figure 4.16: Results of the modeling when applying a single-layer GRU recurrent neural network

2. Scaling the values of all criteria within the range $[0, 1]$:

$$QC_k^{norm} = \frac{QC_k - min(QC)}{max(QC) - min(QC)} \tag{4.10}$$

3. Initializing the weight vector for the utilized criteria: When calculating the comprehensive quality criterion for classification, it was assumed that the weight of the loss function value, calculated on the data for model validation of the neural network, is half as much as the weights of the accuracy and integrated F1 score, which are calculated on the test data subset. Therefore, the weight vector for the criteria vector $(ACC; F1_int,\ loss')$ was initialized as follows: $w = (0.4,\ 0.4,\ 0.2)$.

4. Calculating the value of the comprehensive criterion using formula (4.8):

$$QC_k^{compr} = w[1] \cdot ACC_k^{norm} + w[2] \cdot F1_k^{norm} + w[3] \cdot loss_k^{norm} \tag{4.11}$$

A higher value of this criterion corresponds to a better alternative.

The proposed methodology was tested using results obtained in previous subsections during the simulation of LSTM and GRU recurrent neural networks with

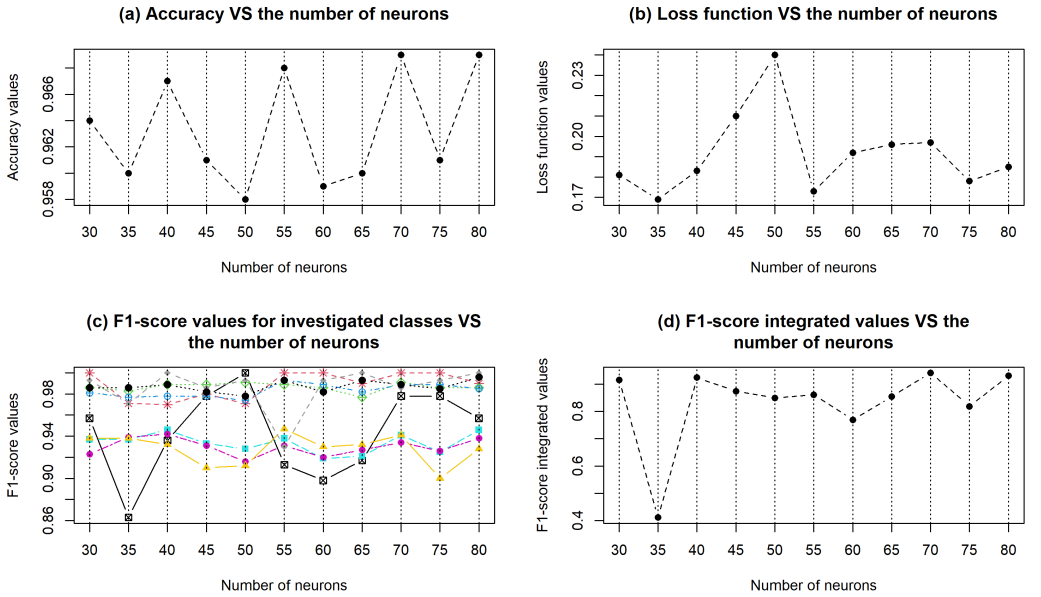

Figure 4.17: Results of the modeling when applying a two-layer GRU recurrent neural network

various sets of hyperparameters. Figures 4.19 and 4.20 illustrate the distribution diagrams of the comprehensive quality criterion value for the performance of LSTM and GRU recurrent neural networks when using varying numbers of neurons and different amounts of recurrent layers.

The analysis of the obtained simulation results allows us to conclude that, in the case of using the LSTM model, a two-layer RNN with 35 neurons in the recurrent layer is optimal according to the comprehensive quality criterion. When applying the GRU model, the results are not unambiguous. A single-layer RNN with 55 neurons is appealing but not the best according to the comprehensive quality criterion. A higher value of the comprehensive criterion corresponds to a single-layer RNN with 75 neurons in the recurrent layer. However, the maximum value of the criterion corresponds to a three-layer GRU recurrent neural network with 60 neurons in the convolutional layer. It's essential to consider the increased training time for the network. Therefore, considering the minor difference in the values of the comprehensive quality criterion, a single-layer GRU RNN with 75 neurons in the recurrent layer is identified as more appealing. The next step involves comparing convolutional and recurrent neural networks with optimal sets of hyperparameters.

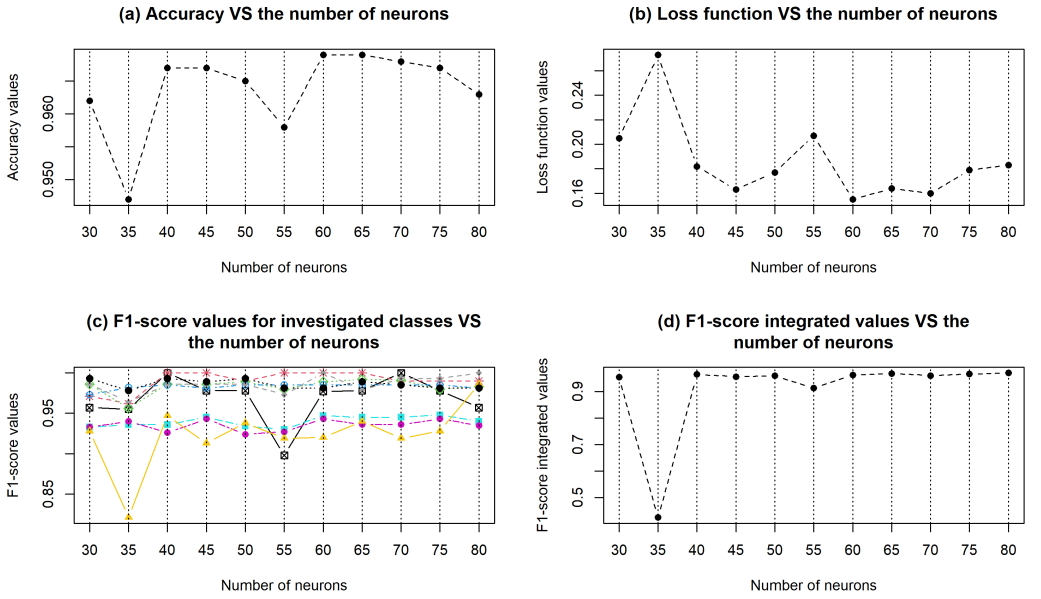

Figure 4.18: Results of the modeling when applying a three-layer GRU recurrent neural network

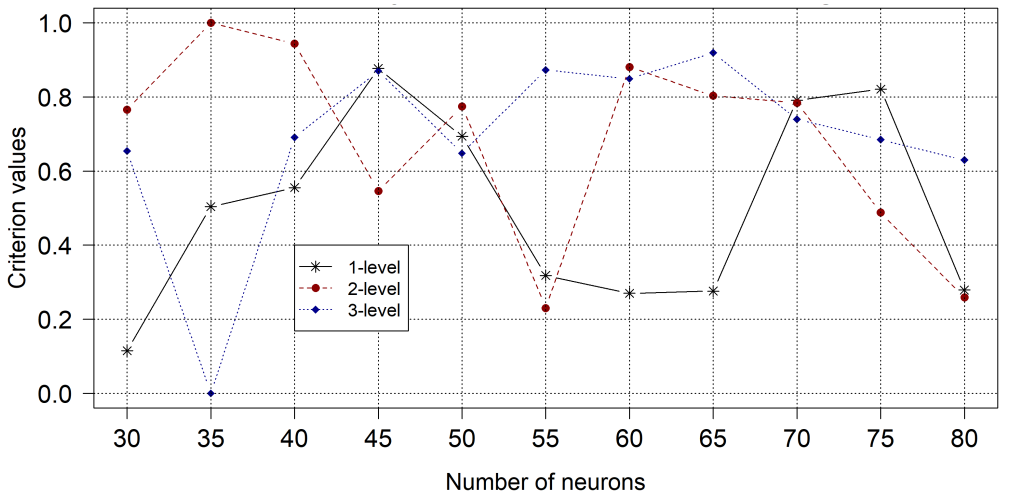

Figure 4.19: Distribution diagrams of the classification comprehensive quality criterion when using LSTM recurrent neural network

4.5 Comparative Analysis of CNN and RNN with Optimal Hyperparameter Values

The comparative analysis of the previously studied deep neural networks was performed by applying them to identical gene expression data. In this case, as in

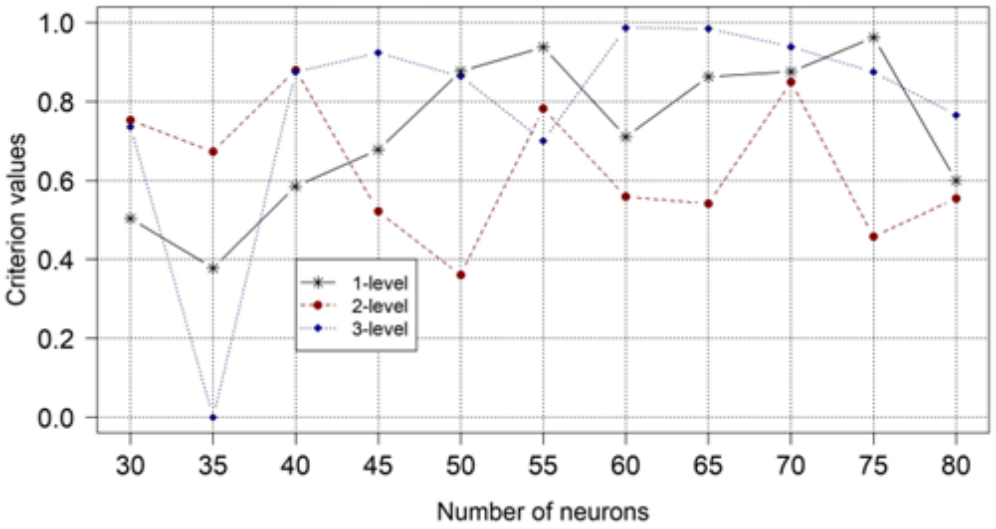

Figure 4.20: Distribution diagrams of the classification comprehensive quality criterion when using GRU recurrent neural network

previous ones, the data were divided into three subsets: for network training, its validation during the training process, and testing of the obtained model. The hyperparameter values of the convolutional neural network (CNN) were set considering our previous studies. In this instance, we applied the single-layer CNN, where: the number of filters = 32, kernel size = 3, Dense kernel = 64, maximal pooling = 3, activation functions for convolution layer, dense layer and output layer were *sigmoid*, *selu* and *softmax*, respectively.

In Figures 4.21 - 4.23, diagrams illustrating changes in the accuracy of sample classification and the loss function values during the training of the investigated neural networks are depicted. The analysis of the obtained diagrams indicates the absence of network overfitting in all cases since the character of changes in the respective criteria values when applying the training data subset and during model validation are consistent with each other. It should be noted that the training time for the convolutional neural network was 39s, which is significantly less than when using LSTM (185s) and GRU (166s) recurrent neural networks.

In Figure 4.24, the diagrams of classification quality criteria based on gene expression data are depicted when applying different types of deep neural networks. Analyzing the obtained results allows us to conclude that in terms of classification accuracy, calculated on the test data subset, the GRU neural network model is slightly better than the CNN and LSTM models. The classification accuracy when using the GRU network was 97.2%; in other cases, it was 97.1%. In the first case,

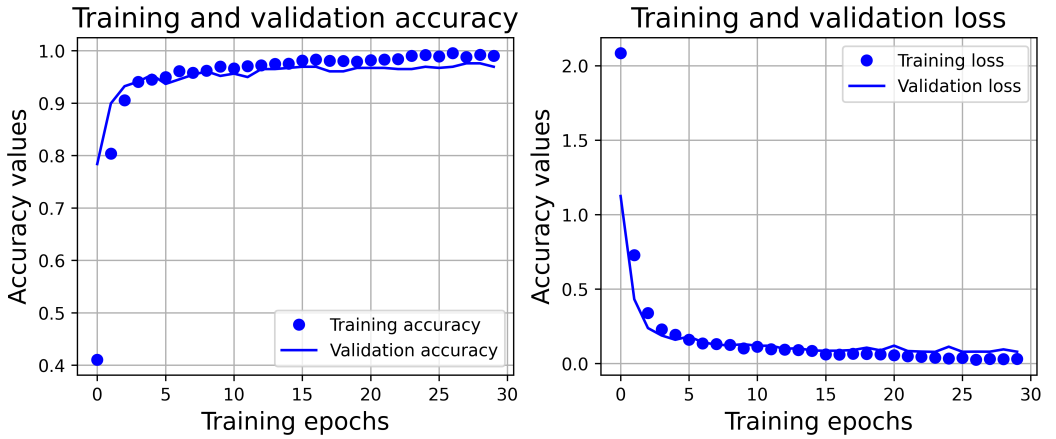

Figure 4.21: Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the CNN model

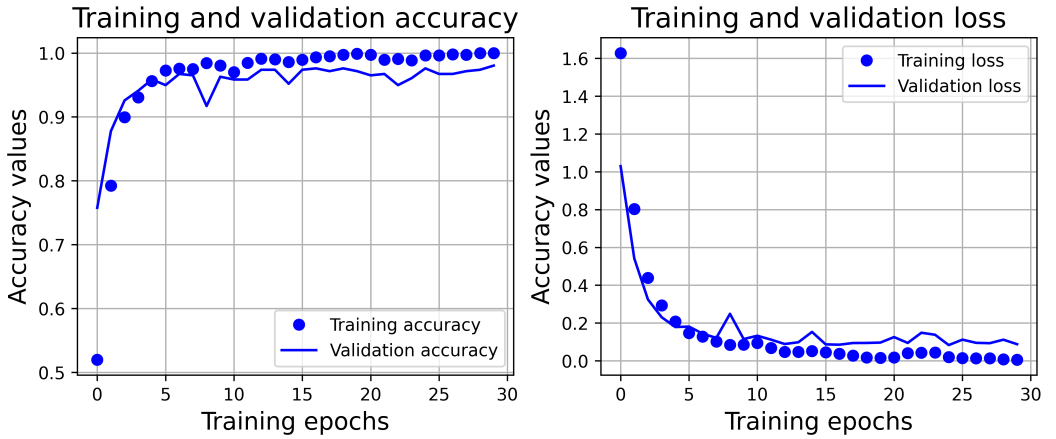

Figure 4.22: Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the LSTN RNN model

954 out of 981 objects were correctly identified. In other cases - 952. In terms of the loss function value and training time, the convolutional neural network is more appealing. The distribution pattern of F1-score values also indicates a small disparity in the sample identification results when distributed into respective classes. Analysis of the comprehensive quality criterion values confirms the conclusion regarding the greater appeal of the GRU recurrent neural network based on a set of criteria. This fact affirms the adequacy of the proposed method for evaluating the quality of the neural network according to a set of quality criteria, enhancing the objectivity of

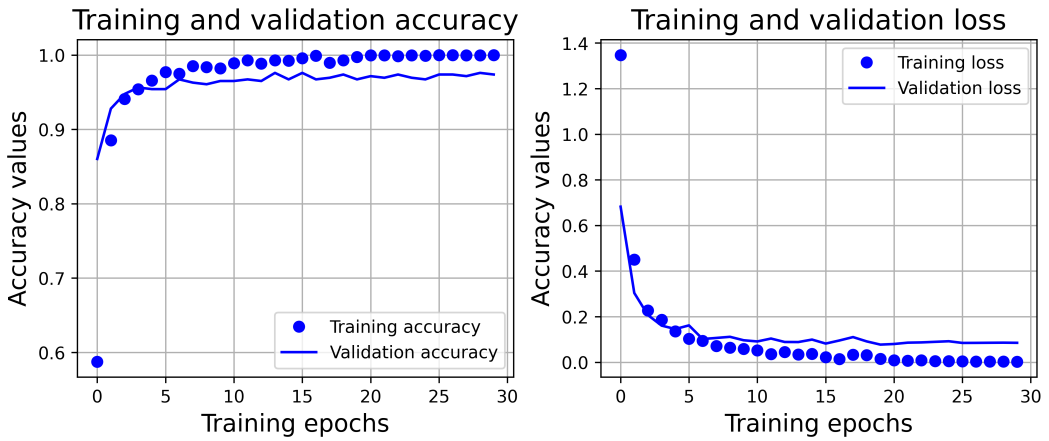

Figure 4.23: Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the GRU RNN model

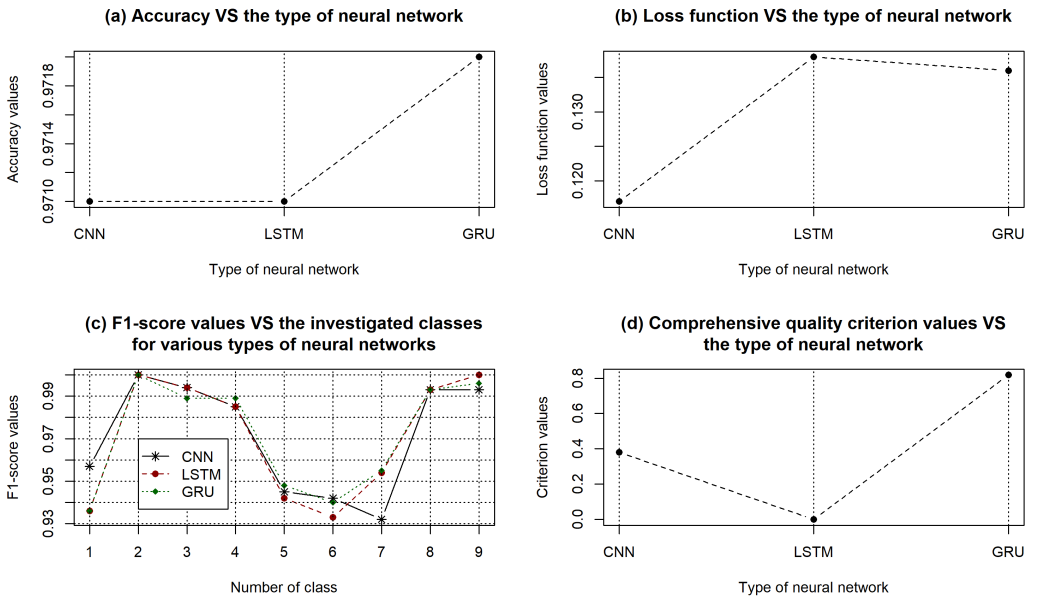

Figure 4.24: Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the GRU RNN model

forming the vector of optimal hyperparameters during the model tuning process.

A comparative analysis of simulation results revealed a preference for the GRU recurrent neural network over both the CNN and LSTM RNN in terms of classifica-

tion quality criteria for gene expression data processing. However, when evaluating based on loss function values, CNN-based models were superior. While CNNs are widely appreciated for their capacity to automatically learn spatial features from gene expression data, effectively discern biologically relevant patterns, and handle high-dimensional datasets without significant feature engineering, it's also known to consistently improve performance as they encounter more data, underscoring their value in genomics. Nevertheless, effectively utilizing CNNs requires identifying an optimal set of hyperparameters, including the architecture itself, a process that can be both time-consuming and resource-intensive.

In this context, the proposed methodology, based on recurrent neural networks, offers advantages in hyperparameter optimization compared to CNNs. Notably, the GRU RNN demonstrated superior performance in gene expression data classification. The introduced method for model effectiveness evaluation, which relies on a comprehensive quality criterion, allows for selecting the most suitable model by considering various classification quality criteria and assigning appropriate significance weights.

However, the limitation of this methodology lies in the approach to determining optimal hyperparameters. We employed a grid search algorithm in our research, which is notably time-intensive. As a future enhancement, we aim to leverage the Bayes optimization algorithm, streamlining the hyperparameters optimization process. This will also pave the way for a comparative analysis of different deep-learning model types for gene expression data processing.

4.6 Determining the Optimal Hyperparameter Values of DL Neural Networks Based on the Bayesian Optimization Algorithm

As demonstrated by the simulation results presented hereinbefore, forming a list of hyperparameters using the grid search method is quite labor-intensive and requires significant computational and time resources. The proposed ordered grid search method partially optimizes the search process, but this process is still not optimal. At the same time, the selection of hyperparameters significantly affects the model's performance, indicating the importance of this process. This subsection presents the simulation results regarding the application of the Bayesian optimization algorithm to automate the procedure of determining the optimal hyperparameters of a deep neural network.

The selection process is carried out according to the equation:

$$x_{opt} = argmin(f(x)), \quad x \in \theta \tag{4.12}$$

where: x is the combination of hyperparameters; $f(x)$ is the objective function that determines the outcome of the current combination of hyperparameters; θ is the range of hyperparameter values being optimized; x_{opt} is the vector of optimal hyperparameter values.

The main idea of Bayesian optimization lies in the intelligent search for the optimal solution, which takes into account previous results [55, 48]. This method uses principles of Bayesian statistics to evaluate the objective function that assesses the quality of the solution. When processing gene expression data, the objective function might be the classification accuracy on a validation subset of the data. The Bayesian optimization model comprises two key components:

- *Surrogate Model*: Typically a Gaussian process, this statistical model approximates the objective function. It accounts for non-linear dependencies and quantifies the uncertainty in predictions.
- *Acquisition Function*: This function guides the selection of new evaluation points within the surrogate model, striking a balance between exploring new areas ('exploration') and utilizing known effective points ('exploitation').

The optimization follows an iterative pattern: it evaluates new points, updates the surrogate model, and uses the acquisition function to pick the next point, continuing until a preset stopping criterion, like a maximum number of iterations, is met.

