

Article

Applying a Recurrent Neural Network-Based Deep Learning Model for Gene Expression Data Classification

Sergii Babichev ^{1,2,*}, Igor Liakh ^{3,t} and Irina Kalinina ^{4,t}

¹ Department of Informatics, Jan Evangelista Purkyně University in Ústí nad Labem, Pasteurova 3632/15, 400 96 Ústí nad Labem, Czech Republic

² Department of Physics, Kherson State University, 73008 Kherson, Ukraine

³ Department of Information Science and Physics and Mathematics Disciplines, Uzhhorod National University, 88000 Uzhhorod, Ukraine; ihor.lyah@uzhnu.edu.ua

⁴ Department of Intelligent Information Systems, Petro Mohyla Black Sea National University, 54000 Mykolaiv, Ukraine; irina.kalinina1612@gmail.com

* Correspondence: sergii.babichev@ujep.cz or sbabichev@ksu.ks.ua; Tel.: +420-777-843-785

[†] The authors contributed to this work as follows: the first author—60%, the second and the third authors—20%.

Abstract: The importance of gene expression data processing in solving the classification task is determined by its ability to discern intricate patterns and relationships within genetic information, enabling the precise categorization and understanding of various gene expression profiles and their consequential impacts on biological processes and traits. In this study, we investigated various architectures and types of recurrent neural networks focusing on gene expression data. The effectiveness of the appropriate model was evaluated using various classification quality criteria based on type 1 and type 2 errors. Moreover, we calculated the integrated F1-score index using the Harrington desirability method, the value of which allowed us to improve the objectivity of the decision making when model effectiveness was evaluated. The final decision regarding model effectiveness was made based on a comprehensive classification quality criterion, which was calculated as the weighted sum of classification accuracy, integrated F1-score index, and loss function values. The simulation results show higher appeal of a single-layer GRU recurrent network with 75 neurons in the recurrent layer. We also compared convolutional and recurrent neural networks on gene expression data classification. Although convolutional neural networks showcase benefits in terms of loss function value and training time, a comparative analysis revealed 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, 954 out of 981 objects were correctly identified. In other cases, 952 objects were correctly identified.

Keywords: gene expression profiles; recurrent neural network; deep learning; classification quality criteria; grid search algorithm; cancer disease



Citation: Babichev, S.; Liakh, I.; Kalinina, I. Applying a Recurrent Neural Network-Based Deep Learning Model for Gene Expression Data Classification. *Appl. Sci.* **2023**, *13*, 11823. <https://doi.org/10.3390/app132111823>

Academic Editors: Jeong Seop Sim and SooJun Park

Received: 9 October 2023

Revised: 25 October 2023

Accepted: 26 October 2023

Published: 29 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The application of deep learning methods to processing gene expression data in order to tackle classification tasks has taken precedence due to several compelling reasons. Primarily, the complexity and high dimensionality of gene expression data necessitate sophisticated computational models, which deep learning algorithms aptly provide, offering a robust mechanism to decipher intricate patterns within the data. The multi-layered architectures of deep learning models, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have proven highly proficient in capturing non-linear relationships in gene expression, which is crucial for accurate classification [1]. Further,

deep learning models can autonomously learn feature representations from raw data, eliminating the need for manual feature extraction, thereby reducing potential biases introduced during this process. Additionally, the models, when adequately trained, demonstrate an impressive ability to minimize error and optimize classification tasks, which is vital to ensuring reliable predictive performance. The applicability of deep learning models, such as autoencoders and variational autoencoders, also facilitates effective dimensionality reduction in gene expression data, aiding in mitigating the curse of dimensionality [2]. Moreover, in an era where precision medicine is pivotal, the application of deep learning models to classifying gene expression data is paramount to identifying specific disease subtypes and enabling targeted therapeutic approaches. The scalability of deep learning models also affords them the flexibility to handle large genomic datasets, which is often a requisite in bioinformatics research. Furthermore, the adaptive nature of deep learning models allows them to evolve with increasing data and complexity, ensuring that they remain relevant and applicable in the dynamically progressing field of genomics. Lastly, the amalgamation of deep learning with existing biological knowledge can potentially unearth novel insights and drive advancements in understanding gene expression mechanisms and their implications in various biological phenomena and diseases. These facts indicate the actuality of the research in this subject area.