5-fold cross-validation is applied at each Bayesian optimization epoch to train the model correctly (without overfitting). This step involves dividing the data into five parts. The model is trained and evaluated five times, using a different fold as the validation set each time. This method enhances the accuracy and generalizability of the model by ensuring hyperparameters are fine-tuned based on varied subsets of data, thus improving the reliability of the optimization process.

Thus, Bayesian optimization is an iterative process that sequentially evaluates new points, updates the surrogate model, and uses the acquisition function to select the next point for evaluation. This process continues until a certain stopping criterion, such as the maximum number of iterations, is achieved.

Figure 4.25 depicts the flowchart of the stepwise procedure for processing gene expression data, based on the joint application of DL models and the Bayesian optimization algorithm, as implemented in the framework of our research. The implementation of this procedure involves the following steps:

1. Formation and pre-processing of gene expression data: the dataset should be organized as a data frame, with rows representing examined samples and columns representing genes.
2. Dataset splitting: The dataset is divided into training and testing subsets in a 0.7/0.3 ratio. The training subset is further split into training and validation

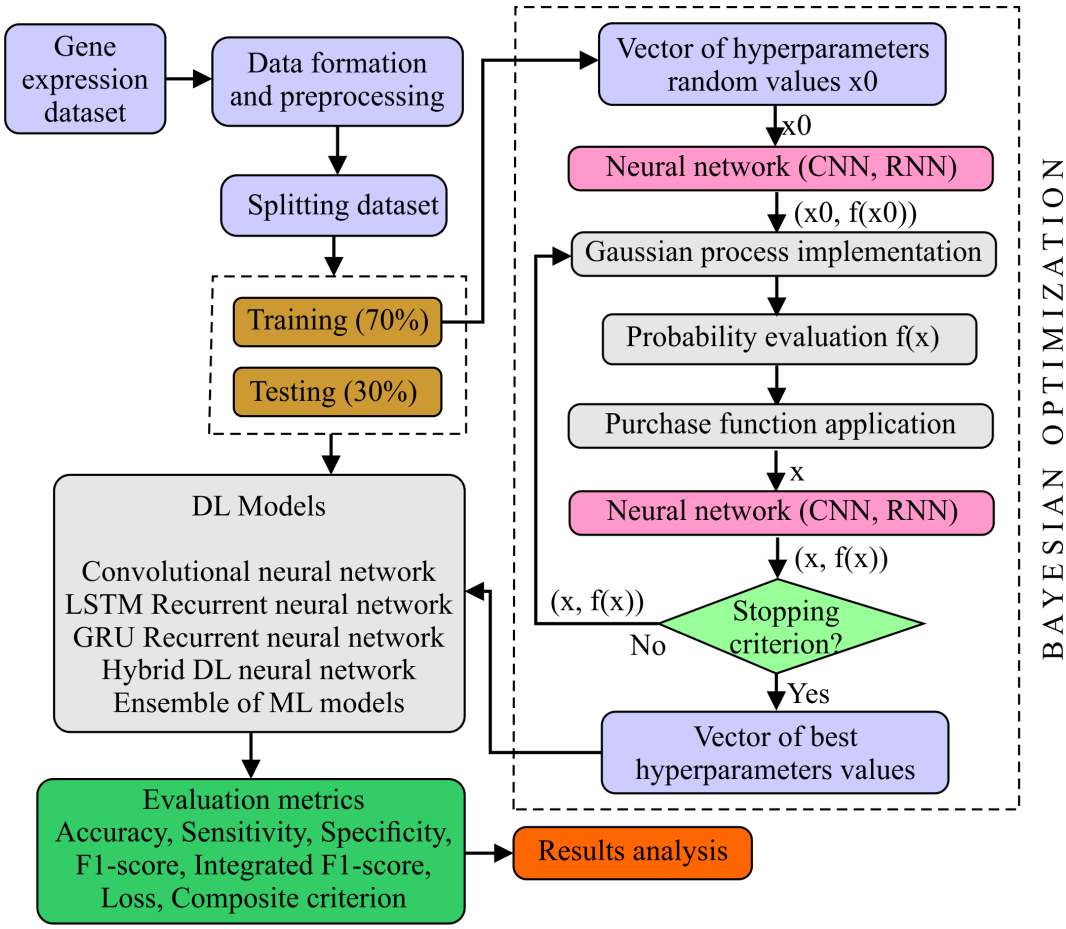

Figure 4.25: Flowchart of stepwise procedure for processing gene expression data, based on the joint application of DL models and the Bayesian optimization algorithm

subsets in a 0.8/0.2 ratio. These subsets are used to operate the Bayesian optimization algorithm and train the Deep Learning (DL) model with optimal hyperparameters.

3. Applying the Bayesian optimization method: This approach is utilized with each DL model to ascertain the vector of optimal hyperparameters. Integral to this process is implementing 5-fold cross-validation at each epoch of the Bayesian algorithm operation.

4. Evaluation of model quality: Forming the classification quality criteria to assess the model's performance.

5. Training and testing of DL models: This step includes calculating the classification quality criteria for each model.

6. Analysis of the obtained results.

4.6.1 DL-based Models

Within the framework of the current research, the following DL-based models were applied: 1D one-layer and two-layer CNNs, one-layer and two-layer LSTM RNNs, and GRU RNNs. To enhance the robustness and generalizability of the models, a 5-fold cross-validation technique was applied during the training process. This approach involved dividing the dataset into five equal parts, training the model in four parts, and validating it on the remaining part. This process was repeated five times, each part being used for validation once.

In applying CNNs, the simulation process involved optimizing several hyperparameters: the number of filters in the convolutional layers, the kernel size, maximal pooling, and the kernel size of the dense layer (dense kernel). Considering the results of previous studies, the activation functions applied were the *sigmoid* function (sigmoid) for convolutional layers, the *SELU* (Scaled Exponential Linear Unit) function for the dense layer, and the *softmax* function for the output layer of neurons. The range of values for the relevant hyperparameters was as follows: $num_filters = [8, 64]$, $kernel_size = [3, 10]$, $max_pooling = [2, 4]$, and $dense_kernel = [16, 256]$. The initial number of points in the hyperparameter feature space was set at 10, and the number of subsequent iterations to search for the optimal hyperparameter combination was 50 when applying a one-layer CNN and 70 for a two-layer CNN. The Dropout rate, representing the proportion of neurons being zeroed at each step during the network training process, was set at 20%.

In the case of RNN model utilized (LSTM and GRU), the number of neurons in layers varied within the range from 20 to 100. We also have investigated sequential and parallel hybrid models based on the integrated application of CNN and RNN. In each case, to determine the optimal hyperparameters and control overfitting, we also applied the Bayesian optimization algorithm and k-fold cross-validation method. Figure 4.26 depicts the block diagram of a hybrid classification model for one-dimensional gene expression data based on the sequential application of two-layer convolutional and recurrent neural networks, where the recurrent network can be implemented using either the LSTM or GRU algorithm. The value of the hyperparameters can be changed during the simulation procedure implementation.

The application of a CNN at the initial stage of model implementation is justified by its ability to detect complex dependencies between genes in the respective gene expression profile. CNNs can identify local dependencies in the gene expression profile, such as local structures, motifs, or patterns indicative of specific functions

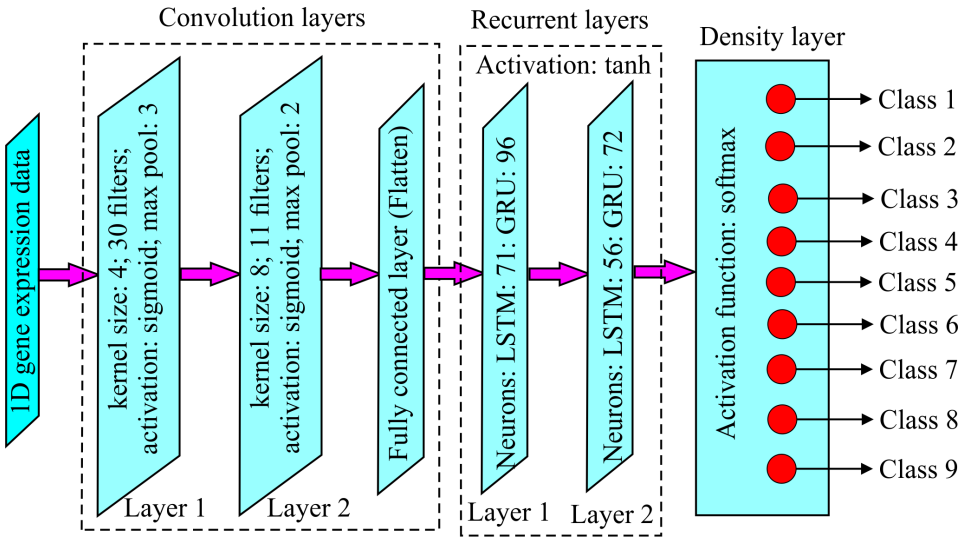

Figure 4.26: The block diagram of the hybrid model for classifying one-dimensional gene expression data, based on the sequential application of two-layer convolutional and recurrent neural networks

or pathological processes. Convolutional layers can provide translation invariance of the data. This means that CNNs can recognize the same dependencies in various positions of the gene expression data, regardless of their exact location. Furthermore, CNNs can automatically select useful features from gene expression data during training, which aids in enhancing the quality of forming a fully connected layer for its subsequent use as input data for the recurrent layer. Applying a pooling layer (maximal pooling) at the output of each convolutional layer helps reduce the dimensionality of the data while preserving important features.

Recurrent layers at the model's output allow it to consider the sequence of data, that is, the order of genes in vectors, which can significantly impact the results of gene expression data classification. Applying recurrent layers at the output of the convolutional layer also allows for reducing the number of model parameters compared to using recurrent layers on a fully sequential input, which can decrease the risk of model overfitting. The absence of overfitting was monitored in all cases through the convergence of the model classification accuracy character changes and the loss function value, calculated on the training and validation data during the model training process.

The second hybrid model explored in our study employs a parallel approach using various top-performing DL models for classifying gene expression data. This model makes intermediate decisions which are then aggregated to form the final decision. A key step in this process involves applying a classifier to these intermediate decisions.

In this context, we utilized the CART (Classification and Regression Trees) machine learning method. CART is an algorithm for building decision trees, chosen for its ability to recursively split a dataset into subgroups. This splitting is based on the values of a specific feature (the most significant intermediate decision), resulting in the construction of a decision tree. Each leaf of this tree corresponds to a distinct class, categorizing the objects within that subset of data. A notable advantage of the CART algorithm is its interpretability; the sections and conditions of the decision tree can be easily understood and explained.

The block diagram of the hybrid model for classifying gene expression data based on an ensemble of machine-learning methods is illustrated in Figure 4.27.

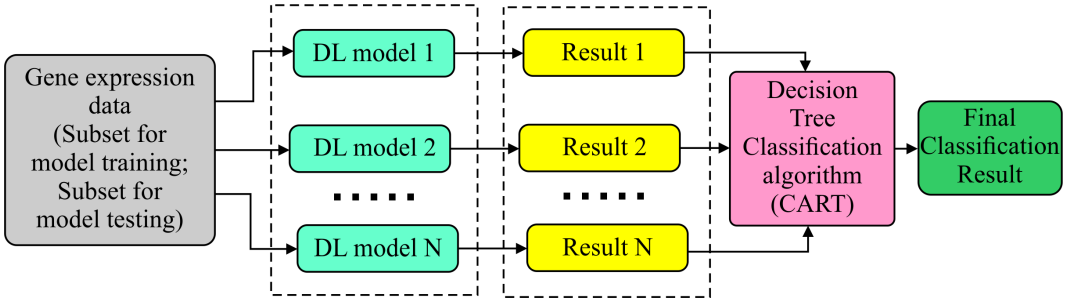

Figure 4.27: The block diagram of the hybrid model for classifying gene expression data, which is based on an ensemble of DL and ML methods.

To obtain objective results, the modeling process was carried out in four stages. In the first stage, four models of deep neural networks were applied: a two-layer recurrent LSTM network; a two-layer recurrent GRU network; a hybrid CNN-LSTM network; and a hybrid CNN-GRU network. The second and third stages involved the application of three hybrid models. In the second stage, the two-layer LSTM model was removed, and in the third – the hybrid CNN-LSTM model. In the fourth stage, two of the best models based on the application of the GRU recurrent network from previous studies were used.

4.6.2 Simulation, Results and Discussion

Figure 4.28 presents charts depicting the Accuracy and Loss metrics for both the training and validation datasets across epochs, specifically during the training of a one-level CNN model. Similar charts were generated for other models. Analysis of these charts reveals no signs of overfitting; this is evidenced by the consistent changes in accuracy and loss values for both the training and validation datasets throughout the training and validation phases of the model.

Table 4.3 and 4.4 displays the modeling results regarding applying the Bayesian

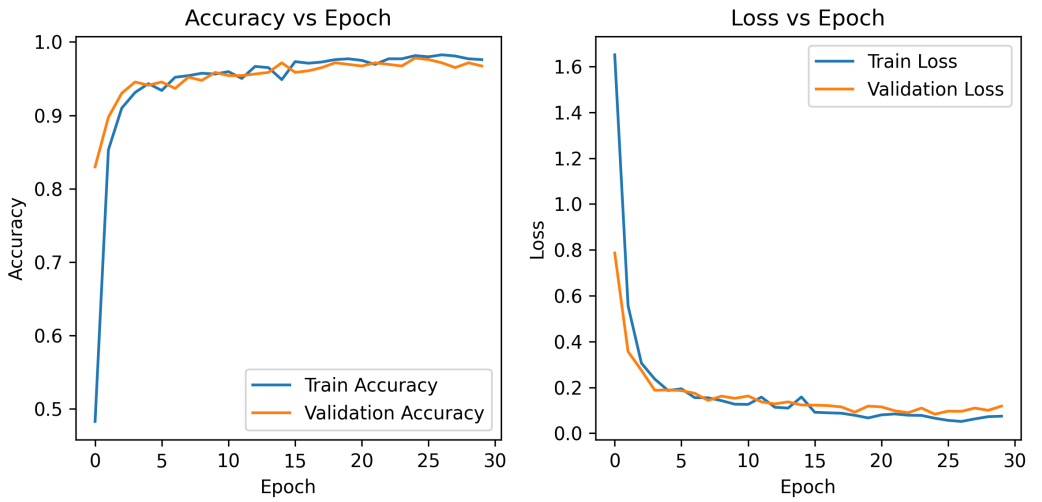

Figure 4.28: Charts depicting the Accuracy and Loss metrics for both the training and validation datasets across epochs, specifically during the training of a one-level CNN model

optimization algorithm for one-layer and two-layer CNNs, LSTM and GRU RNNs to determine the optimal combination of hyperparameters.

Table 4.3: Modeling results regarding the application of the Bayesian optimization algorithm for determining the optimal combination of hyperparameters for one-layer and two-layer CNNs

One-layer CNN				
Accuracy	Number of filters	Kernel size	Max pooling	Dence kernel
0.972	44	5	3	48
Two-layer CNN				
Accuracy	Number of filters 1	Kernel size 1	Max pooling 1	Dence kernel
0.966	53	8	4	38
	Number of filters 2	Kernel size 2	Max pooling 2	
	27	14	3	

Tables 4.5 and 4.6 show the classification results of test subset data samples (981) using one-layer (Table 4.5) and two-layer (Table 4.6) CNNs, the optimal hyperparameters of which were determined using the Bayesian optimization algorithm.

In Tables 4.7, 4.8 and Tables 4.9, 4.10, the modeling results are presented regarding the application of LSTM and GRU recurrent neural networks with an optimal number of neurons in the recurrent layers, respectively.

Table 4.4: Modeling results regarding the application of the Bayesian optimization algorithm for determining the optimal combination of hyperparameters for one-layer and two-layer RNNs

One-layer LSTM RNN	Two-layer LSTM RNN		One-layer GRU RNN	Two-layer GRU RNN	
Number of neurons	Number of neurons 1	Number of neurons 2	Number of neurons	Number of neurons 1	Number of neurons 2
43	75	42	74	84	67

Table 4.5: Modeling results regarding the application of a one-layer CNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.880	1.000	0.936	0.819	0.973	0.127	0.738
GBM	0.981	1.000	0.990				
KIRC	0.994	0.994	0.994				
LGG	0.978	0.985	0.981				
LUAD	0.933	0.982	0.957				
LUSC	0.986	0.922	0.953				
SARC	0.983	0.881	0.929				
STAD	1.000	1.000	1.000				
NORM	0.979	1.000	0.989				

Table 4.6: Modeling results regarding the application of a two-layer CNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.815	1.000	0.898	0.351	0.966	0.122	0.176
GBM	1.000	0.980	0.990				
KIRC	0.994	0.994	0.994				
LGG	0.971	1.000	0.985				
LUAD	0.951	0.917	0.934				
LUSC	0.935	0.941	0.938				
SARC	0.983	0.881	0.929				
STAD	0.986	1.000	0.993				
NORM	0.979	1.000	0.989				

The analysis of the obtained results allows us to conclude that in all cases, the classification accuracy of the samples is quite high and varies within the range from 96.5% when using a single-layer CNN to 97.5% when using a two-layer GRU recurrent neural network. Moreover, the two-layer GRU recurrent neural network demonstrated the highest effectiveness when applying gene expression data, both as per the accuracy criterion for classification of samples across all classes (Accuracy) and for individual classes (Sensitivity, Specificity, F-measure). The analysis of the

Table 4.7: Modeling results regarding the application of a one-layer LSTM-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.880	1.000	0.936				
GBM	0.981	1.000	0.990				
KIRC	0.994	0.983	0.989				
LGG	0.992	0.963	0.977				
LUAD	0.952	0.929	0.940	0.666	0.966	0.141	0.305
LUSC	0.947	0.941	0.944				
SARC	0.867	0.970	0.915				
STAD	0.986	1.000	0.993				
NORM	1.000	0.985	0.993				

Table 4.8: Modeling results regarding the application of a two-layer LSTM-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.917	1.000	0.957				
GBM	1.000	1.000	1.000				
KIRC	0.989	0.983	0.986				
LGG	0.978	0.993	0.985				
LUAD	0.926	0.959	0.942	0.872	0.969	0.139	0.574
LUSC	0.966	0.915	0.940				
SARC	0.955	0.940	0.947				
STAD	0.973	0.985	0.986				
NORM	1.000	0.978	0.993				

Table 4.9: Modeling results regarding the application of a one-layer GRU-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.880	1.000	0.936				
GBM	1.000	1.000	1.000				
KIRC	0.994	0.983	0.989				
LGG	0.978	0.993	0.985				
LUAD	0.942	0.959	0.950	0.847	0.971	0.127	0.693
LUSC	0.953	0.928	0.940				
SARC	0.955	0.940	0.947				
STAD	0.986	1.000	0.993				
NORM	1.000	0.985	0.993				

character of the distribution of values of the composite criterion confirms this conclusion. It should be noted that the values of sensitivity, specificity, and F-score

Table 4.10: Modeling results regarding the application of a two-layer GRU-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	1.000	1.000	1.000				
GBM	1.000	1.000	1.000				
KIRC	0.989	0.989	0.989				
LGG	0.985	0.985	0.985				
LUAD	0.942	0.959	0.950	0.876	0.978	0.152	0.800
LUSC	0.966	0.928	0.947				
SARC	0.926	0.940	0.933				
STAD	0.986	1.000	0.993				
NORM	1.000	1.000	1.000				

criteria vary when applying different types and structures of neural networks, which may indicate the need to enhance the objectivity of obtaining results regarding the identification of the class to which the samples belong by hybridizing different models of gene expression data classification.

The classification results of the gene expression data test subset, obtained by applying hybrid CNN-LSTM and CNN-GRU models, are presented in Tables 4.11 and 4.12, respectively.