Gene expression data processing in molecular biology uses both experimental and computational methods to analyze gene activity. The widely used qRT-PCR method quantifies specific gene expression levels and is excellent for validating high-throughput results [3]. Microarrays, which analyze thousands of genes simultaneously, have long been standard but are gradually being surpassed by RNA sequencing (RNA-Seq) due to its higher resolution and capability to detect novel transcripts. Drop-seq, a newer method, combines RNA-seq with microfluidics to study individual-cell gene expression [4]. After experimental data collection, bioinformatics tools are essential for tasks like normalization and differential expression analysis. DESeq2 and edgeR are examples of popular software for such tasks.

CNNs have emerged as a pivotal tool for classifying gene expression data, thanks to their autonomous feature-learning capability, which curtails the need for manual intervention in high-dimensional genomic data extraction [5–7]. By recognizing patterns effectively, they capture both local and global spatial hierarchies of gene expression profiles, a key aspect in identifying complex biological states. The inherent mechanism of dimensionality reduction through pooling layers in CNNs substantially alleviates computational challenges, proving particularly instrumental when dealing with voluminous gene expression datasets. Moreover, CNNs maintain and interpret spatial relationships within the data, which is essential to understanding the spatial configuration of genomic sequences. Importantly, CNNs facilitate transfer learning, enabling the adaptation of pre-trained models to specific gene expression datasets, a vital attribute in scenarios with limited labeled data. They exhibit commendable robustness against data noise, ensuring predictive accuracy despite potential experimental variations in gene expression data. CNNs, while complex, can be rendered interpretable using specific visualization techniques, revealing insightful features and enhancing model transparency. Ultimately, they recognize features at varied scales and ensure effective generalization through strategic model design, adeptly applying learned patterns to novel, unseen gene expression data, thereby holding substantial promise for practical applications in both biomedical research and diagnostics.

However, it is important to highlight that a significant drawback of CNNs, impeding their efficient utilization, is the substantial set of hyperparameters, the optimal values that define the network's architecture. Typically, values for these hyperparameters are determined using either a grid search algorithm or the Bayesian optimization method [8,9]. Nevertheless, this process demands notable expenditure of time and computational resources. Recurrent neural networks (RNNs) have specific advantages over convolutional neural networks (CNNs) when it comes to processing gene expression data, especially concerning sequence prediction and temporal dynamics understanding [10]. Firstly, RNNs are

inherently adept at handling sequential data, which is crucial to analyzing gene expression time series where temporal dependencies exist. Secondly, they are capable of maintaining memory from previous inputs in their internal state, enabling them to capture dynamic temporal behaviors in gene expression profiles, unlike CNNs, which typically treat input features as independent. Thirdly, the gated units in advanced RNNs, like LSTM and GRU, can selectively remember or forget information, which is essential for learning long-term dependencies observed in certain biological processes. Fourthly, RNNs can model sequential data of varied lengths without the necessity to fix the input size, providing flexibility in dealing with gene expression profiles from varied experimental setups. Fifthly, they naturally support bi-directional input processing, which can be beneficial for exploring gene expression profiles that may have bi-directional influences. Lastly, RNNs may facilitate more intuitive insights into gene expression pathways and cascades due to their sequential processing nature, potentially offering more biologically relevant interpretations compared with the hierarchical feature learning in CNNs. However, it is noteworthy that RNNs also come with challenges, such as susceptibility to vanishing and exploding gradient problems in training, which need careful consideration during model development and application in the bioinformatics domain.

In this study, we extend our previous research on the application of deep learning methods for gene expression data processing, aiming to develop and enhance cancer disease diagnosis systems [11–13]. The main contributions of the current research are as follows:

- Various architectures and types of recurrent neural networks (RNNs), namely, LSTM and GRU, with primary focus on their capacity to process gene expression data effectively, are investigated.
- We introduce an algorithm that optimizes the architecture and hyperparameter values of RNNs, considering both classification accuracy and F1-score, thereby enabling a thorough assessment of sample distribution quality across classes.
- We propose an integrated F1-score index calculated using the Harrington desirability method and a comprehensive classification quality criterion, formulated as the weighted sum of multiple partial quality criteria, like classification accuracy, integrated F1-score index, and loss function values, enhancing objectivity and depth in evaluating model effectiveness.
- We perform a comparison between convolutional neural networks (CNNs) and various RNN architectures, with an evaluation of their effectiveness based on the calculation of classification accuracy, integrated F1-score index, loss function values, and training times.