Table 4.11: Modeling results regarding the application of a hybrid model CNN-LSTM-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.957	1.000	0.978				
GBM	1.000	1.000	0.981				
KIRC	0.994	0.989	0.991				
LGG	0.978	0.993	0.985				
LUAD	0.902	0.976	0.938	0.806	0.967	0.126	0.533
LUSC	0.979	0.895	0.935				
SARC	0.953	0.910	0.931				
STAD	0.973	0.985	0.989				
NORM	0.993	0.971	0.985				

An analysis of the modeling results allows us to conclude that the classification outcomes for the samples are high, both in terms of accuracy across all classes and in terms of the adequacy of sample distribution into separate classes in both cases. In Figure 4.29, the results of a comparative analysis of all types of deep learning neural networks and their combinations used during the simulation process are illustrated. Analyzing the obtained results allows concluding that, by all criteria, models based on applying a two-layer GRU recurrent network are more effective compared

Table 4.12: Modeling results regarding the application of a hybrid model CNN-GRU-RNN for the classification of various types of cancer diseases

Class	PR	RC	F1	F1-int	ACC	LOSS	Comp QC
ACC	0.846	1.000	0.917				
GBM	1.000	1.000	1.000				
KIRC	0.994	0.983	0.989				
LGG	0.978	0.993	0.985				
LUAD	0.941	0.947	0.944	0.675	0.969	0.118	0.547
LUSC	0.947	0.935	0.941				
SARC	0.953	0.910	0.931				
STAD	0.986	1.000	0.993				
NORM	0.993	0.985	0.989				

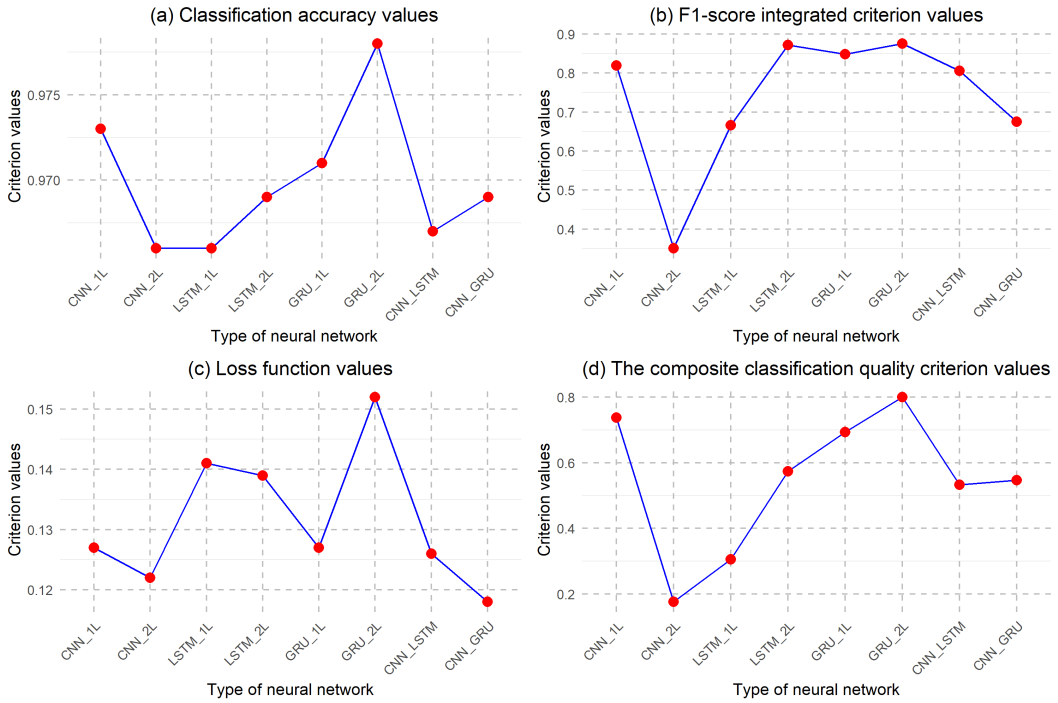

Figure 4.29: Results of the comparative analysis of different types of deep learning neural networks: a) classification accuracy; b) F-score compjsite criterion; c) loss function values; d) compjsite quality criterion for data classification

to other models explored. Thus, when applying a two-layer GRU-RNN, the classification accuracy of test data subset samples is 97.5%, slightly higher than the classification accuracy when applying the CNN-GRU-RNN hybrid model (97.1%).

The composite criterion value, in this case, is slightly higher when using the hybrid model, which is explained by the smaller value of the loss function. By the composite F1 score criterion, both models are appealing, indicating high quality in identifying the investigated samples. It should be noted that the worst classification accuracy corresponds to samples related to the first class. This fact can be explained by the small number of samples (a total of 79 samples and 22 for the test subset), complicating the procedure for quality network training. Increasing the number of samples significantly reduces the spread of F1 score values.

The next phase in the simulation process involves utilizing an ensemble of machine learning methods. As noted in section III(C), this step involves the parallelizing data processing and implementing a consensus decision-making approach based on the interim outcomes from the previous stage. Such a strategy is expected to significantly improve objectivity in finalizing decisions about the object's state. The simulation results on applying the hybrid model based on the ensemble of DL and ML methods are shown in Figure 4.30.

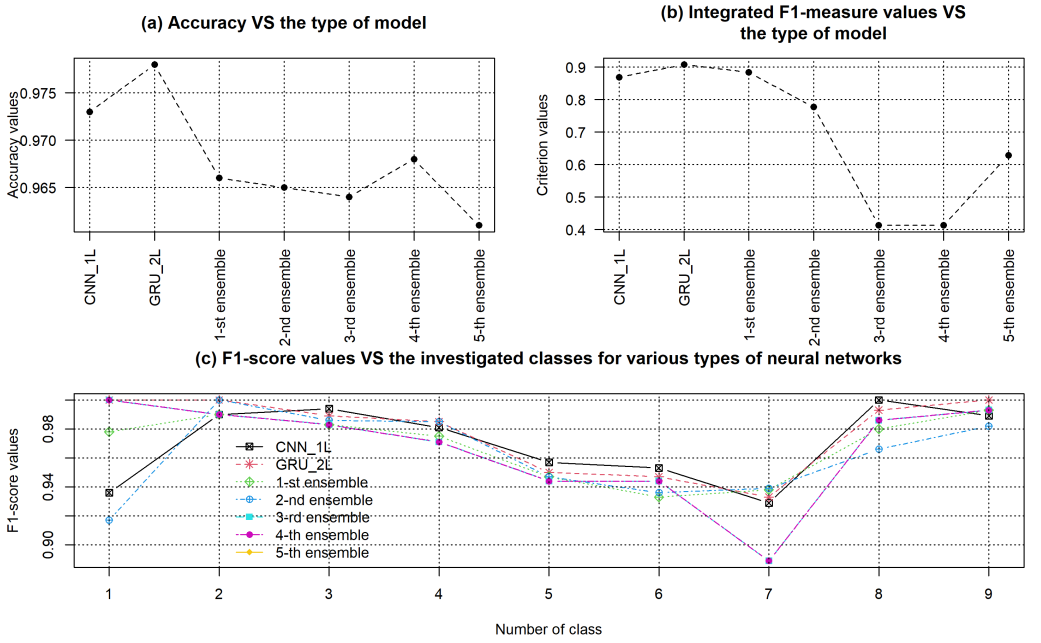

Figure 4.30: Modeling results regarding the comparative analysis of machine learning methods ensembles

From the analysis of the results, it becomes clear that using a DL-based models ensemble for classifying a single gene expression dataset does not necessarily offer an advantage in terms of classification accuracy. The quality of sample identification is

diminished when compared to the use of optimally tuned two-layer CNN and GRU RNN models.

However, it's noteworthy that the first ensemble of DL and ML models shows high accuracy in categorizing objects into individual classes (F1-score integrated value). Furthermore, when compared to similar multiclass problem-solving using different DL models for cancer identification as presented in Table 4.13, the classification accuracy is higher in all instances when using the investigated DL models. This underscores the significance of selecting optimal model hyperparameters tailored to the specific data being analyzed.

Table 4.13: Comparison of various models for multiclass problem-solving using different DL models for cancer identification

Reference	Number of cancer types	Methodology	Accuracy,%
Gupta at al. (2022) [44]	5	DL with CNN	92
Karthika at al. (2023) [54]	2	DL with CNN	94.56
Mostavi at al. (2020) [68]	33	DL with CNN	93.9 - 95.0
Chuang at al. (2021) [36]	11	DL with CNN	95.4 - 97.4
Ramirez at al. (2020) [71]	33	DL with GCNN	89.9 - 94.7
Srikantamurthy at al. (2023) [74]	8	DL with CNN-LSTM	92.5

Considering the research outlined in [4, 51, 65], we can highlight the key performances of our proposed technique. In these prior studies, the authors developed effective methods for selecting informative attributes (genes) and applied both machine learning and deep learning techniques to identify various types of cancer. While these studies yielded interesting results, their focus was primarily on feature selection followed by the application of suitable classifiers for sample identification, utilizing 10-fold cross-validation during model training.

In contrast, the presented research explored a range of deep learning models, including hybrid models, for the classification of various cancer types based on a comprehensive set of genes (19,947). The main objective of our study was to optimize the hyperparameters of the models. This was achieved through the combined use of the Bayesian optimization algorithm and k-fold cross-validation in each epoch of the algorithm's application. Additionally, we enhanced the classification quality criteria by introducing an integrated quality criterion, allowing for a more meticulous evaluation of the classification results. This approach represents the principal distinction between our methodology and the existing ones, offering a more comprehensive and refined analysis in the field of cancer classification.

A minor decrease in sample classification accuracy with ensemble-based DL models could be offset by the increased objectivity in making final decisions about the object's state. In multiclass problems addressed by models ensemble. Models can show for individual samples different identification results. This can lead to a slight

drop in classification accuracy, as observed in our results. Nevertheless, higher objectivity is attained through the consistent identification of sample states across various methods. Improving the accuracy of the samples identification, in this instance, could be achieved by a more detailed pre-processing of gene expression data, employing gene ontology analysis, cluster, and bicluster analyses. Exploring these methods further will be the focus of our subsequent research.

Chapter 5

Application of Developed Models, Methods, and Algorithms in the Disease Diagnosis System Based on Gene Expression Data

5.1 Introduction

This chapter presents the results of the practical implementation of the proposed models, methods, and algorithms in disease diagnosis systems based on gene expression data. The framework of the information technology, along with the flow chart and detailed stepwise procedure, is presented in the Introduction of this thesis.

The first subsection provides a detailed description of the experimental data used during the simulation process implementation. Gene expression data from objects studied for Alzheimer's disease were used. This data was obtained through DNA microarray experiments and included samples from objects in which the disease was identified based on clinical research results, as well as samples in which the disease was not detected. The second group of experimental data consists of gene expression data studied for various types of cancer. This data also included samples from both diseased objects and objects in which cancer was not identified based on clinical research results. The second group of samples was obtained using the RNA sequencing method.

The subsequent subsections present the results of the step-by-step implementation of the proposed information technology stages: from data preprocessing using

the functions and modules of the "Bioconductor" package in the R programming language, forming subsets of significant and mutually expressed gene expression profiles using methods based on gene ontology analysis, cluster and bicluster analyses, to diagnosing the condition of the studied objects by applying a classifier based on deep learning methods to the gene expression data subsets formed in the previous stages, culminating in the final decision regarding the state of the investigated objects.

5.2 Experimental Gene Expression Data Used in the Modeling Process

The modeling process utilized gene expression data from objects studied for Alzheimer's disease and various types of cancer. The data were obtained using different technologies, allowing for the assessment of the proposed information technology's effectiveness across various data types. Gene expression data for objects studied for Alzheimer's disease were obtained through DNA microarray experiments, while gene expression data for various types of cancer were obtained through RNA molecule sequencing experiments. All types of data included gene expression information from objects in which the respective disease was identified based on clinical research results, as well as from objects in which the disease was not detected.

5.2.1 Experimental Gene Expression Data Studied for Alzheimer's Disease

The first type of experimental data contained gene expression analysis results in the human brain, performed to understand the molecular mechanisms of Alzheimer's disease (AD) and age-related neurological disorders. The data, GSE5281, is freely available on the Gene Expression Omnibus (GEO) website [2] and includes 161 samples taken from three Alzheimer's Disease Brain Centers [59, 61, 72, 60]. Gene expression profiling was performed using the Affymetrix U133 Plus 2.0 array, with each array containing 54,674 transcripts. The formation of the gene expression array was carried out by implementing background correction using the 'rma' method, normalization using the 'quantiles' method, PM correction using the 'mas' method, and summarization using the 'avgdiff' method. Data annotation analysis showed that Alzheimer's disease was not detected in 74 objects. In this case, the data can be divided into two classes: 74 objects correspond to the first class of objects in which the disease was not detected, and 87 objects correspond to the class of objects in which Alzheimer's disease was identified. After removing genes whose identifiers did not match gene ontology identifiers, the total number of genes was reduced to 44,662. Thus, initially, the gene expression data of the objects studied for Alzheimer's disease had the form: (161×44662). Figure 5.1 shows the distribution

pattern of gene expression values of the samples analysed. As can be seen from

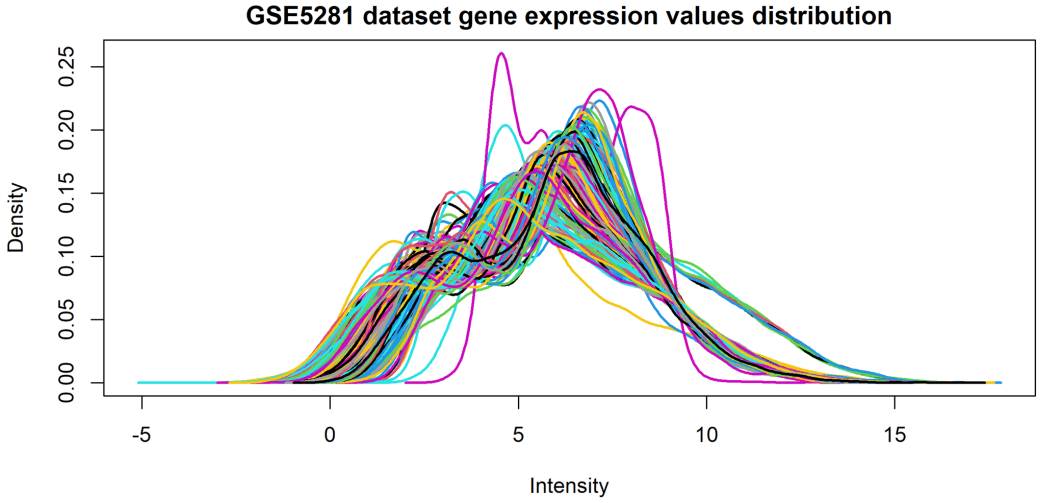

Figure 5.1: Distribution pattern of normalized gene expression values of samples studied for Alzheimer's disease

Figure 5.1, the gene expression values of all samples fall within a fairly narrow range, but there is a certain group of low-expressed genes. This fact confirms the need for further data processing to remove insignificant genes.

5.2.2 Experimental Gene Expression Data Studied for Cancer Disease

The second type of data consists of gene expression data obtained through RNA-Sequencing (RNA-Seq) method. These data are publicly available on the website of the TCGA project (The Cancer Genome Atlas) [3]. During the simulation process, samples corresponding to 13 types of cancer diseases were investigated. A separate group was composed of samples for which no cancer disease was detected. The data classification is presented in Table 5.1.

Initially, the data contained 60,660 genes. After removing genes not expressed for any of the samples, genes with zero variance in expression profiles (the expression values for all samples are identical), and genes whose identifiers do not match the identifiers of the studied organism's genes according to gene ontology, 25,566 genes remained. At this stage, the gene expression data were defined by the number of genes of the relevant type that determine the state of the organism being studied. Table 5.2 illustrates the distribution of maximum gene counts in the respective profiles for all studied samples (6,344).

Table 5.1: Classification of data from patients investigated for various types of cancer diseases

No	Type of cancer disease	Number of samples
1	BLCA – Bladder Cancer	412
2	BRCA – Breast Cancer	1118
3	CESC – Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma	306
4	COAD – Colon Adenocarcinoma	483
5	ESCA – Esophageal Carcinoma	185
6	GBM – Glioblastoma Multiforme	170
7	HNSC – Head and Neck Squamous Cell Carcinoma	522
8	KIRC – Kidney Renal Clear Cell Carcinoma	542
9	LAML – Acute Myeloid Leukemia	151
10	LGG – Lower Grade Glioma	534
11	LIHC – Liver Hepatocellular Carcinoma	374
12	LUSC – Lung Squamous Cell Carcinoma	502
13	LUAD – Lung Adenocarcinoma	541
14	Cancer is not identified (Normal)	504

Table 5.2: The character of the distribution of gene counts absolute maximum values in respective profiles for all studied samples

Min	1st Qu (25%)	Median	Mean	3rd Qu (75%)	Max
1	861	9525	49389	30920	12581910

As the analysis of the distribution of maximum gene count values in respective profiles shows, the absolute values are inconvenient for further data processing. The data normalization was carried out in several stages. In the first step, the absolute gene expression values were transformed into a more convenient range, CPM – Count Per Million, according to the formula:

$$CPM_{ij} = \frac{count_{ij}}{\sum_{j=1}^{m} count_{ij}} \cdot 10^6 \tag{5.1}$$

where: i is the identifier of the sample being studied; j is the number of gene identifiers corresponding to the i-th sample; count_{ij} is the count of j-type genes in the i-th sample; the multiplier 10^6 acts as a normalizing factor to some extent.

To reduce the impact of very high gene count values on further analysis, a logarithmic transformation was applied to the CPM values in the subsequent step. The results of the distribution of the normalized data are presented in Table 5.3.

At the next step, genes that were low-expressed across all samples (maximum expression value of the respective profile < 0) were removed. At the final step, negative gene expression values were replaced with zeros, corresponding to genes not expressed in individual samples. The number of genes at this stage was reduced

Table 5.3: The character of the distribution of normalized gene expression values in respective profiles for all studied samples

Min	1st Qu (25%)	Median	Mean	3rd Qu (75%)	Max
-4.351	4.329	7.443	6.595	9.104	17.714

to 23,655. Thus, initially, the gene expression data of patients studied for various types of cancer appeared as: (6344 × 23955). Figure 5.2 shows the distribution pattern of gene expression values for the corresponding samples. As can be seen, the

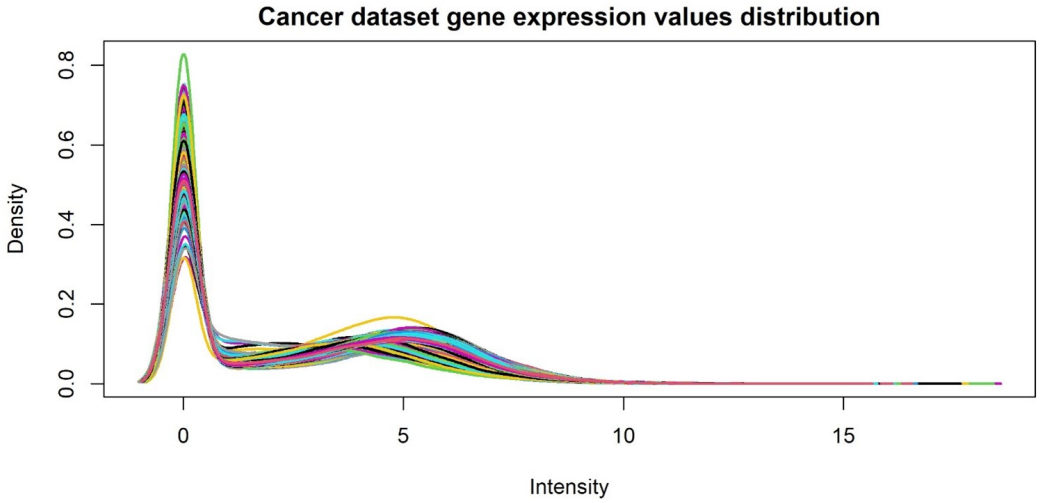

Figure 5.2: Distribution pattern of normalized gene expression values of samples studied for cancer disease

distribution pattern of gene expression values obtained using the RNA sequencing method differs from the distribution pattern of gene expression values obtained using DNA microarray experiments (Figure 5.1. However, in all cases, the gene expression values for all samples change consistently, indicating the correctness of the gene expression data normalization procedure. The next step is to apply gene ontology analysis to the normalized gene expression data to form subsets of significant gene expression data.

5.3 Application of Gene Ontology Analysis for the Formation of Subsets of Significant Genes

At this stage of the research, an analysis of gene ontology was applied to the formed gene expression data using the ANOVA test (analysis of variance between and within

groups), functions of the TopGO package, and the comprehensive application of Fisher's and Kolmogorov-Smirnov tests for the final formation of a list of significant gene identifiers. The implementation of this procedure involves the following steps:

1. Data preparation and formation of a list of gene identifiers.
2. Installation and download of necessary packages. Within the current research, the "TopGO" [5] and "org.Hs.eg.db" [35] packages from the "bioconductor" module [1] of the R programming environment were used.
3. Performing the ANOVA test to identify genes that show significant changes in expression levels between different groups (classes).
4. Application of the Benjamini-Hochberg (BH) method to correct the p-values obtained in the previous step, which helps to minimize the type I error when forming the list of significant gene identifiers.
5. Application of the TopGO package functions (Tool for the Ontological Analysis and Visualization of Differential Gene Expression Data) for gene ontology analysis, employing the BP ontology – biological processes, and the list of gene identifiers from the "org.Hs.eg.db" package, corresponding to the human genome (Homo Sapiens). The probability threshold (p-value) for distinguishing significant from non-significant genes was set empirically, taking into account the type of data being studied.
6. Application of Fisher's test to assess the statistical significance of the ratio of genes in certain biological processes. This step allows estimating the likelihood that certain biological processes are overrepresented in the selected list of genes.
7. Application of the Kolmogorov-Smirnov test to compare the distributions of gene expression levels in different groups, enabling the identification of distribution differences (probabilities) that may be important for further identification of the object's condition.
8. Analysis of results and formation of subsets of significant genes based on the outcomes obtained from the application of the two tests.

5.3.1 Application of Gene Ontology Analysis to Samples Studied for Alzheimer's Disease

Table 5.4 presents the results of applying gene ontology analysis to the gene expression data of patients studied for Alzheimer's disease.

Table 5.4: The result of applying gene ontology analysis to the gene expression data of patients investigated for Alzheimer's disease

GO:ID	Term	Annot.	Sign.	Fisher's test	KS test
GO:0009060	aerobic respiration	185	112	1.4e-23	<1e-30
GO:0006119	oxidative phosphorylation	137	90	9.4e-23	<1e-30
GO:0042773	ATP synthesis coupled electron transport	93	69	1.8e-22	<1e-30
GO:0042775	mitochondrial ATP synthesis coupled elec...	93	69	1.8e-22	<1e-30
GO:0007005	mitochondrion organization	539	239	2.3e-21	<1e-30
GO:0045333	cellular respiration	231	127	2.6e-21	<1e-30
GO:0019646	aerobic electron transport chain	85	63	1.4e-20	<1e-30
GO:0009144	purine nucleoside triphosphate metabolic...	159	94	5.4e-19	<1e-30
GO:0022904	respiratory electron transport chain	115	75	6.5e-19	<1e-30
GO:0009142	nucleoside triphosphate biosynthetic pro...	119	76	2.4e-18	<1e-30

Initially, the data contained 44,662 genes. After analyzing gene identifiers for correspondence with the identifiers of all genes in the "org.Hs.eg.db" database, the number of relevant genes was reduced to 21,367. The adjusted p-value threshold was set at 0.01. In this case, with a 99% probability, 4,841 ontologies were identified as significant. The table 5.4 shows the 10 most significant ontologies considering the adjusted p-values, terms corresponding to each ontology, and the number of genes. In Figure 5.3, the distribution character of the identified ontologies at different p-values obtained by applying the Fisher's test and the Kolmogorov-Smirnov test is depicted. Ontologies corresponding to a gene count greater than 10 are marked in red, while those with a gene count less than 10 are marked in blue. As can be seen from the Figure 5.3, the significance of ontologies when applying different tests can contradict each other. Thus, ontologies significant according to Fisher's test may be insignificant according to the Kolmogorov-Smirnov test and vice versa. Therefore, in forming the list of significant gene identifiers, the results of both tests were used. An ontology was considered significant if it was significant according to both the Fisher's test and the Kolmogorov-Smirnov test.

Figure 5.4 shows the interaction graph of the ten most significant ontologies with each other and with other ontologies that are in some way interrelated with each other and with the most significant ontologies. In this case, the results of Fisher's test were used. A similar graph was obtained using the Kolmogorov-Smirnov test. Analysis of the obtained graph confirms the complex nature of the interaction between ontologies and their corresponding genes. It should be noted that the interaction pattern of the network nodes changes significantly when the type of test

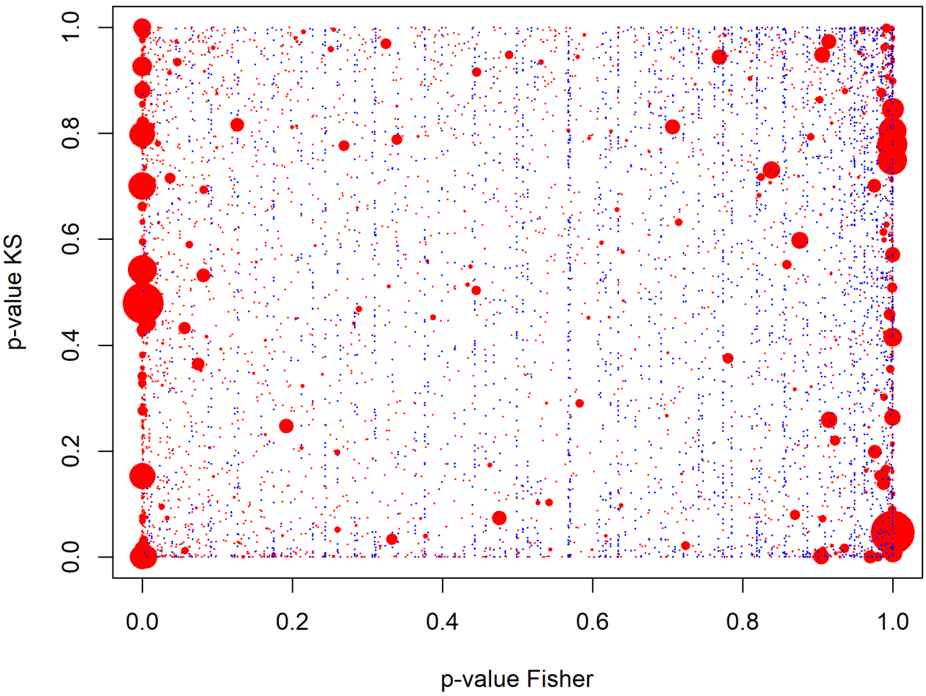

Figure 5.3: A bubble chart of the distribution of identified ontologies at different p-values obtained from applying Fisher's and Kolmogorov-Smirnov tests for the gene expression data of patients studied for Alzheimer's diseases

is changed, which is consistent with the results shown in Figure 5.3. At this stage, 14,601 significant genes were identified, and the gene expression data of patients studied for Alzheimer's disease reached a dimension of $(161 \times 14,601)$.

5.3.2 Application of Gene Ontology Analysis to Samples Studied for Cancer Disease

Unlike the gene expression data investigated for Alzheimer's disease, the gene expression data of patients examined for various types of cancer were obtained using the RNA sequencing method, which is significantly more accurate in assessing gene activity levels compared to the method based on DNA microarrays. The results of applying gene ontology analysis for the ten most significant ontologies using gene expression data of patients studied for various types of cancer are presented in Table 5.5.

The analysis of the data in Table 5.5 confirms the assumption regarding the higher quality of data obtained using the RNA sequencing method. Specifically, the

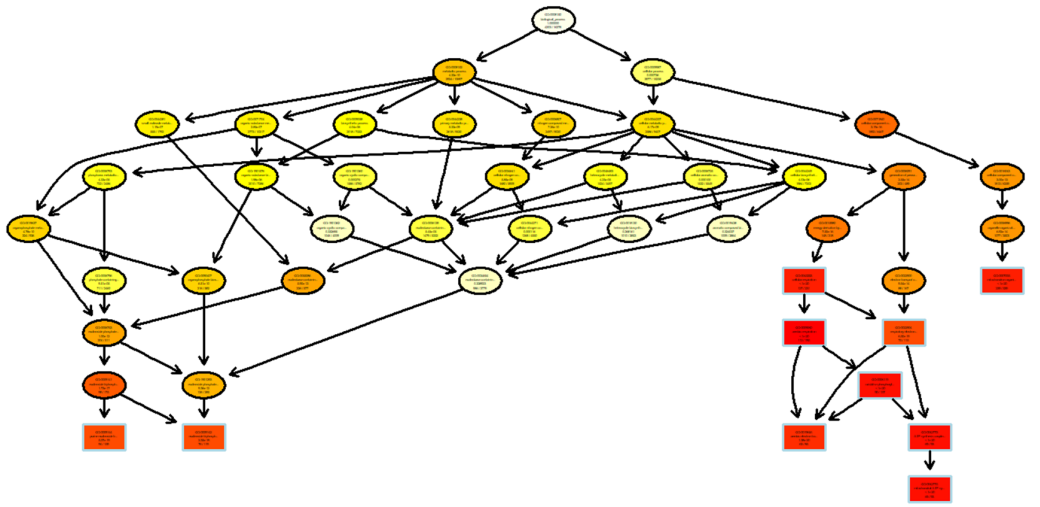

Figure 5.4: Interaction graph of the 10 most significant ontologies (rectangles in red) with each other and with other ontologies

number of annotated and significant genes in the identified ontologies is substantially larger compared to previous data, with the p-values, both in the Fisher's test and the Kolmogorov-Smirnov test, being significantly lower. This indicates a high probability that the identified ontologies are significant. When applying the ANOVA test with correction of the obtained p-values, 21,582 significant ontologies were identified. Figure 5.5 depicts a bubble chart of the distribution of identified ontologies at different p-values obtained using Fisher's and Kolmogorov-Smirnov tests. In this case, the threshold defining the color of the corresponding ontologies was doubled compared to previous studies and was set at 20 genes. Analysis of the obtained chart allows the conclusion that there is a small number of ontologies corresponding to a large number of genes, with several ontologies having maximum significance both in the Fisher's test and the Kolmogorov-Smirnov test. There is also a certain number of significant ontologies in the Kolmogorov-Smirnov test, whose significance in the Fisher's test is lesser. It is also important to highlight an ontology that is insignificant in both tests and also contains a large number of genes. Figure 5.6 shows the interaction graph of the twenty most significant ontologies, with the color intensity in the rectangles indicating the level of significance. Analysis of the distribution pattern of the graph nodes allows the conclusion that there is a higher level of orderliness in the connections between the corresponding ontologies, which may also indicate higher quality of the experimental data.As a result of applying gene ontology analysis, 17,069 significant genes were identified for further research, resulting in the gene expression data matrix taking the form (6344×17069).

Table 5.5: The result of applying gene ontology analysis to the gene expression data of patients investigated for cancer disease

GO:ID	Term	Annot.	Sign.	Fisher's test	KS test
GO:0071840	cellular component organization or biogenesis	6616	6517	<1e-30	<1e-30
GO:0016043	cellular component organization	6410	6311	<1e-30	<1e-30
GO:0051179	localization	5203	5141	<1e-30	<1e-30
GO:0051234	establishment of localization	4572	4518	<1e-30	<1e-30
GO:0051641	cellular localization	3375	3351	<1e-30	<1e-30
GO:0006810	transport	4379	4327	<1e-30	<1e-30
GO:0048518	positive regulation of biological process	6156	6036	<1e-30	<1e-30
GO:0044238	primary metabolic process	9998	9717	<1e-30	<1e-30
GO:0007275	multicellular organism development	4616	4544	<1e-30	<1e-30
GO:0033036	macromolecule localization	2942	2917	<1e-30	<1e-30

5.4 Application of Cluster Analysis for Forming Subsets of Mutually Expressed Gene Expression Profiles

As mentioned in the previous sections, identifying relationships between genes is critical to understanding biological processes and developing new diagnostic and treatment methods. Cluster analysis, which aims to group objects with similar characteristics, is an integral part of this process. At this stage of implementing the proposed information technology, spectral clustering algorithms [78, 73, 86, 62] and the Self-Organizing Tree Algorithm (SOTA) [38, 41] have been applied, which are currently advanced methods in the field of gene expression data processing.

Spectral clustering uses mathematical and statistical methods to detect complex structures in data, providing high accuracy in grouping gene profiles. On the other hand, SOTA employs a hierarchical approach for the gradual refinement of clusters, allowing for detailed analysis of large datasets. The application of these algorithms enables the identification of gene expression patterns, which can lead to increased accuracy and objectivity in identifying objects based on gene expression data.

Spectral clustering is particularly useful for analyzing complex biological data, as it can detect differences in expression profiles. Thanks to its hierarchical structure, SOTA is effective for organizing and visualizing large volumes of genetic data.

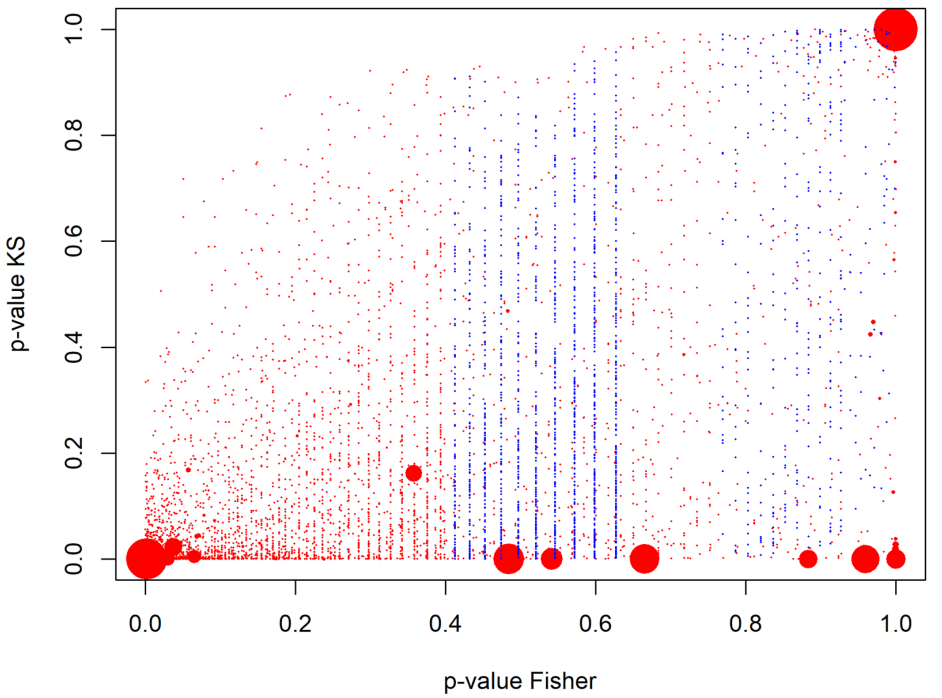

Figure 5.5: A bubble chart of the distribution of identified ontologies at different p-values obtained from applying Fisher's and Kolmogorov-Smirnov tests for the gene expression data of patients studied for cancer diseases

5.4.1 Application of the Spectral Clustering Algorithm for Forming Clusters of Mutually Expressed Gene Expression Profiles

Considering the research results presented hereinbefore, at the current stage of modelling, the Bayesian optimization algorithm was applied to determine the optimal number of clusters when using the spectral clustering algorithm. The objective function, whose maximum value determines the optimal parameters of the clustering algorithm, was based on the internal clustering quality criterion calculated using formulas (3.13) – (3.15), while the distance between gene expression profiles was estimated based on mutual information using formula (3.5). The choice of the Bayesian optimization algorithm at this stage was determined by its lower requirements for computer and time resources compared to the inductive objective clustering technology. However, as shown by the research results presented in section 3.6, the outcomes for selecting optimal hyperparameters do not differ significantlu when using either method.

The process of clustering gene expression profiles was carried out in two stages.

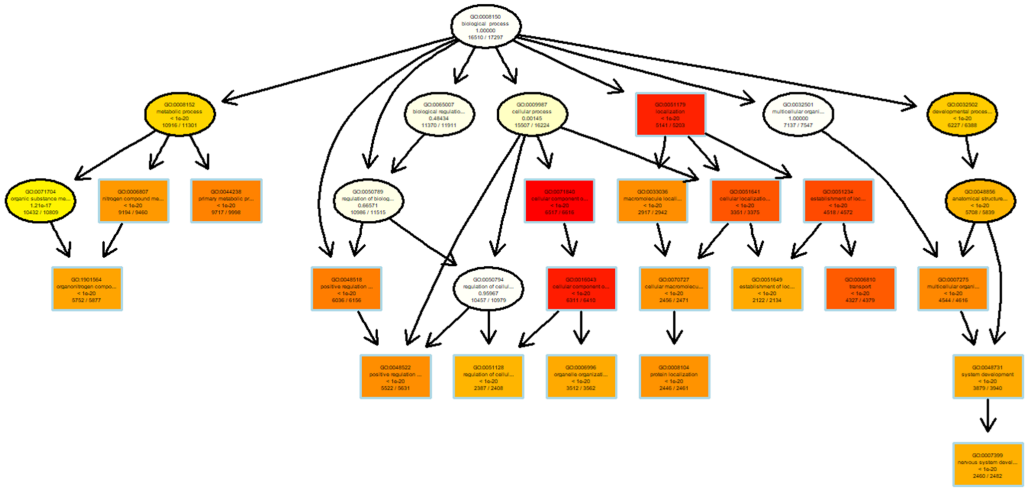

Figure 5.6: Interaction graph of the twenty most significant ontologies using Fisher's test for data studied on cancer diseases

In the first stage, the Bayesian optimization method was applied to the spectral clustering algorithm, with the number of clusters varying from 2 to 10. The optimal number of clusters corresponded to the maximum value of the objective function (the value of the internal criterion with a negative sign). In the second stage, the spectral clustering algorithm with optimal hyperparameters was applied to the gene expression data matrix, followed by the formation of the cluster structure. Figures 5.7 and 5.8 depict the diagrams of the dependence of the absolute value of the objective function on the number of clusters during the operation of the Bayesian optimization method when applying gene expression data from objects studied for Alzheimer's disease and various types of cancer, respectively.

The analysis of the obtained results allowed us to conclude that, in the case of applying gene expression data from subjects studied for Alzheimer's disease, a two-cluster structure is optimal. Increasing the number of clusters results in a decrease in clustering quality according to internal quality criteria. When using gene expression data from subjects studied for various types of cancer, the quality criterion values for forming three- and four-cluster structures are almost identical. However, it should be noted that the four-cluster structure corresponds to a higher quality criterion value. A more detailed analysis also showed that, for this dataset, the three-cluster structure corresponds to a lower quality criterion value. Therefore, for data from subjects studied for various types of cancer, a three-cluster structure was identified as optimal.

Based on the modelling results, subsets of gene expression data for each disease type have been formed for further processing according to the proposed information

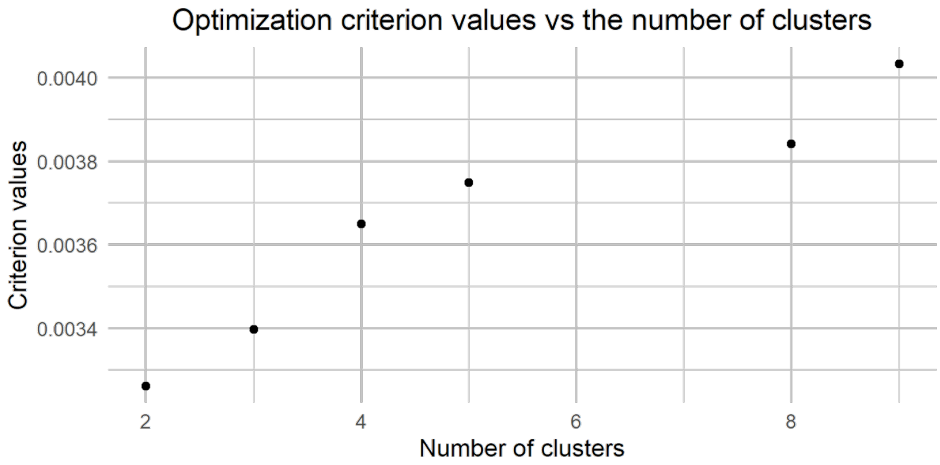

Figure 5.7: The modeling results regarding the application of the Bayesian optimization algorithm to the gene expression data of samples being studied for Alzheimer's disease using the spectral clustering algorithm

technology:

- Gene expression data of objects studied for Alzheimer's disease: 161×5074; 161×9527;
- Gene expression data of objects studied for various types of cancer: 6344×17046; 6344×15; 6344×8.

It should be noted that applying the spectral clustering algorithm to the gene expression data studied for various types of cancer proved inefficient. In fact, one large cluster containing 17046 genes was identified. This can be explained by the high dimensionality of the gene expression profiles (6344), which undoubtedly affects the nature of the distance matrix formation between profiles, which is crucial in the subsequent formation of clusters. In this case, two small clusters contain gene expression profiles that differ significantly from the profiles of the main cluster. Given the small number of these genes, clusters 2 and 3 were not considered in the subsequent stages of the information technology implementation.

5.4.2 Application of the SOTA Clustering Algorithm for Forming Clusters of Mutually Expressed Gene Expression Profiles

As mentioned above, the SOTA (Self-Organizing Tree Algorithm) clustering algorithm is currently one of the modern self-organizing algorithms aimed at processing high-dimensional data. It is a clustering method that uses the principles of neural

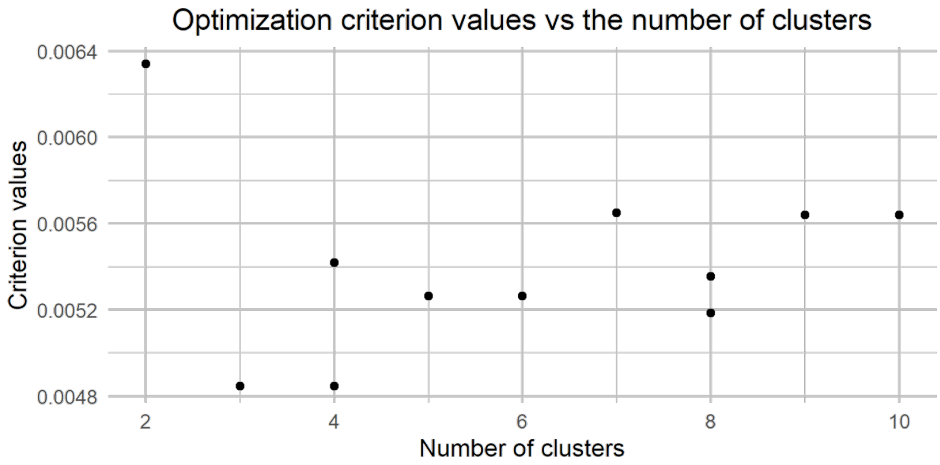

Figure 5.8: The modeling results regarding the application of the Bayesian optimization algorithm to the gene expression data of samples being studied for cancer disease using the spectral clustering algorithm

network self-organization, based on the ideas of Kohonen maps [38] and Fritzke's growing cell structures method [41]. Unlike Kohonen maps, which transform high-dimensional input data into a two-dimensional array of small dimensions, SOTA forms a binary topological tree that reflects the data structure according to Fritzke's cell growth principles. This process implies that the number of network nodes grows in regions with high object density, while in less dense areas, the number of nodes remains unchanged, allowing for the consideration of uneven object distribution.

The practical implementation of the SOTA algorithm involves the following steps:

1. Initialization

At this stage, weights are assigned to the root node and cells based on the average value of all columns of the studied data. This means that the length of the weight vector corresponds to the number of features in the studied data. Parameters are also set for the weight correction of the winning cell, root node, and neighbouring cells: $w_{cell} > p_{cell} > s_{cell}$, where w_{cell}, p_{cell}, and s_{cell} are parameters for the weight correction of the winning cell, root (parent) node, and neighbouring cell, respectively. At this stage, the threshold value of the threshold coefficient E is also determined, which is essential for determining the stopping point of the algorithm.