2. Related Works

Currently, many studies are devoted to solving the problem of gene expression processing using deep learning methods. Thus, the paper [14] explores the application of deep learning methods to the intricate task of understanding gene expression processes, which traditionally leans on high-throughput sequencing technologies to classify and identify transcription factors accurately. Considering the shortcomings of both traditional technologies and existing bioinformatic models, which tend to become bogged down by complex analysis function modules and burgeoning parameter counts, the authors propose a novel approach, DeepCAC, utilizing deep convolutional neural networks and a multi-head self-attention mechanism to adeptly capture local and long-distance hidden features in DNA transcription factor sequences while judiciously curtailing the number of parameters. However, it should be noted that while DeepCAC signifies a promising advance, demonstrating augmented performance and parameter efficiency, its limitations—particularly its unverified reliability across a range of biological contexts and potential susceptibility to sparse datasets—necessitate exhaustive validation in broader applications to substantiate its efficacy in deep learning applications to gene expression processing. The exploration [15] focuses on employing deep learning methods to decipher the complex genetic underpinnings of late-onset Alzheimer’s disease (LOAD), the predominant multi-

factorial neurodegenerative condition afflicting the elderly, utilizing Japanese genome-wide association study (GWAS) data to identify two disparate patient groups, each characterized by distinct genetic markers related to major risk and immune-related genes or kidney disorder-associated genes, hence suggesting a potential interplay between impaired kidney function and LOAD pathogenesis. A predictive model for LOAD subtypes was constructed using a deep neural network, yielding accuracy values of 0.694 and 0.687 in the discovery and validation cohorts, respectively, presenting a notable advancement in the nuanced categorization of LOAD subtypes. However, while offering new insights into LOAD pathogenic mechanisms, it is crucial to recognize the model's limitations, with the model necessitating further optimization and rigorous validation across larger and more diverse population samples to confirm these findings' broader applicability and robustness.

Many recent studies have explored the potential of convolutional neural networks (CNNs) for classifying extensive datasets [8,9,16,17]. For instance, the research study in [8] transformed gene expression datasets from 11 cancer types into 2D images using spectral clustering and achieved a classification accuracy ranging between 97.7% and 98.4% using CNNs. Another work, [16], utilized both support vector machines and CNNs to detect early breast cancer signs, outperforming existing classification methods for benign and malignant mass regions. This approach could aid radiologists in improving their diagnostic accuracy. The technique of a dense skip connection encoding–decoding structure based on CNNs is discussed in [9], where an image preprocessing method amplified the contrast between thymoma and surrounding tissues, improving the classification accuracy by 4% compared with other methods. The study in [17] introduced a Mixed Skin Lesion Picture Generate method using Mask R-CNN to address data imbalance issues, achieving 90.61% accuracy and 78.00% sensitivity on the ISIC dataset. Further works, [18,19], affirmed the efficiency of CNNs in classifying objects with large attribute counts based on gene expression data.

The study [20] considers the application of deep learning methods to navigate the challenging landscape of breast cancer subtyping, with its intrinsic heterogeneity and consequent varied prognostic outcomes, by introducing moBRCA-net, an interpretable deep learning-based classification framework that harnesses multi-omics datasets—specifically, by integrating gene expression, DNA methylation, and microRNA expression data while respecting the biological interrelationships among them. By employing a self-attention module for each omics dataset to ascertain the relative importance of each feature and subsequently transforming these features into new representations based on learned importance, moBRCA-net aims to adeptly predict breast cancer subtypes, demonstrating notably enhanced performance and effective multi-omics integration in comparison with other methods, as substantiated by experimental results. Despite the promising advancements brought forth by moBRCA-net, the potential limitations of this approach, such as the need for comprehensive validation across varied demographic cohorts and the adaptability to accommodate evolving omics data types, warrant careful consideration and further exploration to ensure its applicability and reliability in real-world clinical settings. In [21], the authors proposed employing machine learning and deep learning methodologies, specifically utilizing an autoencoder neural network and various statistical models, to identify prognostic biomarkers predictive of time to development and survival stratification in oral cancer (OC) by analyzing the gene expression profiles of 86 patients from the GSE26549 dataset. The approaches used allowed for the extraction of 100 encoded features, of which 70 were found to be significantly related to time to OC development, and further analyses identified two survival risk groups and 21 top genes, demonstrating the overall random forest classifier accuracy of 0.916 over the test set, thus potentially illuminating transcriptional biomarkers pertinent to determining high-risk OC patients and offering promising therapeutic targets. However, despite the insightful findings, the study bears limitations, such as the restricted patient sample size and the need for thorough validation across diverse demographic cohorts and varied types of OC, ensuring the identified biomarkers' broader applicability and efficacy in prospective, retrospective, and real-world clinical settings.

In [22], the authors considered the application of deep learning methods, such as convolutional neural network (CNN), deep neural network (DNN), and Long Short-Term Memory (LSTM) recurrent neural network (RNN), to address the critical need for efficient and rapid computational models for breast cancer prognosis, thereby proposing an ensemble model for breast cancer survivability prediction (EBCSP) that harnesses multi-modal data and stacks the outputs of various neural networks dedicated to distinct data modalities. By employing a CNN for clinical modalities, a DNN for copy number variations (CNV), and an LSTM-RNN for gene expression modalities and subsequently utilizing the individual models' outputs for binary classification based on survivability using the random forest method, the EBCSP model successfully outperforms models reliant on a single data modality for prediction, as well as existing benchmarks. Despite its promising results, it is imperative to acknowledge inherent limitations, such as the necessity for comprehensive validation across diverse demographic and clinical cohorts, as well as the evaluation of model robustness and accuracy in real-world practical medical settings, to ascertain its efficacy and applicability.

The studies [23–25] delve into the meticulous application of recurrent neural networks (RNNs) for diverse approaches to understanding and classifying gene regulatory networks (GRNs) and cancer detection from gene expression data. One approach [23] leverages a dual-attention RNN to not only predict gene temporal dynamics with high accuracy across various GRN architectures but also exploit the attention mechanism of RNNs, employing graph theory tools to hierarchically distinguish different architectures of the GRN, though the robustness of this method against varied noise types and its applicability to non-synthetic data present potential limitations. Another research study [24] introduces a strategy for cancer classification using a JayaAnt lion optimization-based Deep RNN (JayaALO-based DeepRNN), involving data normalization, transformation, feature dimension detection, and classification, and while achieving high classification accuracy, sensitivity, and specificity, the method's generalized applicability and performance consistency across different types and stages of cancers are yet to be fully explored. In a similar vein, the third study [25] proposes a Rider Chicken Optimization algorithm-dependent RNN (RCO-RNN) classifier for cancer detection and classification, which, despite demonstrating promising results across several datasets, still demands a thorough investigation regarding its performance on varied genomic profiles and under possible computational constraints.

However, we would like to note that the application of deep learning methods to gene expression data frequently encounters the notable challenge of optimizing hyperparameters, such as learning rate, batch size, and network architecture, which is critical to model performance but is often performed with computationally expensive and time-consuming trial-and-error or grid search processes. Additionally, deep learning models can struggle with the high dimensionality and often sparse nature of gene expression data, requiring robust data preprocessing and feature selection to prevent overfitting and ensure generalizable, biologically relevant predictions. For this reason, the study in this subject area is relevant.

3. Experimental Dataset

The research simulation leveraged gene expression data from patients evaluated for various cancer types, available through The Cancer Genome Atlas (TCGA) [26]. The data, obtained from the Illumina platform [27] through RNA genomic sequencing, initially included 3269 samples and 19,947 genes, with each sample pinpointing numerous relevant genes defining its state, and Table 1 provides a detailed breakdown of the experimental data characteristics, categorizing disease type, corresponding sample numbers, and counts of non-cancerous patient samples.

The gene expression value of a sample, as outlined in Table 1, signifies its activity level, indicative of the intensity of the associated protein synthesis process, and is proportionate to the volume of similar genes. In compliance with the methodology detailed in [11,13],

firstly, absolute gene counts were converted into a more conducive range (Count Per Million—CPM) using the subsequent formula:

$$CPM_{ij} = \frac{count_{ij}}{\sum_{j=1}^m count_{ij}} \cdot 10^6 \quad (1)$$

where $count_{ij}$ denotes the number of the j th type of gene related to the i th sample and m indicates the total number of unique gene types explored in the experiment.