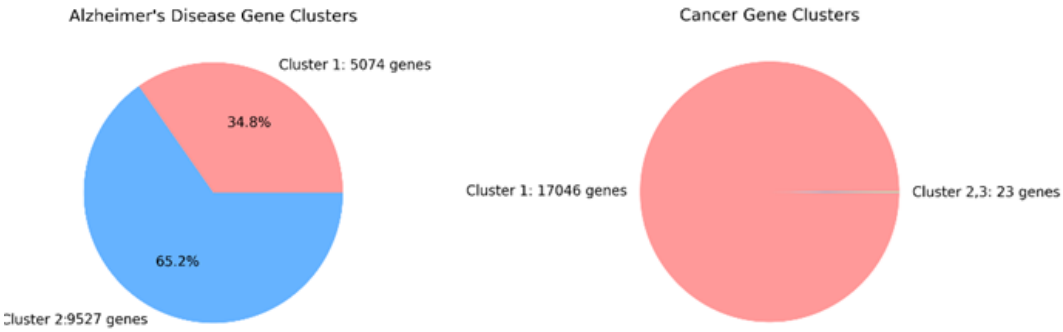

Figure 5.9: The result of applying the spectral clustering algorithm to the gene expression data of objects studied for Alzheimer's disease and various types of cancer

2. Adaptation

During the algorithm operation, feature vectors are sequentially fed to the input of external cells. The degree of similarity between these vectors and the weight vectors of the cells is calculated using the chosen proximity function. According to the "winner takes all" principle, the winning cell is determined whose weight vector has the smallest distance to the submitted vector. The weights of this cell and its neighbours (neighbouring and root cells) are adjusted. If the neighbouring cell has no descendants, the weights of the winning cell, a neighbouring cell, and a root node are adapted. If the neighbouring cell has descendants, only the weights of the winning cell are adjusted.

3. Convergence of the Algorithm and Formation of the Distribution Tree

At this stage, the structure of the clustering tree is determined by calculating the variation coefficient for each cell as the arithmetic mean of the distances from the cell weights to the feature vectors in that cell, which allows for assessing the average deviation of the cell weights from the feature vectors:

$$R_i = \frac{\sum_{n=1}^{N} d(p_k, C_i)}{N} \tag{5.2}$$

where N is the number of feature vectors in the i-th cell; p_k is the k-th feature vector in this cell; C_i is the weight vector of the i-th cell; $d(\cdot)$ is the proximity function between vectors used in the algorithm.

The total value of the variation coefficient for all external cells at step t is deter-

mined as the sum of individual variation coefficients:

$$\epsilon_t = \sum_{i=1}^{K} R_i \tag{5.3}$$

A key indicator of the algorithm's convergence is the relative change of this total variation coefficient:

$$\left| \frac{\epsilon_t - \epsilon_{t-1}}{\epsilon_{t-1}} \right| \leq E \tag{5.4}$$

where E is the threshold value of the relative change of the variation coefficient.

If the relative change does not reach the specified threshold E, the cell with the highest relative change coefficient is selected for further splitting into two parts, forming a new node. The process continues until the convergence criterion is reached, allowing the desired network structure and object distribution into clusters to be formed or until the maximum number of iterations is reached.

The analysis of the above procedure allows us to conclude that the character of the object grouping using the SOTA is determined by parameters for the correction of cell weights *wcell*, *pcell*, and *scell*, as well as by the threshold value of the relative change in the coefficient of variation E, which defines the stopping point of the algorithm. Within the scope of current research, the task of determining the optimal hyperparameters of the algorithm is solved using the Bayesian optimization algorithm. Taking into account the recommendations of the algorithm's authors and the results presented in [21], the following relationship between the weights of the corresponding cell vectors was adopted:

$$pcell = scell \cdot 5; \quad wcell = pcell \cdot 2 \tag{5.5}$$

When applying the correlation metric as a proximity function, the threshold value of the coefficient of variation was set to zero. In this case, the algorithm stopped upon the repetition of two consecutive configurations. Preliminary results of the modeling confirmed the appropriateness of this approach. Thus, within the current research, only the value of the hyperparameter *scell* was optimized. Figures 5.10 and 5.11 illustrate the results of applying the Bayesian optimization algorithm to determine the optimal value of the hyperparameter scell using gene expression data of patients studied for Alzheimer's disease, Parkinson's disease, and various types of cancer, respectively. The value of the parameter scell was varied during the modeling within the range from 0.0001 to 0.01. As a simulation result, the following values of the hyperparameter *scell* were determined:

- for the gene expression data of patients studied for Alzheimer's disease: 0.0295;
- for the gene expression data of patients studied for various types of cancer: 0.00229.

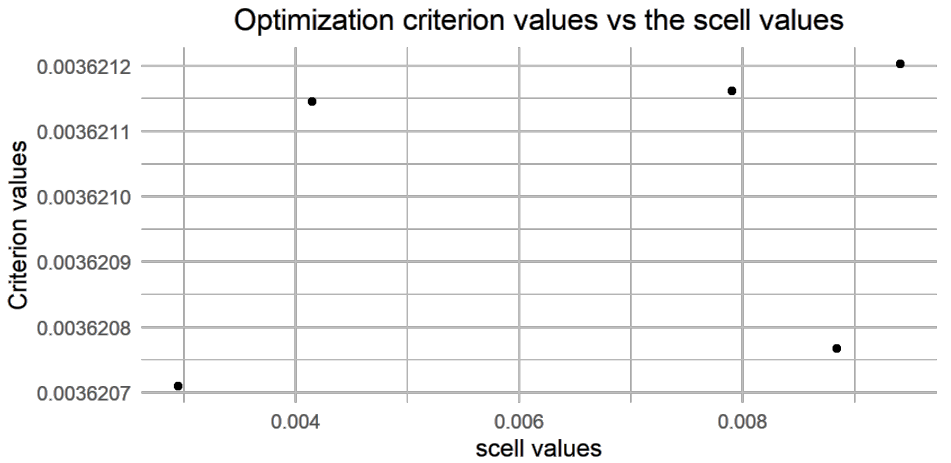

Figure 5.10: The simulation results regarding the application of the Bayesian optimization algorithm to the gene expression data of patients being investigated for Alzheimer's disease

The result of applying the SOTA clustering algorithm with optimal hyperparameter values to the studied gene expression data is depicted in Figure 5.12.

The analysis of the obtained results allows concluding that the application of the SOTA algorithm in all cases led to the gene expression profiles being divided into two clusters. This division enabled the formation of two subsets of gene expression data for each dataset:

- for the gene expression data of patients studied for Alzheimer's disease: (161×8453); (161×6148);
- for the gene expression data of patients studied for various types of cancer: (6344×7442); (6344×9327).

5.5 Application of Biclustering and Gene Ontology Analysis for Forming Subsets of Significant and Co-Expressed Gene Expression Data

The practical implementation of bicluster analysis for forming subsets of significant and mutually correlated gene expression data within the proposed information technology involves four stages. At the first stage, the biclustering model is configured to optimize the hyperparameters of the corresponding biclustering algorithm using Bayesian optimization. The objective function is calculated using a metric based on the mutual information estimation method as presented in section 3.5.2, formu-

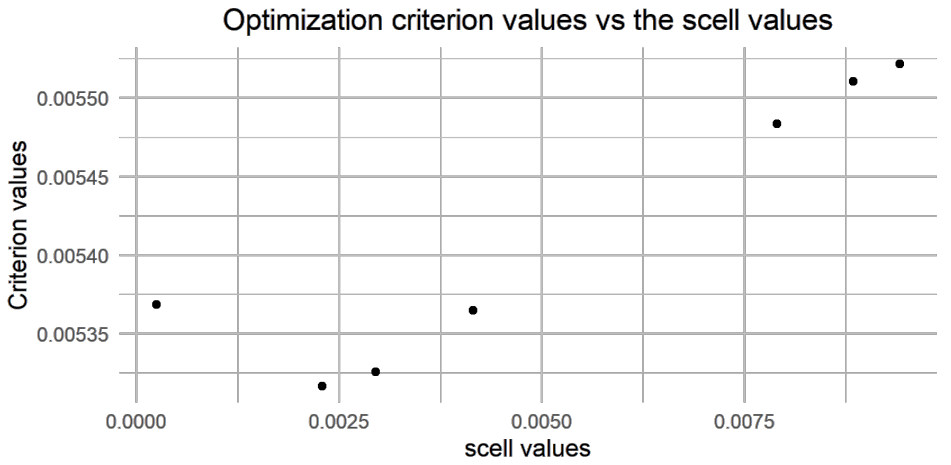

Figure 5.11: The simulation results regarding the application of the Bayesian optimization algorithm to the gene expression data of patients being investigated for cancer disease

las (3.33) – (3.36). At the second stage, the biclustering algorithm with optimal hyperparameter values is applied to the corresponding gene expression data, forming biclusters of coherent gene expression data. At the third stage, gene ontology analysis (function *entrihGO()*) is applied to each bicluster's data to form a list of significant genes. This results in the extraction of a list of unique gene identifiers for the studied data. At the fourth stage, gene expression data is formed, the attributes of which are the significant genes identified in the previous step using the gene ontology analysis.

5.5.1 Application of Bicluster and GO Analysis to Gene Expression Data of Objects Studied for Alzheimer's Disease

As a result of applying both the spectral clustering and the SOTA algorithms, the gene expression profiles of samples studied for Alzheimer's disease were divided into two clusters. According to the research results presented in section 3.5 of this thesis, the *ensemble* biclustering algorithm was applied to the gene expression data. The modelling was carried out in the R software environment using functions from the *Biclust* package. Figure 5.13 shows the results of applying the Bayesian optimization method to optimize two key hyperparameters of the ensemble algorithm: thr and simthr for the gene expression data studied for Alzheimer's disease.

The modelling results determined the optimal hyperparameters of the ensemble biclustering algorithm, which were subsequently used to form the bicluster structure for each type of data. The optimal hyperparameter values are presented in Table 5.6.

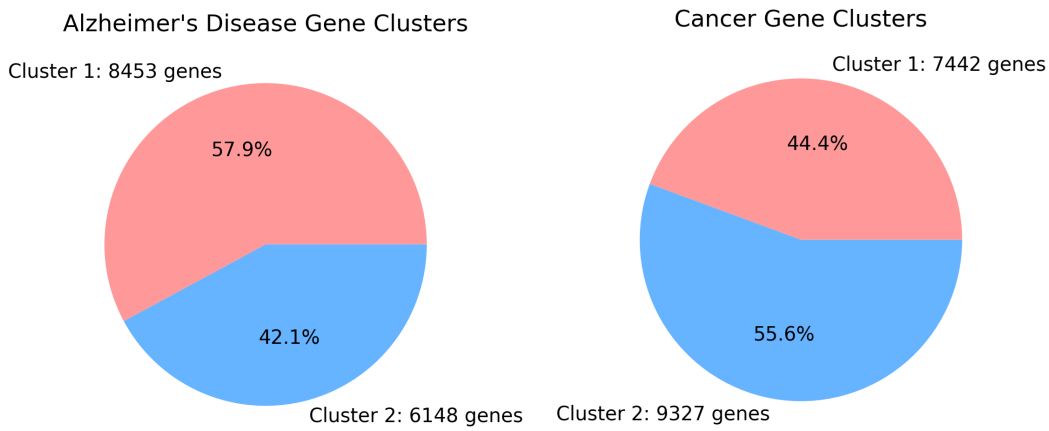

Figure 5.12: The result of applying the SOTA clustering algorithm to the gene expression data of patients studied for Alzheimer's disease and various types of cancer

Table 5.6: Optimal hyperparameters of the *ensemble* biclustering algorithm using gene expression data of samples studied for Alzheimer's disease

Hyperparameter	**Spectral Clustering**		**SOTA**	
	Cluster 1	**Cluster 2**	**Cluster 1**	**Cluster 2**
thr	0.175	0.539	0.106	0.249
simthr	0.107	0.216	0.101	0.173

Considering the results of bicluster analysis using biclusters obtained with the spectral clustering algorithm, 21 and 9 biclusters were identified when analyzing the gene expression data of the first and second clusters, respectively. When analyzing the data of clusters obtained using the SOTA clustering algorithm, 7 and 62 biclusters were identified when applying the gene expression data of the first and second clusters, respectively. When applying gene ontology analysis at the third stage of this procedure, the p-value threshold separating significant and non-significant genes was set at 0.05, meaning that the identified genes were deemed significant with a 95% probability. Table 5.7 presents the structure of the formed gene expression data for objects studied for Alzheimer's disease.

Table 5.7: Results of bicluster and GO analysis for gene expression data of samples studied for Alzheimer's disease

Results	Spectral Clustering		SOTA	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
Number of Biclusters	21	9	7	62
Data Structure	(161 × 2845)	(161 × 474)	(161 × 1158)	(161 × 2846)

5.5.2 Application of Bicluster and GO Analysis to Gene Expression Data of Objects Studied for Cancer Disease

As mentioned above, in the case of processing gene expression data of objects studied for various types of cancer, during the bicluster analysis stage followed by gene ontology analysis applied to the data in the identified biclusters, the gene expression data of the first cluster (17046 genes) were used, as well as the first (7742 genes) and second (9326 genes) clusters when using the spectral clustering algorithm and the SOTA algorithm, respectively. Figure 5.14 shows the result of applying the Bayesian optimization algorithm to determine the optimal hyperparameters of the ensemble biclustering algorithm for each type of data. The modelling results for each data subset are presented in Table 5.8.

Table 5.8: Optimal hyperparameters of the *ensemble* biclustering algorithm using gene expression data studied for various types of cancer

Hyperparameter	Spectral Clustering	SOTA	
	Cluster 1	Cluster 1	Cluster 2
thr	0.091	0.108	0.423
simthr	0.400	0.268	0.400

Based on the results of bicluster analysis using clusters obtained with the spectral clustering algorithm, 83 biclusters were identified. Using clusters obtained with the SOTA algorithm, 47 biclusters were identified from the first cluster data and 74 biclusters from the second cluster data. The results of applying gene ontology analysis to the data in the identified biclusters for the final formation of gene expression data subsets of objects studied for various types of cancer are presented in Table 5.9.

The analysis of the obtained results allows us to conclude that in all cases, a sufficiently large number of biclusters were identified, indicating the presence of numerous subsets of mutually coherent gene expression data. The application of gene ontology analysis, in this case, allowed the formation of subsets of mutually correlated and significant genes for their further use to identify the state of the studied objects.

Table 5.9: Modelling results regarding the formation of subsets of significant and mutually expressed gene expression data of objects studied for various types of cancer

Results	Spectral Clustering	SOTA	
	Cluster 1	Cluster 1	Cluster 2
Number of Biclusters	83	47	74
Data Structure	(6344 × 10124)	(6344 × 3955)	(6344 × 6467)

5.6 Application of CNN to Identify Samples Based on Formed Subsets of Gene Expression Data

The final step in implementing the proposed information technology is the identification of samples, the attributes of which are gene expression data formed based on the comprehensive application of cluster-bicluster analysis and gene ontology analysis. In previous sections of the thesis, the classification procedure was performed using various deep-learning models. The modelling results showed that when a large number of genes are used as attributes, recurrent neural networks (RNN) demonstrate higher efficiency compared to convolutional neural networks (CNN) and hybrid models based on the combined application of various machine learning models. However, reducing the number of genes revealed a high sensitivity of RNNs to overfitting, which complicated their use for correctly identifying objects. On the other hand, CNN showed high resistance to overfitting, given proper model training. Considering the obtained results, CNNs were used at this research stage, with optimal model hyperparameters for each data type determined using the Bayesian optimization method with k-fold cross-validation. The value of k for each data type was determined based on the number of samples in the corresponding data. To evaluate the effectiveness of the proposed methodology, sample classification in all cases was performed using the complete set of significant genes (after the application of the first filtration step using gene ontology analysis) and subsets of gene expression data obtained in the previous stage of the implementation of the information technology.

5.6.1 Identification of objects' state based on gene expression data of samples studied for Alzheimer's disease

Table 5.10 presents the results of applying the Bayesian optimization algorithm with 5-fold cross-validation at each epoch of the algorithm's operation for gene expression data of samples studied for Alzheimer's disease. During the model's operation, the optimal hyperparameters of a two-layer convolutional neural network were determined for the complete data set and subsets formed using the spectral clustering and SOTA clustering algorithms. Figure 5.15 shows diagrams of changes in the accuracy of sample classification and the loss function values, which were

Table 5.10: Results of applying the Bayesian optimization algorithm to gene expression data of samples studied for Alzheimer's disease

Hyperparameters	**Data Type**				
	Full data	**SP 1**	**SP 2**	**SOTA 1**	**SOTA 2**
Filter_size 1	44	32	40	48	62
Kernel_1	1	3	14	3	3
Max_pool_1	4	3	4	2	4
Filter_size 2	9	64	32	41	26
Kernel_2	6	14	14	14	14
Max_pool_2	4	3	2	4	4
Dropout	0.2	0	0	0.1	0.1
Dense	27	59	75	24	58

calculated for the complete data during the CNN training stage.

The analysis of the obtained diagrams indicates the absence of model overfitting since the classification accuracy and loss function values, calculated on the data subsets for training and validating the model during the network training process implementation, change consistently within an acceptable range. Figure 5.16 shows the classification results of samples that make up the test subset of the complete gene expression data studied for Alzheimer's disease. The analysis of the results indicates a low effectiveness of the classifier when applying the complete gene expression data set. Nine of the 49 samples that made up the test subset were incorrectly identified. The classification accuracy in this case was only 81.6%. Figures 5.17 - 5.20 show the classification results of samples whose attributes are data subsets of genes expression formed using the proposed information technology.

Table 5.11 presents a criterial analysis of the classification results of all types of gene expression data used at this modelling stage. The analysis of the obtained results allows us to conclude that the classification results are better in almost all cases when using subsets of gene expression data formed using the proposed information technology. The exception is the second cluster obtained using the spectral clustering algorithm. It contained only 474 genes, which is significantly fewer compared to the number of genes in other data subsets. This fact may affect the classification results of the samples. The highest classification accuracy is achieved when the first subset of data is formed using the spectral clustering algorithm. Only three of the 49 samples in the test data subset were incorrectly identified. However, it should be noted that when using the gene expression data subsets obtained using the SOTA clustering algorithm, the classification results are also quite high in both cases. Out of 49 samples, five were incorrectly identified in the first case and four in the second. The classification accuracy was 89.8% and 91.8%, respectively. When making

Table 5.11: Classification results of gene expression data from samples studied for Alzheimer's disease

Class	**Full Gene Expression Data Set (Alzheimer's Disease)**					
	Num.genes	Precision	Recall	F1-score	Accuracy	AUC
Disease	14602	0.871	0.852	0.836	0.816	0.81
Normal		0.810	0.773	0.773		
	First Subset, Spectral Clustering Algorithm					
Disease	2845	0.929	0.963	0.945	0.939	0.94
Normal		0.952	0.909	0.930		
	Second Subset, Spectral Clustering Algorithm					
Disease	474	0.750	0.889	0.814	0.776	0.76
Normal		0.824	0.824	0.636		
	First Subset, SOTA Clustering Algorithm					
Disease	1158	0.923	0.889	0.906	0.898	0.90
Normal		0.870	0.909	0.889		
	Second Subset, SOTA Clustering Algorithm					
Disease	2846	0.926	0.926	0.926	0.918	0.92
Normal		0.909	0.909	0.909		

a compromise decision regarding the state of the studied objects, the application of the SOTA clustering algorithm is more attractive.

5.6.2 Identification of objects' state based on gene expression data of samples studied for cancer disease

In contrast to previous data, the gene expression data of objects studied for different types of cancer were obtained using the RNA molecules sequencing method, which inherently has significantly higher accuracy compared to the DNA microarray method. Table 5.12 presents the results of applying the Bayesian optimization algorithm with 10-fold cross-validation for gene expression data of samples studied for different types of cancer.

Figures 5.21 and 5.22 show the training results of the model on training and validation data created using the complete data set and the first data subset formed using the SOTA clustering algorithm. Similar diagrams were obtained when using other data subsets. The analysis of the obtained diagrams indicates the absence of model overfitting, too, since the average values of both classification accuracy and loss function, calculated on the data subsets for training and validation, change consistently during the neural network training process.

Figures 5.23 - 5.26 and Tables 5.13 - 5.16 show the results of sample identification

Table 5.12: Results of applying the Bayesian optimization algorithm to gene expression data of samples studied for different types of cancer

Hyperparameters	**Data Type**			
	Full data	**SP**	**SOTA 1**	**SOTA 2**
Filter_size 1	35	31	40	39
Kernel_1	4	4	5	14
Max_pool_1	2	3	3	3
Filter_size 2	42	44	56	38
Kernel_2	7	6	12	14
Max_pool_2	4	3	2	3
Dropout	0.01	0.12	0.1	0
Dense	231	169	40	45

using test data from the complete gene expression data set and subsets obtained using the spectral and the SOTA clustering algorithms.

Table 5.13: Classification results of objects based on the full dataset of gene expression data from patients studied for various types of cancer diseases

Class	Number of samples	Falsely identified	Classification quality criteria			
			Precision	Recall	F1-score	Accuracy
BLCA	137	6	0.949	0.956	0.953	0.975
BRCA	346	2	0.994	0.994	0.994	
CESC	93	7	0.966	0.925	0.945	
COAD	139	0	0.993	1.000	0.966	
ESCA	48	0	0.960	1.000	0.980	
GBM	52	2	0.962	0.962	0.962	
HNSC	157	6	0.981	0.962	0.971	
KIRC	155	0	0.994	1.000	0.997	
LAML	51	0	1.000	1.000	1.000	
LGG	158	2	0.987	0.987	0.987	
LIHC	100	1	0.971	0.990	0.980	
LUSC	166	8	0.952	0.952	0.952	
LUAD	160	9	0.944	0.944	0.944	
NORMAL	142	4	0.972	0.972	0.972	

The analysis of the obtained results allows us to conclude that the accuracy of sample identification in all cases is high both for individual classes (F1-score values) and for all classes as a whole. This confirms the high quality of the gene expression data obtained using the RNA molecules sequencing method. Moreover, the high results of sample identification based on selected subsets of gene expression data indicate the high efficiency of the proposed information technology for forming subsets of significant and mutually correlated gene expression data. As in the

Table 5.14: Classification results of objects based on the subset of gene expression data from patients studied for various types of cancer diseases formed using the spectral clustering algorithm

Class	Number of samples	Falsely identified	Classification quality criteria			
			Precision	Recall	F1-score	Accuracy
BLCA	137	6	0.949	0.956	0.953	0.974
BRCA	346	3	0.994	0.991	0.993	
CESC	93	7	0.956	0.925	0.940	
COAD	139	0	0.993	1.000	0.996	
ESCA	48	0	0.960	1.000	0.980	
GBM	52	2	0.962	0.962	0.962	
HNSC	157	7	0.974	0.955	0.965	
KIRC	155	0	0.994	1.000	0.997	
LAML	51	0	1.000	1.000	1.000	
LGG	158	2	0.987	0.987	0.987	
LIHC	100	1	0.961	0.990	0.975	
LUSC	166	8	0.952	0.952	0.952	
LUAD	160	10	0.943	0.938	0.940	
NORMAL	142	4	0.972	0.972	0.972	

case of applying gene expression data of objects studied for Alzheimer's disease, the obtained results allow us to conclude the greater attractiveness of the SOTA clustering algorithm compared to the spectral clustering algorithm. When applying the spectral clustering algorithm, additional data filtering was effectively implemented by removing the gene expression profiles of small clusters. The application of the SOTA algorithm allowed the separation of gene expression profiles into two clusters, considering the level of their mutual correlation. It made creating two subsets of gene expression data possible for further use in a classifier. This fact contributes to increasing the objectivity of the final decision regarding the state of the studied object by parallel evaluation of clustering results on two subsets of gene expression data to make a compromise decision. It should be noted that both subsets of gene expression data are significant and mutually correlated.

Table 5.15: Classification results of objects based on the first gene expression data subset from patients studied for various types of cancer diseases formed using the SOTA clustering algorithm

Class	Number of samples	Falsely identified	Classification quality criteria			
			Precision	Recall	F1-score	Accuracy
BLCA	137	7	0.949	0.949	0.949	0.976
BRCA	346	2	0.997	0.994	0.996	
CESC	93	4	0.978	0.957	0.967	
COAD	139	1	0.993	0.993	0.993	
ESCA	48	0	0.980	1.000	0.990	
GBM	52	4	0.923	0.923	0.923	
HNSC	157	2	0.987	0.987	0.987	
KIRC	155	0	1.000	1.000	1.000	
LAML	51	0	1.000	1.000	1.000	
LGG	158	4	0.981	0.975	0.978	
LIHC	100	2	0.990	0.980	0.985	
LUSC	166	6	0.952	0.964	0.958	
LUAD	160	13	0.974	0.919	0.945	
NORMAL	142	1	0.922	0.993	0.956	

Table 5.16: Classification results of objects based on the second gene expression data subset from patients studied for various types of cancer diseases formed using the SOTA clustering algorithm

Class	Number of samples	Falsely identified	Classification quality criteria			
			Precision	Recall	F1-score	Accuracy
BLCA	137	6	0.942	0.956	0.949	0.970
BRCA	346	1	0.994	0.997	0.996	
CESC	93	9	0.955	0.903	0.928	
COAD	139	0	0.993	1.000	0.996	
ESCA	48	0	0.980	1.000	0.990	
GBM	52	6	0.920	0.885	0.902	
HNSC	157	10	0.987	0.936	0.961	
KIRC	155	0	0.994	1.000	0.997	
LAML	51	0	1.000	1.000	1.000	
LGG	158	4	0.969	0.975	0.972	
LIHC	100	1	0.980	0.990	0.985	
LUSC	166	8	0.924	0.952	0.938	
LUAD	160	11	0.937	0.931	0.934	
NORMAL	142	2	0.966	0.986	0.976	

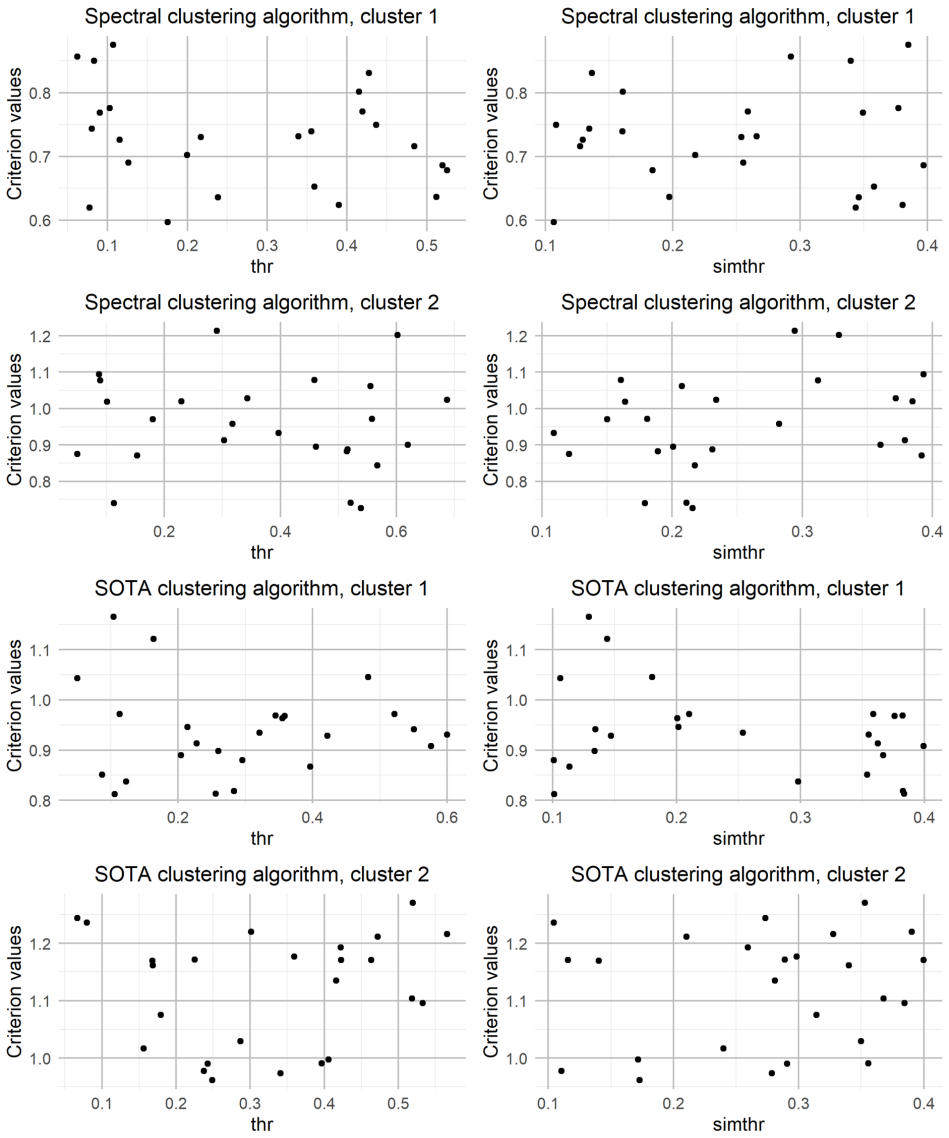

Figure 5.13: Results of modelling the application of Bayesian optimization algorithm for determining optimal parameters of the *ensemble* algorithm for gene expression data studied for Alzheimer's disease

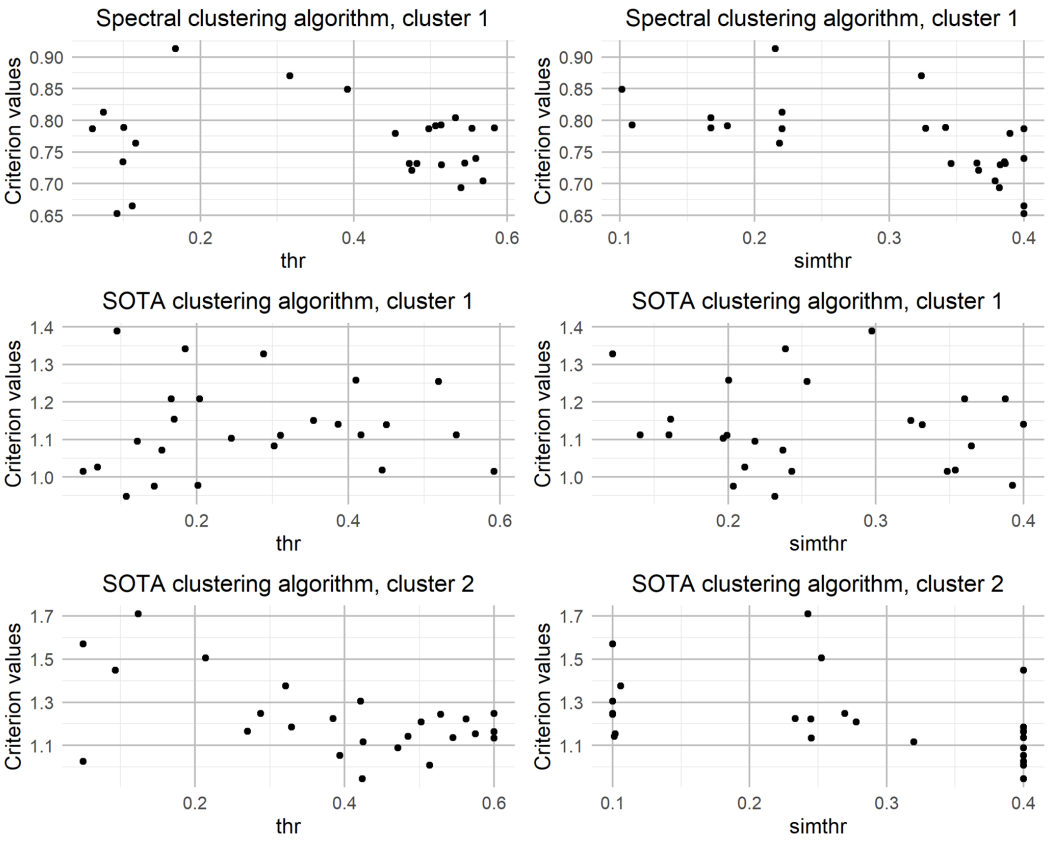

Figure 5.14: Results of modelling the application of Bayesian optimization algorithm for determining optimal parameters of the *ensemble* algorithm for gene expression data studied for cancer disease

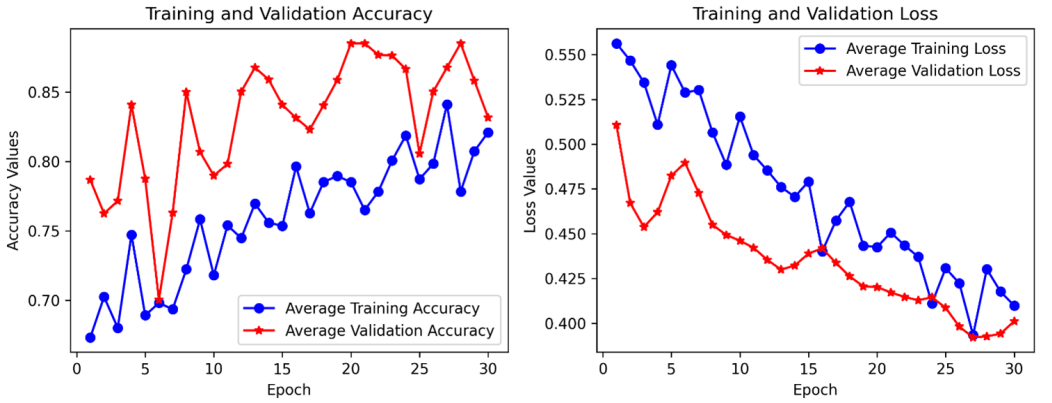

Figure 5.15: Diagrams of changes in the classification accuracy and loss function values calculated for the full gene expression data during the CNN training process implementation

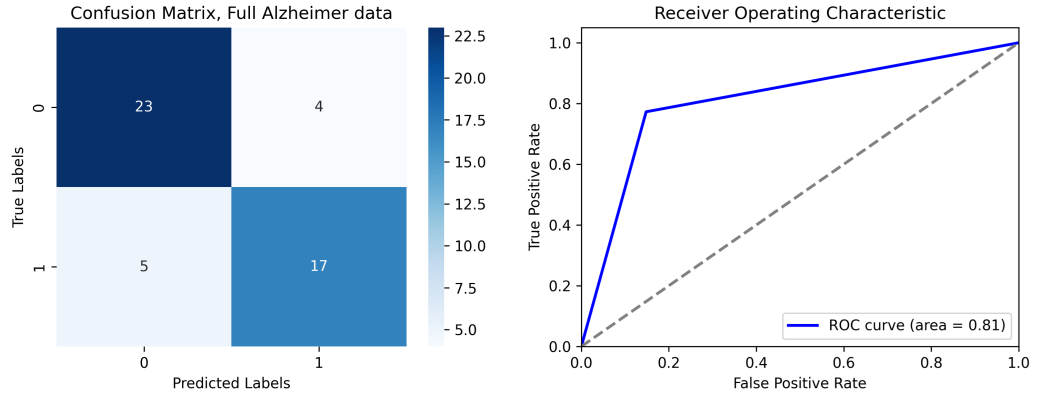

Figure 5.16: Classification results of the test subset samples of the complete gene expression data of samples studied for Alzheimer's disease

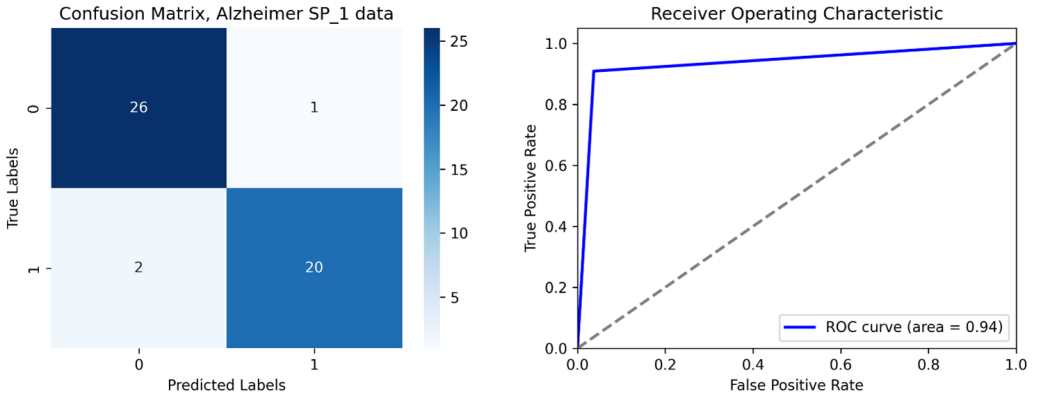

Figure 5.17: Classification results of the test subset samples of gene expression data for objects in the first cluster, obtained using the spectral clustering algorithm

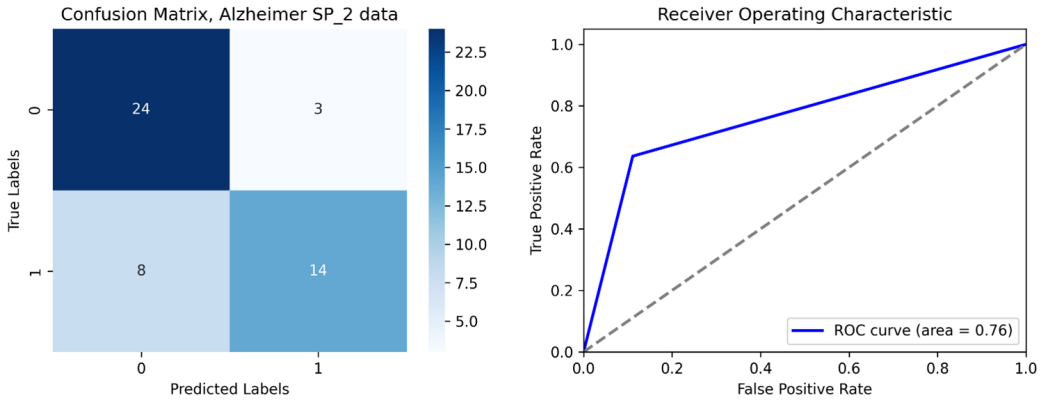

Figure 5.18: Classification results of the test subset samples of gene expression data for objects in the second cluster, obtained using the spectral clustering algorithm

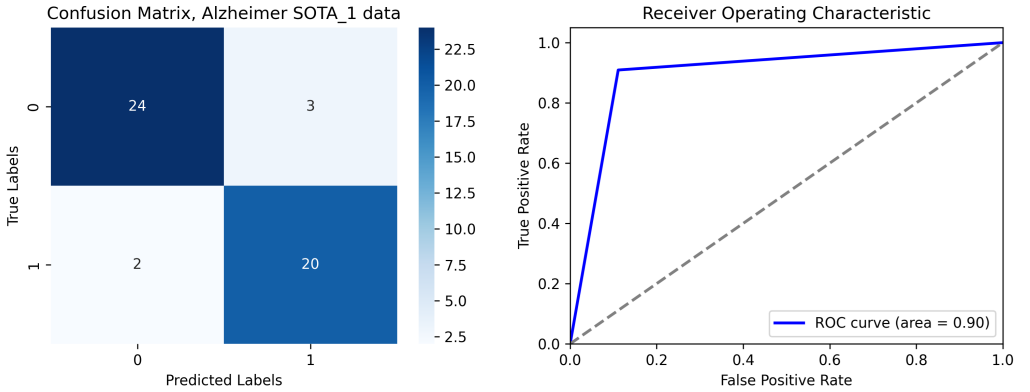

Figure 5.19: Classification results of the test subset samples of gene expression data for objects in the first cluster, obtained using the SOTA clustering algorithm

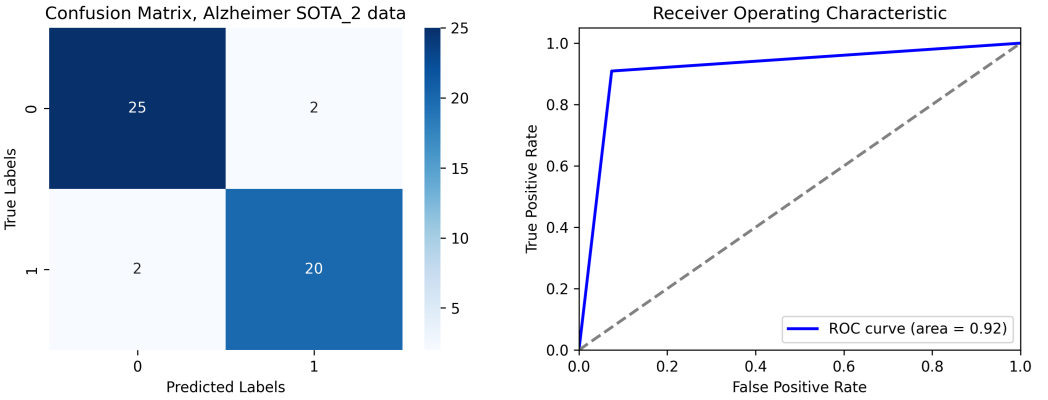

Figure 5.20: Classification results of the test subset samples of gene expression data for objects in the second cluster, obtained using the SOTA clustering algorithm

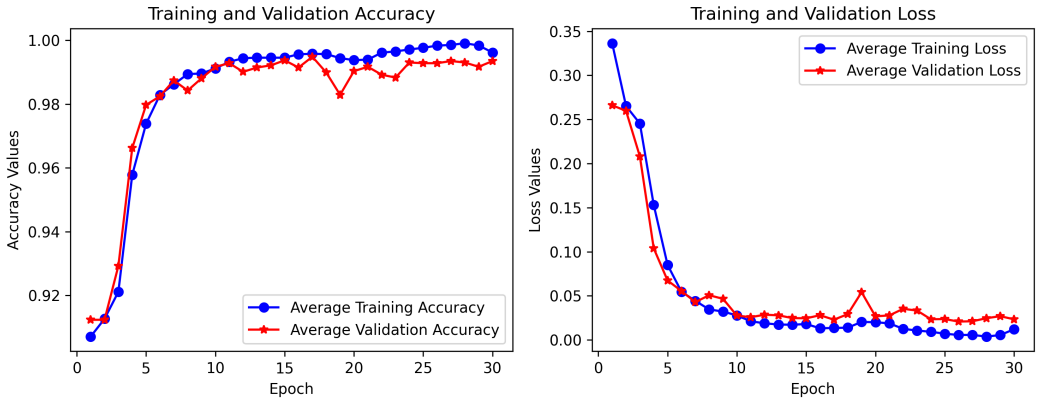

Figure 5.21: Diagrams of changes in the classification accuracy and loss function values calculated for the full gene expression data during the CNN training process implementation