Table 1. The 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	224

The application of this step notably diminished the range of variation in the absolute values defining each gene's expression (activity level). In the second step, data normalization was performed by applying the function $\log_2(CPM)$ to all values. Next, non-expressed genes were omitted based on the condition $\log_2(CPM) \leq 0$ across all analyzed samples, decreasing the gene count by 682 and formatting the gene expression experimental data matrix to $E = (3269 \times 19,265)$.

4. Criteria for Evaluating Sample Classification Quality

In the present study, object classification was carried out by leveraging metrics that evaluate type 1 and type 2 errors [28]:

- Classification accuracy is the overall percentage of correctly identified samples, calculated as

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

- F1-score is a metric that assesses the precision and recall of sample distribution across the relevant classes, computed as

$$F1 = \frac{2 \cdot PR \cdot RC}{PR + RC} \quad (3)$$

where precision (PR), the probability of correct sample identification for a class, is

$$PR = \frac{TP}{TP + FP} \quad (4)$$

and recall (RC), the probability of correctly identifying true positive cases for a class, is

$$RC = \frac{TP}{TP + FN} \quad (5)$$

In these formulas, TP (true positive) and TN (true negative) indicate the count of objects correctly classified, while FP (false positive) and FN (false negative) represent those classified incorrectly. It is crucial to highlight that for a multi-class issue, the

accuracy gauges the overall sample distribution accuracy among classes, while the *F1-score* appraises the precision of sample distribution within each class independently.

- Cross-entropy loss function (*Loss*)—computed during model validation to measure the disparity between predicted and actual probability distributions, in a multi-class classification task involving *C* classes:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij}) \tag{6}$$

where y_{ij} denotes a binary indicator of whether class j is the correct classification for observation i , \hat{y}_{ij} is the predicted probability of observation i being of class j , and N represents the total number of observations.

Given the challenges of analyzing *F1-score* values corresponding to multiple classes to select the optimal alternative from a hyperparameter list, an integrated *F1-score* value was computed. This calculation utilized Harrington’s desirability method [29], a potent method for addressing multicriterion issues. The implementation algorithm of this procedure encompasses the following stages:

1. Initialization. Transform the *F1-score* values in a matrix format, with rows representing classes and columns representing the hyperparameter values under examination in this phase.
2. Private desirability calculation. Identify the minimum and maximum values of the *F1-score* during the relevant neural network model operation phase (using the respective hyperparameter combination). Subsequently, transform the scale of *F1-score* values into a linear scale of the dimensionless indicator Y , considering the boundary values of the *F1-score* identified in the prior step (the value of parameter Y , according to the desirability method, varies from $Y_{min} = -2$ to $Y_{max} = 5$):

$$\begin{aligned} Y_{min} &= a + b \cdot F1_{min} \\ Y_{max} &= a + b \cdot F1_{max} \end{aligned} \tag{7}$$

Transform the *F1-score* values into Y values:

$$Y = a + b \cdot F1 \tag{8}$$

Compute private desirability for each *F1-score* value:

$$d = \exp(-\exp(-Y)) \tag{9}$$

3. Integrated *F1-score* value calculation. For each column of the matrix obtained in step 2, compute the integrated *F1-score* value as the geometric mean of all private desirability functions:

$$F1_{int}^j = \left(\prod_{i=1}^9 d_{ij} \right)^{\frac{1}{9}} \tag{10}$$

4. Results analysis: Generate a diagram illustrating the dependency of the integrated *F1-score* values on the respective combination of hyperparameter values. Select the optimal combination of hyperparameter values that correlates with the maximum integrated *F1-score*.

5. 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 [10,30,31].

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 in each data processing step. It is noteworthy that a distinctive feature of RNNs is that the hidden state in step t encompasses information from both the preceding step $t-1$ and the current input signal. The recurrent layer applies an appropriate activation function to transform the combined input in 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:

$$\begin{aligned} h(t) &= f_{act}^h(w_{ih}x(t) + w_{hh}f(t-1)) \\ y(t) &= f_{act}^0(w_{h0}f(t)) \end{aligned} \quad (11)$$

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; w_{ih} , w_{hh} , and w_{h0} are the weight coefficient matrices for the input layer to the first hidden layer, among hidden layers, and the last hidden layer to the output layer, respectively; $h(t-1)$ and $h(t)$ are the output values of the neurons in the hidden layers in 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 with 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 using grid search. The primary hyperparameters that determine the performance efficiency of RNNs include the following:

- The number of recurrent (hidden) layers.
- The number of neurons in the recurrent layers.
- Activation functions for the recurrent and output layers. Typically, as with the previous type of network mentioned, the *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 the *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 the number of rows or samples under investigation; m is the number of genes, the expression values of which determine the state of the respective samples.
- 1.2. Forming 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); $k = 30, \dots, 80$, $dk = 5$ (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 one, E_{test} , for testing.
- 1.4. Further splitting training subset E_{train} into two subsets in a 0.8/0.2 ratio, where the first subset, E'_{train} , is directly used for training and the second one, E_{valid} , for model validation during training. Ensuring that 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 model training.

Stage II. Algorithm operation within the hyperparameter 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 $k < k_{max}$, increasing the number of neurons in the recurrent layers by 5 ($k = k + 5$), and escaping to step 2.3 of this procedure. Otherwise, calculating the integrated F1-score value; analyzing the obtained results; and forming the optimal decision regarding the number of neurons in the recurrent layers in this stage.
- 2.6. If the number of recurrent layers is less than the maximum number ($layers < layers_{max}$), increasing the number of layers by 1, and going to step 2.2 of this algorithm. Otherwise, proceeding to Stage III.

Stage III. Analysis of the obtained results and formulation of 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.

5.1. Modeling of LSTM Recurrent Neural Network

In Figures 1–3, the simulation results to determine the optimal hyperparameters of the LSTM recurrent neural network are depicted.

Single-layer, two-layer, and three-layer neural networks were investigated during the simulation process. As the simulation results show, 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.

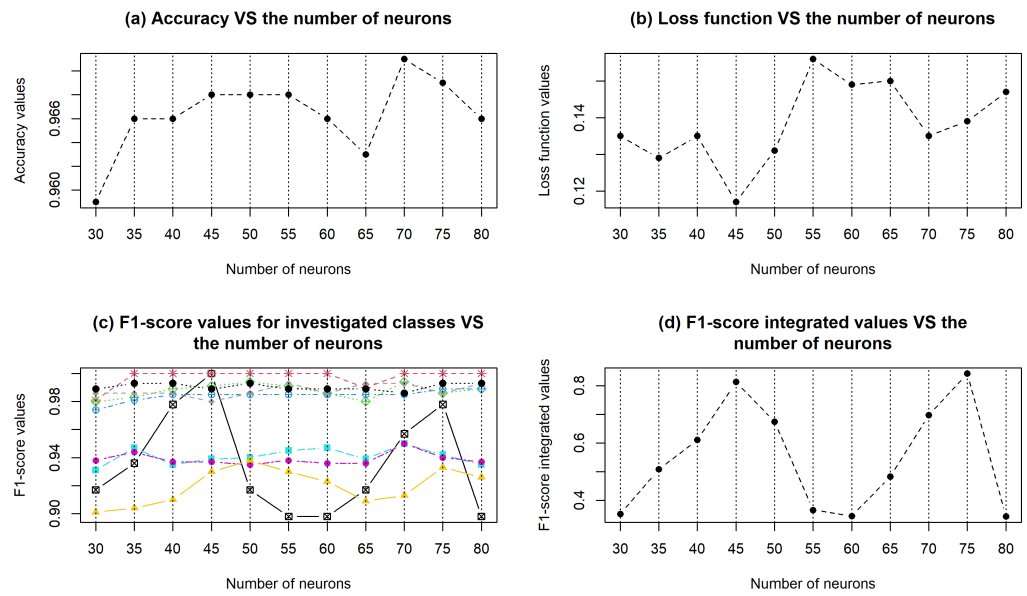


Figure 1. Results of modeling when applying a single-layer LSTM recurrent neural network. The different colors in (c) correspond to different classes.

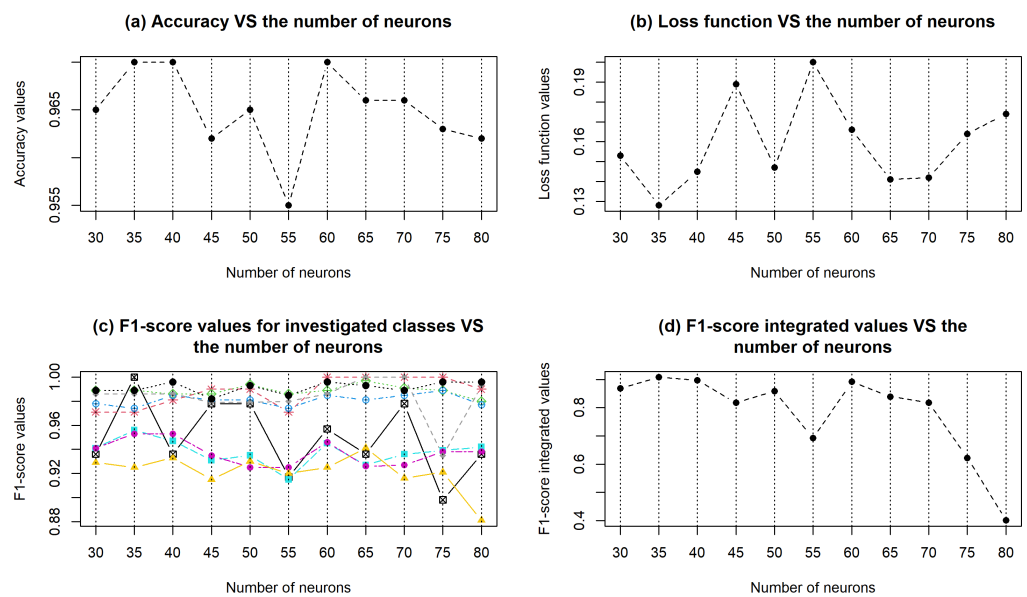


Figure 2. Results of modeling when applying a two-layer LSTM recurrent neural network. The different colors in (c) correspond to different classes.

Analyzing the modeling results allows us to conclude that in all cases, the accuracy of classifying samples comprising the test data subset varied 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 higher efficiency of a two-layer LSTM recurrent neural network with 35 neurons in the recurrent layers, according to all utilized quality criteria. This RNN model was used in subsequent studies.

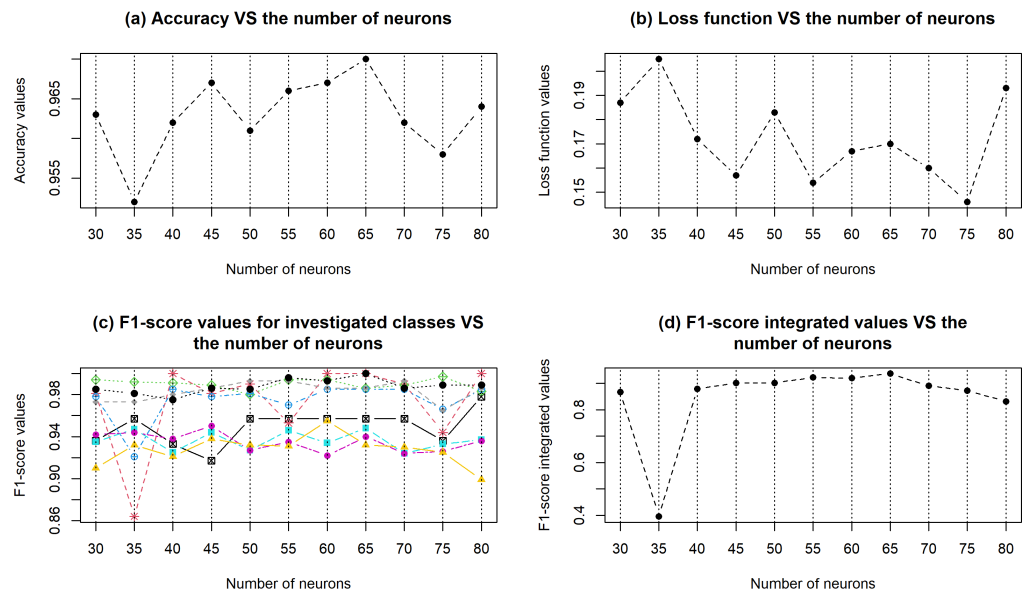


Figure 3. Results of modeling when applying a three-layer LSTM recurrent neural network. The different colors in (c) correspond to different classes.

5.2. Modeling of GRU Recurrent Neural Network

In Figures 4–6, the modeling results using a GRU recurrent neural network are depicted. The analysis of the obtained results also indicates the high effectiveness of this type of RNN for classifying data based on gene expression. However, compared with 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 the achieving of 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.