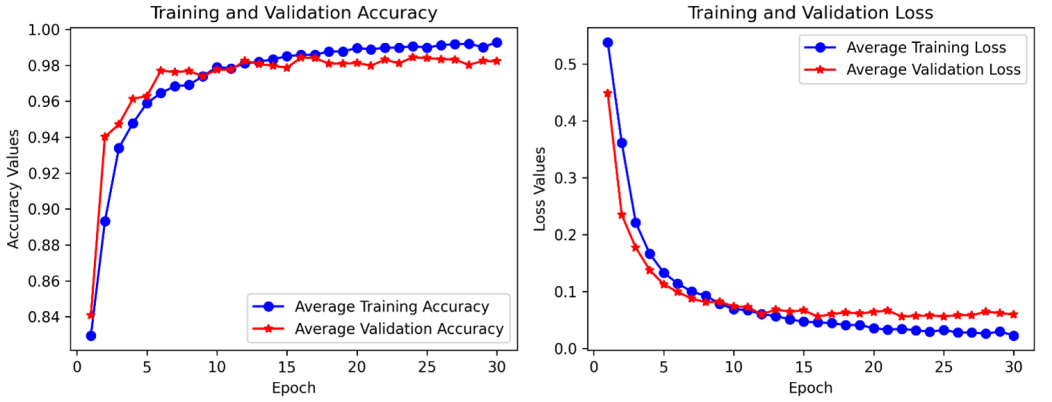

Figure 5.22: Diagrams of changes in the classification accuracy and loss function values calculated for the first data subset formed using the SOTA clustering algorithm during the CNN training process implementation

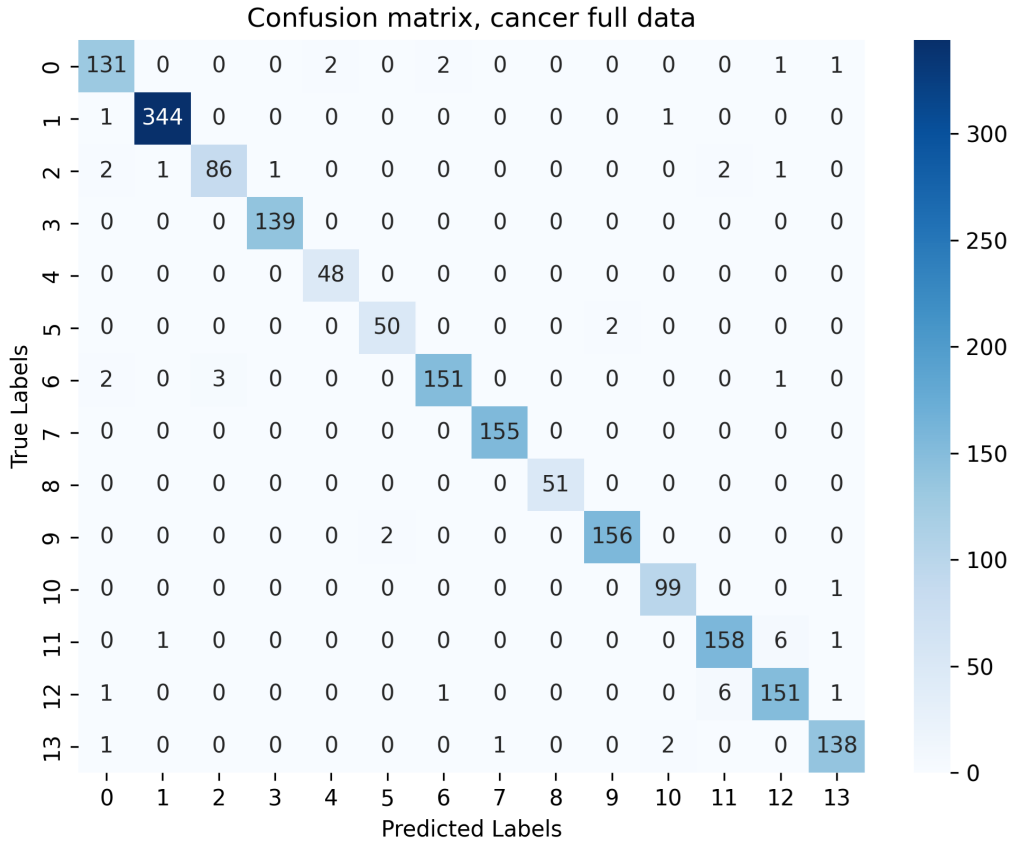

Figure 5.23: Modelling results for sample identification based on the complete set of gene expression data studied for various types of cancer (test subset)

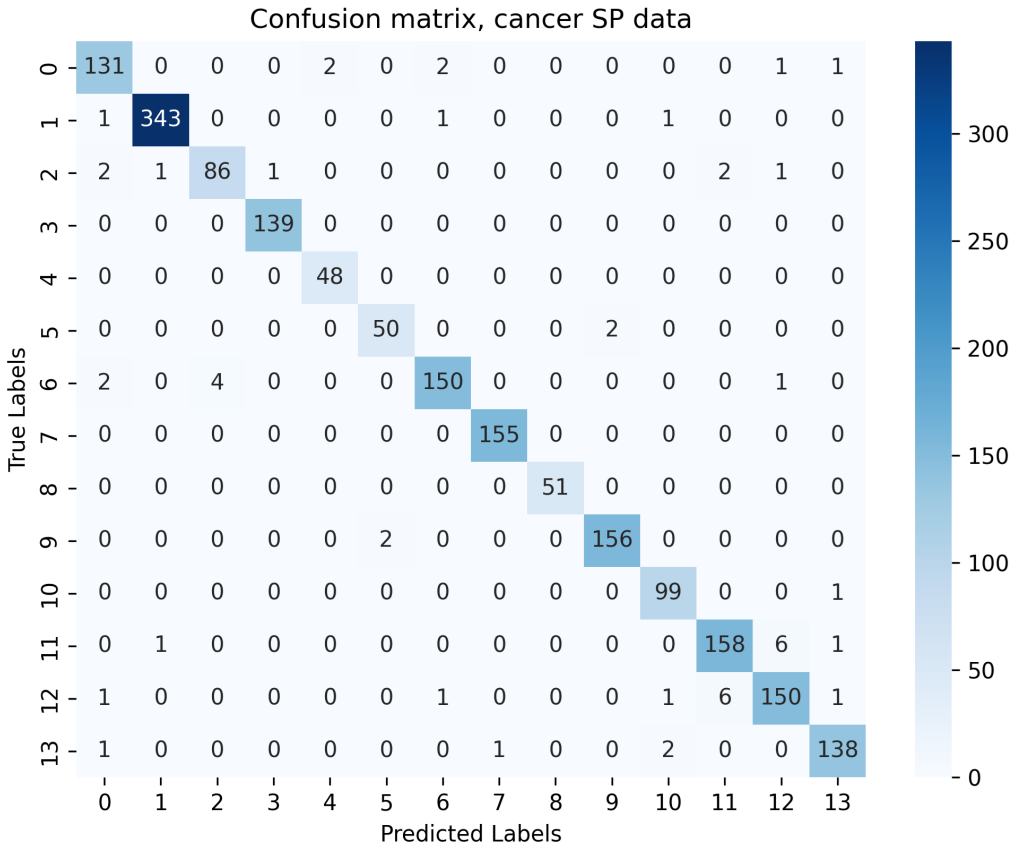

Figure 5.24: The result of modelling on the identification of samples based on a subset of the gene expression data of the objects studied for different types of cancer (test subset) and formed using the spectral clustering algorithm

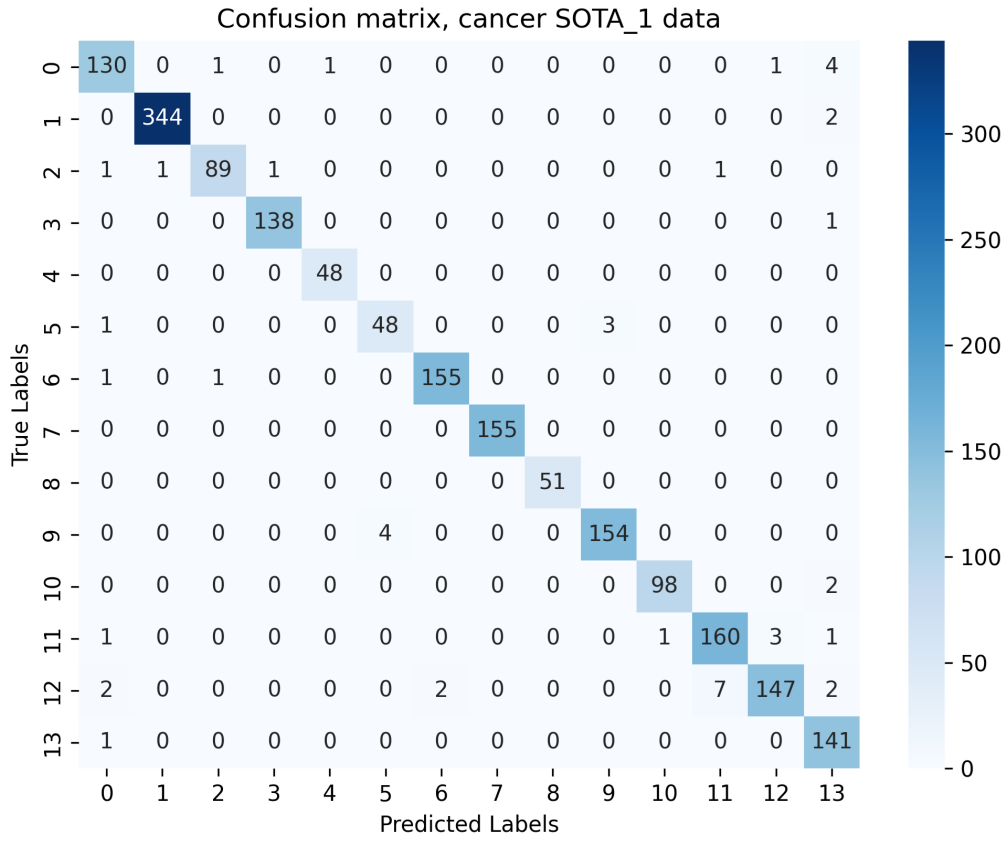

Figure 5.25: The result of the simulation on the identification of samples based on the first subset of gene expression data of objects studied for different types of cancer (test subset) and formed using the SOTA clustering algorithm

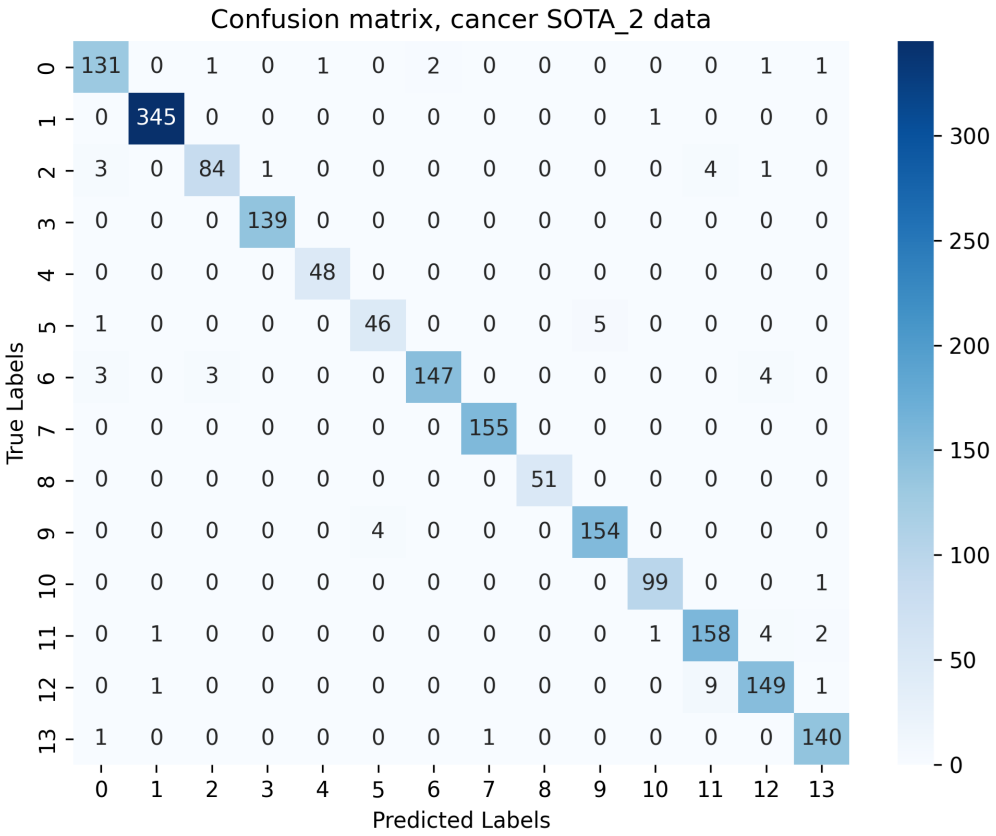

Figure 5.26: The result of the simulation on the identification of samples based on the second subset of gene expression data of objects studied for different types of cancer (test subset) and formed using the SOTA clustering algorithm

Chapter 6

Conclusions and Final Remarks

Based on the theoretical and experimental research carried out, a significant scientific and applied problem in the field of applied information technologies has been solved: the development and practical application of the methodological foundations of information technology for processing gene expression data to solve problems in the field of bioinformatics. This is achieved through the comprehensive application of gene ontology analysis, cluster-bicluster analysis, and deep learning methods. The distinctive feature of this approach is its higher adequacy in assessing the state of an object compared to existing methods, achieved through the hybridization of existing methods and algorithms for processing big data, optimizing model hyperparameters using quantitative quality criteria at the relevant stage, and considering the type of data being studied.

The following results have been obtained:

1. An information technology for processing gene expression data has been proposed based on the comprehensive application of gene ontology analysis, cluster-bicluster analysis, and deep learning methods for diagnosing the disease based on gene expression data. A conceptual description and a flowchart of the step-by-step procedure for processing gene expression data have been presented, the application of which contributes to improving the accuracy and objectivity of decisions regarding the condition of the object being studied.

2. Methods for forming subsets of mutually expressed and significant gene expression profiles for further application in diagnostic systems based on gene expression data have been further developed. A technique for removing uninformative genes based on statistical criteria and Shannon entropy has been proposed, considering the degree of priority of the respective criterion. As a result, a fuzzy model for forming a subset of informative gene expression profiles has been developed, which was validated by applying a classifier to objects

that contain gene expression values in the formed subsets as attributes.

3. A hybrid model for forming subsets of gene expression profiles has been improved through the comprehensive application of Harrington's desirability function and the object classification method to evaluate the model's adequacy. This approach allows for the objective division of gene expression profiles into subsets based on statistical and entropy criteria. The proposed model was tested using gene expression data from objects studied for lung cancer. The adequacy of the model was demonstrated by assessing the quality criteria values for the classification of the studied objects.

4. A hybrid model for forming subsets of significant genes has been developed and practically implemented based on the comprehensive application of cluster-bicluster analysis and gene ontology analysis. This approach has improved the accuracy and objectivity of diagnosing the condition of complex objects based on gene expression data by enabling more precise formation of subsets of significant and mutually expressed genes, parallelizing the information processing, and reasonably tuning the hyperparameters of the models used at appropriate stages of information processing.

5. The methods for applying convolutional neural networks (CNNs) for sample classification based on gene expression data have been further developed by more thoroughly determining the network hyperparameters using grid search and Bayesian optimization algorithms. Various architectures of one-dimensional CNNs have been considered. As optimization hyperparameters, the activation functions of convolutional and dense layers, the number of filters, the kernel size of neurons in the convolutional and dense layers, and maximum pooling were investigated. The criteria used to evaluate the quality of the corresponding model included sample classification accuracy (Accuracy), the loss function value calculated on a subset of data for model validation, and the F1-score, which incorporates Type I and Type II errors (sensitivity and specificity) and is an effective criterion for the quality of sample classification into separate classes. An integrated F1-score criterion was proposed, the calculation of which involves applying Harrington's desirability function to the partial F1-score values calculated for individual classes.

6. The methods for applying recurrent neural networks (RNNs) for processing gene expression data have been further developed. Two types of RNNs were investigated: LSTM and GRU. An algorithm for optimizing the architecture and hyperparameter values of RNNs was proposed, and a comparative analysis of optimization methods based on grid search and Bayesian optimization was conducted. A comprehensive quality criterion for data classification using

the respective type of deep learning network was proposed, calculated as a weighted sum of partial quality criteria determined during the modelling process. Modelling of various RNN architectures was performed, resulting in the determination of optimal hyperparameter values for each type of network.

7. A comparative analysis of various types and architectures of both convolutional and recurrent neural networks, including hybrid models based on the comprehensive application of convolutional and recurrent networks, was utilized for the classification of samples based on gene expression data. The optimal hyperparameters for each model were determined using Bayesian optimization algorithms. It was shown that the highest efficiency in gene expression data classification, according to a group of quality criteria, is demonstrated by models based on the application of the GRU recurrent neural network. The classification accuracy on the test data subset using a two-layer GRU network was 97.5%, while using a hybrid model based on the sequential application of a two-layer CNN and a two-layer GRU recurrent neural network was 97.1%, surpassing the results obtained with other types of models.

8. A hybrid model for classifying gene expression data based on deep and machine learning methods has been proposed and implemented, enhancing the objectivity of decision-making regarding identifying samples under study. The model is presented as a flowchart outlining a step-by-step information processing procedure. In the first stage, various deep learning models are applied in parallel to the set of gene expression data, forming intermediate decisions that are structured into a data table for further processing by a classifier at the second hierarchical level of the model. As the classifier at the final step of the model implementation, a decision tree algorithm (CART) was used, providing the object's final identification decision. Modelling was performed to apply different combinations and varying numbers of deep learning models at the first hierarchical level of the model implementation.

9. The methods of biclustering gene expression data have been further developed by more carefully forming the quality criteria for biclustering, which determine the bicluster structure created during the implementation of the corresponding biclustering algorithm. An internal quality criterion for biclustering based on the assessment of mutual information between the bicluster's rows and between its columns has been proposed. The process of evaluating the effectiveness of the proposed quality criterion was modelled using artificial biclusters. It was shown that the values of the classical biclustering quality criterion, based on assessing the mean squared distance between all pairs of rows and columns in the bicluster, and the values of the criterion based on the assessment of mutual

information change consistently. The extrema of these criteria correspond to perfect biclustering, indicating the adequacy of the proposed criterion.

10. A hybrid model for biclustering gene expression data has been developed based on the comprehensive application of the ensemble biclustering algorithm and Bayesian optimization algorithm. This model uses a distance assessment metric between the rows and columns of the bicluster based on the evaluation of mutual information, allowing the optimization of the biclustering algorithm's parameters through the correct application of the target objective function based on the proposed biclustering quality criterion.

11. The methods based on gene ontology analysis in models analyzing gene expression data have been further developed. A two-step procedure for applying this method has been proposed, enhancing objectivity in forming subsets of significant and mutually expressed genes for their further use in diagnostic systems for the objects under study. A hybrid gene expression data identification model has been developed, combining gene ontology analysis, cluster-bicluster analysis, and a convolutional neural network. The model's effectiveness was evaluated, demonstrating the advantage of the proposed model by increasing the accuracy of object state identification with a smaller number of significant genes. This allows for more precise tuning of the diagnostic model for disease diagnosis based on gene expression data.

12. The results of the application of the proposed information technology for creating a disease diagnostic system are presented. Gene expression data from objects studied for Alzheimer's disease and various types of cancers were used as experimental data. The first type of data was obtained using DNA microarray experiments, while the second type of data was obtained using the RNA molecules sequencing method. The modelling results allowed us to conclude that the model's effectiveness is significantly higher when using gene expression data obtained through the RNA molecules sequencing method, which can be attributed to the higher quality of the data. Moreover, the high identification results of samples based on selected subsets of gene expression data indicate the high efficiency of the proposed information technology for forming subsets of significant and mutually correlated gene expression data for their further application in a disease diagnostic system.

Acknowledgements

The authors are grateful to professors Sharko A. and Lytvynenko V. for fruitful cooperation during formation of the book content. We would like also to thank the

reviewers prof. Gozhyj A., Mashkov V. and Hnatushenko V. for their remarks and comments, which contributed to the improvement of the book.

Bibliography

[1] Bioconductor: Open source software for Bioinformatics, https://bioconductor.org/

[2] Gene Expression Omnibus. *GEO* https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi (2006)

[3] The Cancer Genome Atlas Program (2023), https://www.cancer.gov/ccg/research/genome-sequencing/tcga

[4] Afreen, S., Bhurjee, A., Aziz, R.: Gene selection with Game Shapley Harris hawks optimizer for cancer classification," *Chemometrics and Intelligent Laboratory Systems*, journal = Chemometrics and Intelligent Laboratory Systems, volume = 242, year = 2023, number = 1, pages = art. no. 104989, doi = 10.1016/j.chemolab.2023.104989

[5] Alexa, A., Rahnenfuhrer, J.: topgo: Enrichment Analysis for Gene Ontology. R package version 2.54.0 (2023), https://bioconductor.org/packages/topGO

[6] Ali, E., Farinaz, R.: Systems biology approaches to identify driver genes and drug combinations for treating covid-19. Scientific Reports **14**, art. no. 2257 (2024). https://doi.org/10.1038/s41598-024-52484-8

[7] Amendolara, A., Sant, D., Rotstein, H., Fortune, E.: LSTM-based recurrent neural network provides effective short term flu forecasting. BMC Public Health **23**(1), art. no. 1788 (2023). https://doi.org/10.1186/s12889-023-16720-6

[8] Archer, E., Park, I., Pillow, J.: Bayesian entropy estimation for countable discrete distributions. Journal of Machine Learning Research **15**, 2833–2868 (2014). https://doi.org/10.48550/arXiv.1302.0328

[9] Ashburner, M., Ball, C., Blake, J.A., e.a.: Gene ontology: Tool for the unification of biology. the gene ontology consortium. Nature Genetics **25**(1), 25–34 (2000). https://doi.org/10.1038/75556

[10] Babichev, S., Barilla, J., Fišer, J., Škvor, J.: A hybrid model of gene expression profiles reducing based on the complex use of fuzzy inference system and clustering quality criteria. In: Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2019. pp. 128–133 (2020)

[11] Babichev, S., Durnyak, B., Pikh, I., Senkivskyy, V.: An evaluation of the objective clustering inductive technology effectiveness implemented using density-based and agglomerative hierarchical clustering algorithms. Advances in Intelligent Systems and Computing **1020**, 532–553 (2020). https://doi.org/10.1007/978-3-030-26474-1_37

[12] Babichev, S., Durnyak, B., Zhydetskyy, V., Pikh, I., Senkivskyy, V.: Application of optics density-based clustering algorithm using inductive methods of complex system analysis. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies. vol. 1, pp. 169—-172. Institute of Electrical and Electronics Engineers Inc. (2019). https://doi.org/10.1109/STC-CSIT.2018.8929869

[13] Babichev, S., Khamula, O., Durnyak, B., Škvor, J.: Technique of gene expression profiles selection based on sota clustering algorithm using statistical criteria and shannon entropy. Advances in Intelligent Systems and Computing **1246**, 23–38 (2021). https://doi.org/10.1007/978-3-030-54215-3_2

[14] Babichev, S., Korobchynskyi, M., Rudenko, M., Batenko, H.: Applying biclustering technique and gene ontology analysis for gene expression data processing. In: CEUR Workshop Proceedings. pp. 14–28 (2024)

[15] Babichev, S., Krejci, J., Bicanek, J., Lytvynenko, V.: Gene expression sequences clustering based on the internal and external clustering quality criteria. In: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017. vol. 1, pp. 91—-94. Institute of Electrical and Electronics Engineers Inc. (2017). https://doi.org/10.1109/STC-CSIT.2017.8098744

[16] Babichev, S., Liakh, I., Kalinina, I.: Applying a Recurrent Neural Network-Based Deep Learning Model for Gene Expression Data Classification. Applied Sciences **13**(21), art. no. 11823 (2023). https://doi.org/10.3390/app132111823

[17] Babichev, S., Liakh, I., Kalinina, I.: Applying the Deep Learning Techniques to Solve Classification Tasks Using Gene Expression Data. IEEE Access **12**, 28437–28448 (2024). https://doi.org/10.1109/ACCESS.2024.3368070

[18] Babichev, S., Liakh, I., Morokhovych, V., et al.: Applying convolutional neural network for cancer disease diagnosis based on gene expression data. CEUR Workshop Proceedings **3609**, 48–61 (2023)

[19] Babichev, S., Lytvynenko, V., Korobchynskyi, M., Taiff, M.: Objective clustering inductive technology of gene expression sequences features. Communications in Computer and Information Science **715**, 359–372 (2017). https://doi.org/10.1007/978-3-319-58274-0_29

[20] Babichev, S., Lytvynenko, V., Osypenko, V.: Implementation of the objective clustering inductive technology based on dbscan clustering algorithm. In: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017. vol. 1, pp. 479—-484. Institute of Electrical and Electronics Engineers Inc. (2017). https://doi.org/10.1109/STC-CSIT.2017.8098832

[21] Babichev, S., Lytvynenko, V., Skvor, J., Fiser, J.: Model of the objective clustering inductive technology of gene expression profiles based on sota and dbscan clustering algorithms. Advances in Intelligent Systems and Computing **689**, 21–39 (2018). https://doi.org/10.1007/978-3-319-70581-1_2

[22] Babichev, S., Lytvynenko, V., Škvor, J., et al.: Information technology of gene expression profiles processing for purpose of gene regulatory networks reconstruction. In: Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018. p. art. no. 8478452 (2018). https://doi.org/10.1109/DSMP.2018.8478452

[23] Babichev, S., Osypenko, V., Lytvynenko, V., Voronenko, M., Korobchynskyi, M.: Comparison Analysis of Biclustering Algorithms with the use of Artificial Data and Gene Expression Profiles. In: 2018 IEEE 38th International Conference on Electronics and Nanotechnology, ELNANO 2018 - Proceedings. pp. 298–304 (2018). https://doi.org/10.1109/ELNANO.2018.8477439

[24] Babichev, S., Osypenko, V., Lytvynenko, V., Voronenko, M., Korobchynskyi, M.: Comparison analysis of biclustering algorithms with the use of artificial data and gene expression profiles. In: (2018) 2018 IEEE 38th International Conference on Electronics and Nanotechnology, ELNANO 2018 - Proceedings. pp. 298–304 (2018). https://doi.org/10.1109/ELNANO.2018.8477439

[25] Babichev, S., Spivakovskiy, A., Škvor, J.: Comparison analysis of clustering quality criteria using inductive methods of objective clustering. Communications in Computer and Information Science **1158**, 150–166 (2020). https://doi.org/10.1007/978-3-030-61656-4_10

[26] Babichev, S., Yasinska-Damri, L., Liakh, I.: A hybrid model of cancer diseases diagnosis based on gene expression data with joint use of data mining methods and machine learning techniques. Applied Sciences **13**(10), art. no. 6022 (2023). https://doi.org/10.3390/app13106022

[27] Babichev, S., Yasinska-Damri, L., Liakh, I., Durnyak, B.: Comparison analysis of gene expression profiles proximity metrics. Symmetry **13**(10), art no. 1812 (2021). https://doi.org/10.3390/sym13101812

[28] Babichev, S., Yasinska-Damri, L., Liakh, I., Škvor, J.: Hybrid inductive model of differentially and co-expressed gene expression profile extraction based on the joint use of clustering technique and convolutional neural network. Applied Sciences (Switzerland) **12**(22), art no. 11795 (2022). https://doi.org/10.3390/app122211795

[29] Babichev, S., Yasinskyi, M., Yasinska-Damri, L., Ratushniak, Y., Lytvynenko, V.: Current state of the problem of gene expression data processing and extraction to solve the reverse engineering tasks in the field of bioinformatics. CEUR Workshop Proceedings **2853**, 62–71 (2021)

[30] Babichev, S., Škvor, J.: Technique of gene expression profiles extraction based on the complex use of clustering and classification methods. Diagnostics **10**(8), art. no. 584 (2020). https://doi.org/10.3390/diagnostics10080584

[31] Babichev, S., Gozhyj, A., Kornelyuk, A., Lytvynenko, V.: Objective clustering inductive technology of gene expression profiles based on sota clustering algorithm. Biopolymers and Cell **33**(5), 379–392 (2017). https://doi.org/10.7124/bc.000961

[32] Busaleh, M., Hussain, M., Aboalsamh, H.: Breast mass classification using diverse contextual information and convolutional neural network. Biosensors **11**(11), art. no. 419 (2021). https://doi.org/10.3390/bios11110419

[33] Cao, W., Ji, Z., Zhu, S., Wang, M., Sun, R.: Bioinformatic identification and experiment validation reveal 6 hub genes, promising diagnostic and therapeutic targets for Alzheimer's disease. BMC Medical Genomics **17**, art. no. 5 (2024). https://doi.org/10.1186/s12920-023-01775-6

[34] Carlson, M.: Go.db: A set of annotation maps describing the entire gene ontology. r package version 3.19.1 (2019), `https://bioconductor.org/packages/GO.db/`

[35] Carlson, M.: org.hs.eg.db: Genome wide annotation for Human. R package version 3.8.2. (2019), `https://bioconductor.org/packages/release/data/annotation/html/org.Hs.eg.db.html`

[36] Chuang, Y.H., Huang, S.H., et al.: Convolutional neural network for human cancer types prediction by integrating protein interaction networks and omics data. Scientific Reports **11**(1), art. no. 20691 (2021). https://doi.org/10.1038/s41598-021-98814-y

[37] Chuang, Y.H., Huang, S.H., Hung, T.M., et al.: Convolutional neural network for human cancer types prediction by integrating protein interaction networks and omics data. Scientific Reports **11**(1), art. no. 20691 (2021). https://doi.org/10.1038/s41598-021-98814-y

[38] Dopazo, J., Carazo, J.: Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. Journal of Molecular Evolution **44**, 226–233 (1997). https://doi.org/10.1007/PL00006139

[39] Durinck, S., Spellman, P., Birney, E., W., H.: Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. Nature Protocols **4**(8), 1184–1275 (2009). https://doi.org/10.1038/nprot.2009.97

[40] Dutta, S., Hore, M., Ahmad, F., et al.: Sbi-msreimpute: A sequential biclustering technique based on mean squared residue and euclidean distance to predict missing values in microarray gene expression data. Advances in Intelligent Systems and Computing **813**, 673–685 (2019). https://doi.org/10.1007/978-981-13-1498-8_59

[41] Fritzke, B.: Growing cell structures-A self-organizing network for unsupervised and supervised learning. Neural Networks **7**, 1441–1460 (1994). https://doi.org/10.1016/0893-6080(94)90091-4

[42] Gates, A., Ahn, Y.Y.: The impact of random models on clustering similarity. Journal of Machine Learning Research **18**, 1–28 (2017). https://doi.org/10.48550/arXiv.1701.06508

[43] Gholami, H., Mohammadifar, A., Golzari, S., et al.: Interpretability of simple RNN and GRU deep learning models used to map land susceptibility to gully erosion. Science of the Total Environment **904**, art. no. 166960 (2023). https://doi.org/10.1016/j.scitotenv.2023.166960

[44] Gupta, S., Gupta, M., Shabaz, A., Sharma, A.: Deep learning techniques for cancer classification using microarray gene expression data. Fronters in Physiology **13**, art. no. 952709 (2022). https://doi.org/10.3389/fphys.2022.952709

[45] Hasan, N., Mishra, A., A.K., R.: Fuzzy logic based cross-layer design to improve quality of service in mobile ad-hoc networks for next-gen cyber physical system. Engineering Science and Technology **35**, art. no. 101099 (2022). https://doi.org/10.1016/j.jestch.2022.101099

[46] Hausser, J., Strimmer, K.: Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. Journal of Machine Learning Research **10**, 1469–1484 (2009). https://doi.org/10.48550/arXiv.0811.3579

[47] Hou, J., Aerts, J., den Hamer, B., et al.: Gene expression-based classification of non-small cell lung carcinomas and survival prediction. PLoS ONE **5**, art. no. e10312 (2010). https://doi.org/10.1371/journal.pone.0010312

[48] Isabona, J., Imoize, A., Kim, Y.: Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning. Sensors **22**(10), art. no. 3776 (2016). https://doi.org/10.3390/s22103776

[49] Ivakhnenko, A.: Objective clustering based on the theory of self-organization models. Automatics **5**, 6–15 (1987)

[50] Iwański, M., Mazurek, G., Buczyński, P., Iwański, M.: Effects of hydraulic binder composition on the rheological characteristics of recycled mixtures with foamed bitumen for full depth reclamation. Construction and Building Materials **330**, 127274 (2022). https://doi.org/10.1016/j.conbuildmat.2022.127274

[51] Joshi, A.A. anf Aziz, R.: Deep learning approach for brain tumor classification using metaheuristic optimization with gene expression data. International Journal of Imaging Systems and Technology p. 10.1002/ima.23007 (2023). https://doi.org/10.1002/ima.23007

[52] Kaiser, S., Santamaria, R., Khamiakova, T., et al.: biclust: BiCluster Algorithms. *ensemble* `https://cran.r-project.org/package=biclust` (2023)

[53] Kaiser, S., Santamaria, R., Khamiakova, T., et al.: Biclust: Bicluster Algorithms (2023), `https://cran.r-project.org/web/packages/biclust/index.html`

[54] Karthika, M., Rajaguru, H., Nair, A.: Evaluation and Exploration of Machine Learning and Convolutional Neural Network Classifiers in Detection of Lung Cancer from Microarray Gene - A Paradigm Shift. Bioengineering **10**(8), art. no. 933 (2023). https://doi.org/10.3390/bioengineering10080933

[55] Kim, D., Kwon, K., Pham, K., et al.: Surface settlement prediction for urban tunneling using machine learning algorithms with Bayesian optimization. Automation in Construction **140**, art. no. 104331 (2022). https://doi.org/10.1016/j.autcon.2022.104331

[56] Li, J., Sun, W., Feng, X., et al.: A dense connection encoding–decoding convolutional neural network structure for semantic segmentation of thymoma. Neurocomputing **451**, 1–11 (2021). https://doi.org/10.1016/j.neucom.2021.04.023

[57] Li, X., Li, J., Li, J., Liu, N., Zhuang, L.: Development and validation of epigenetic modification-related signals for the diagnosis and prognosis of colorectal cancer. BMC Genomics **25**, art. no. 51 (2024). https://doi.org/10.1186/s12864-023-09815-2

[58] Liakh, I., Babichev, S., Durnyak, B., Gado, I.: Formation of subsets of co-expressed gene expression profiles based on joint use of fuzzy inference system, statistical criteria and shannon entropy. Lecture Notes on Data Engineering and Communications Technologies **149**, 25–41 (2023). https://doi.org/10.1007/978-3-031-16203-9_2

[59] Liang, W., Dunckley, T., Beach, T., et al.: Gene expression profiles in anatomically and functionally distinct regions of the normal aged human brain. Physiological Genomics **28**, 311–322 (2007). https://doi.org/10.1152/physiolgenomics.00208.2006

[60] Liang, W., Dunckley, T., Beach, T., et al.: Altered neuronal gene expression in brain regions differentially affected by Alzheimer's disease: A reference data set. Physiological Genomics **33**, 240–256 (2008). https://doi.org/10.1016/j.neuron.2018.05.02

[61] Liang, W., Reiman, E., Valla, J., et al.: Alzheimer's disease is associated with reduced expression of energy metabolism genes in posterior cingulate neurons. Proceedings of the National Academy of Sciences of the United States of America **105**, 4441–4446 (2008). https://doi.org/10.1073/pnas.0709259105

[62] Liu, J., Ge, S., Cheng, Y., Wang, X.: Multi-view spectral clustering based on multi-smooth representation fusion for cancer subtype prediction. Frontiers in Genetics **12**, art. no. 718915 (2021). https://doi.org/10.3389/fgene.2021.718915

[63] Luo, Y., Liu, L., Zhang, C.: Identification and analysis of diverse cell death patterns in diabetic kidney disease using microarray-based transcriptome profiling and single-nucleus rna sequencing. Computers in Biology and Medicine **169**, art. no. 107780 (2024). https://doi.org/10.1016/j.compbiomed.2023.107780

[64] Madala, H., Ivakhnenko, A.: Inductive Learning Algorithms for Complex Systems Modeling, chap. 5:Clusterization and Recognition, p. 380. CRC Press (1994)

[65] Mahto, R., Ahmed, S., Rahman, R., et al.: A novel and innovative cancer classification framework through a consecutive utilization of hybrid feature selection. BMC Bioinformatics **10**(1), art. no. 479 (2023). https://doi.org/10.1186/s12859-023-05605-5

[66] Midi, H., Aziz, N.: Augmented desirability function for multiple responses with contaminated data. Journal of Engineering and Applied Sciences **13**(16), 16629–6633 (2018). https://doi.org/10.36478/jeasci.2018.6626.6633

[67] Morandat, F., Hill, B., Osvald, L., Vitek, J.: Evaluating the design of the r language. Lecture Notes in Computer Science **7313**, 104–131 (2012). https://doi.org/10.1007/978-3-642-31057-7_6

[68] Mostavi, M., Chiu, Y.C., et al.: Convolutional neural network models for cancer type prediction based on gene expression. BMC Medical Genomics **13**, art. no. 44 (2020). https://doi.org/10.1186/s12920-020-0677-2

[69] Nikolados, E.M., Oyarzún, D.: Deep learning for optimization of protein expression. Current Opinion in Biotechnology **81**, art. no. 102941 (2023). https://doi.org/10.1016/j.copbio.2023.102941

[70] Paninski, L.: Estimation of entropy and mutual information. Neural Computation **15**, 1191–253 (2003). https://doi.org/10.1162/089976603321780272

[71] Ramirez, R., Chiu, Y.C., et al.: Classification of Cancer Types Using Graph Convolutional Neural Networks. Scientific Reports **8**, art. no. 203 (2020). https://doi.org/10.3389/fphy.2020.00203

[72] Readhead, B., Haure-Mirande, J.V., Funk, C., et al.: Multiscale Analysis of Independent Alzheimer's Cohorts Finds Disruption of Molecular, Genetic, and Clinical Networks by Human Herpesvirus. Neuron **99**, 64–82 (2018). https://doi.org/10.1016/j.neuron.2018.05.023

[73] Romero, M., Ramírez, O., Finke, J., C., R.: Supervised gene function prediction using spectral clustering on gene co-expression networks. Studies in Computational Intelligence **1016**, 652–663 (2022). https://doi.org/10.1007/978-3-030-93413-2_54

[74] Srikantamurthy, M., Rallabandi, V., Dudekula, D., Natarajan, S., Park, J.: Classification of benign and malignant subtypes of breast cancer histopathology imaging using hybrid CNN-LSTM based transfer learning. BMC Medical Imaging **23**(1), art. no. 19 (2023). https://doi.org/10.1186/s12880-023-00964-0

[75] Sun, X., Guo, P.and Wang, N., Shi, Y., Li, Y.: A refined therapeutic plan based on the machine-learning prognostic model of liver hepatocellular carcinoma. Computers in Biology and Medicine **169**, art. no. 107907 (2024). https://doi.org/10.1016/j.compbiomed.2023.107907

[76] Taherkhani, N., Sepehri, M., Khasha, R., Shafaghi, S.: Ranking patients on the kidney transplant waiting list based on fuzzy inference system. BMC Nephrology **23**(1), art. no. 31 (2022). https://doi.org/10.1186/s12882-022-02662-5

[77] Tomas, P., Ebert, D., Muruganujan, A., et al.: Panther: Making genome-scale phylogenetics accessible to all. Protein Science **31**(1), 8–22 (2021). https://doi.org/10.1002/pro.4218

[78] Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**, 395–416 (2007). https://doi.org/10.48550/arXiv.0711.0189

[79] Wu, T., Hu, E., Xu, S., et al.: clusterprofiler 4.0: A universal enrichment tool for interpreting omics data. Innovation **2**(3), art. no. 100141 (2021). https://doi.org/10.1016/j.xinn.2021.100141

[80] Yasinska-Damri, L., Babichev, S., Durnyak, B., Goncharenko, T.: Application of convolutional neural network for gene expression data classification. In: Babichev, S., Lytvynenko, V. (eds.) Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making. pp. 3–24. Springer International Publishing (2023). https://doi.org/10.1007/978-3-031-16203-9_1

[81] Yasinska-Damri, L., Babichev, S., Liakh, I.: Comparison analysis of the pearson's phi-square test and correlation metric effectiveness to form the subset of differently expressed and mutually correlated genes. CEUR Workshop Proceedings **3150**, 93–102 (2022)

[82] Yasinska-Damri, L., Liakh, I., Babichev, S., Durnyak, B.: Evaluation of the gene expression profiles complex proximity metric effectiveness based on a hybrid technique of gene expression data extraction. CEUR Workshop Proceedings **3038**, 150–160 (2021)

[83] Yasinska-Damri, L., Liakh, I., Babichev, S., Durnyak, B.: Current state of methods, models, and information technologies of genes expression profiling

extraction: A review. Lecture Notes on Data Engineering and Communications Technologies **77**, 69–81 (2022). https://doi.org/10.1007/978-3-030-82014-5_5

[84] Yin, L., Qiu, J., Gao, S.: Biclustering of gene expression data using cuckoo search and genetic algorithm. International Journal of Pattern Recognition and Artificial Intelligence **32**(11), art. no. 1850039 (2018). https://doi.org/10.1142/S0218001418500398

[85] Yu, G., Wang, L.G., Han, Y., He, Q.Y.: ClusterProfiler: An R package for comparing biological themes among gene clusters. OMICS A Journal of Integrative Biology **16**(5), 284–291 (2012). https://doi.org/10.1089/omi.2011.0118

[86] Yu, K., Xie, W., Wang, L., Zhang, S., Li, W.: Determination of biomarkers from microarray data using graph neural network and spectral clustering. Scientific Reports **11**, art. no. 23828 (2021). https://doi.org/10.1038/s41598-021-03316-6

[87] Yuan, X., Liebelt, M., Shi, P., Phillips, B.: Cognitive decisions based on a rule-based fuzzy system. Information Sciences **600**, 323–341 (2022). https://doi.org/10.1016/j.ins.2022.03.089

[88] Zadeh, L.: Fuzzy logic. Computational Complexity: Theory, Techniques, and Applications pp. 1177–1200 (2013). https://doi.org/10.1007/978-1-4614-1800-9_73

[89] Zan, T., Wang, H., Wang, M., Liu, Z., Gao, X.: Application of Multi-Dimension Input Convolutional Neural Network in Fault Diagnosis of Rolling Bearings. Applied Sciences **9**(13), art. no. 2690 (2019). https://doi.org/10.3390/app9132690

[90] Zhao, Y., Chen, Z., Dong, Y., Tu, J.: An interpretable LSTM deep learning model predicts the time-dependent swelling behavior in CERCER composite fuels. Materials Today Communications **37**, art. no. 106998 (2023). https://doi.org/10.1016/j.mtcomm.2023.106998

For notes

Scientific edition

Sergii Babichev, Ihor Liakh and Bohdan Durnyak

Application of Data Mining and Machine Learning Methods to Develop a Disease Diagnosis System Based on Gene Expression Data

Collective Monograph

State Enterprise All-Ukrainian Specialized Publishing House "Svit"
21 Halytzka St. Lviv 79008 Ukraine
tel.: + 38 (032) 235-6525

Signed for printing **17.01.2025**
Format 70 × 100/16. Offset paper. Offset printing technique.
Print run **22.01.2025**. Order No. **500**.

Printed in SE All-Ukrainian Specialized
Publishing House "Svit"
21 Halytzka St. Lviv 79008 Ukraine