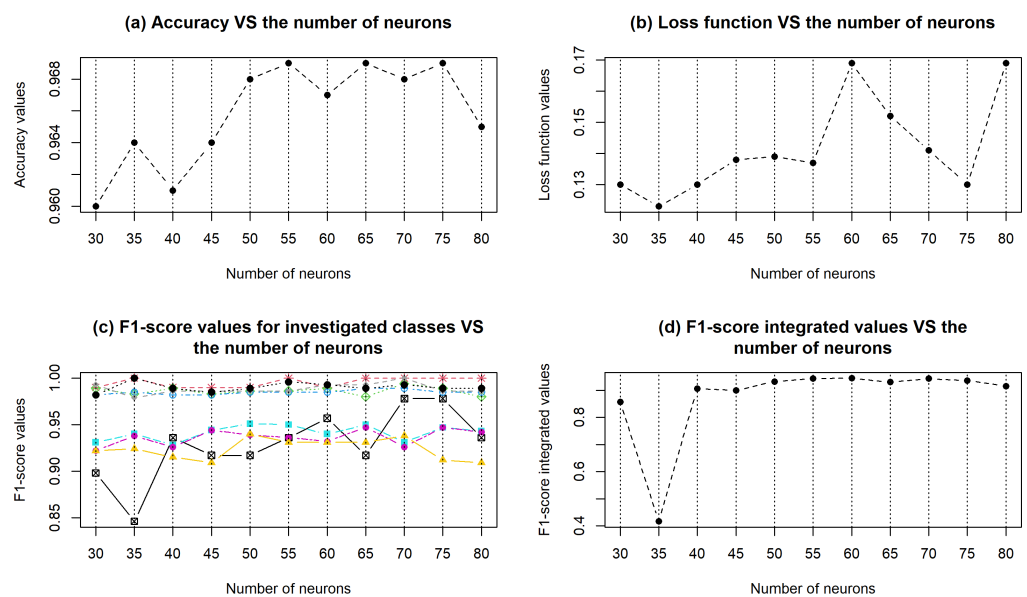


Figure 4. Results of modeling when applying a single-layer GRU recurrent neural network. The different colors in (c) correspond to different classes.

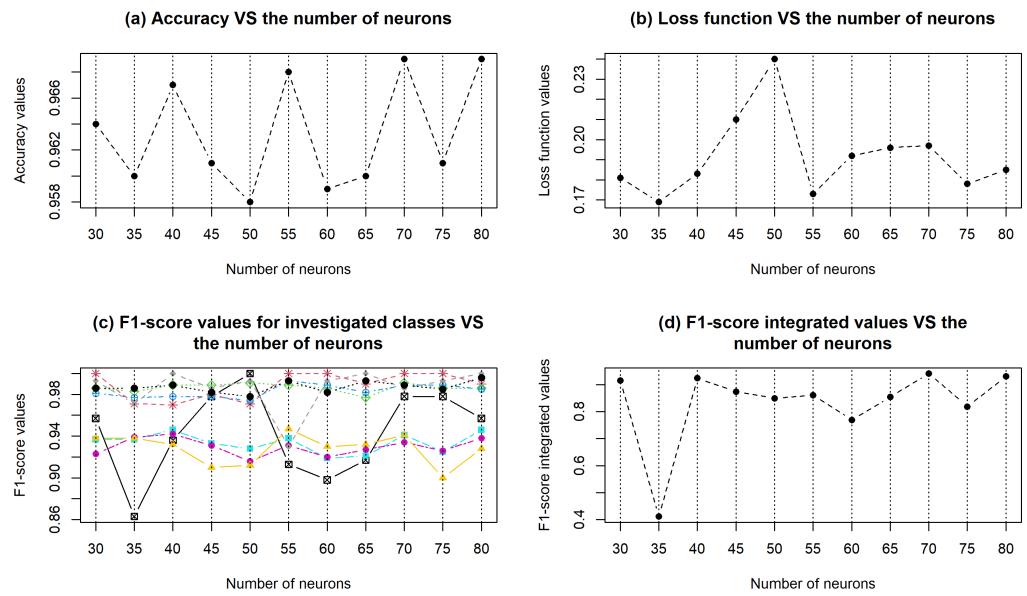


Figure 5. Results of modeling when applying a two-layer GRU recurrent neural network. The different colors in (c) correspond to different classes.

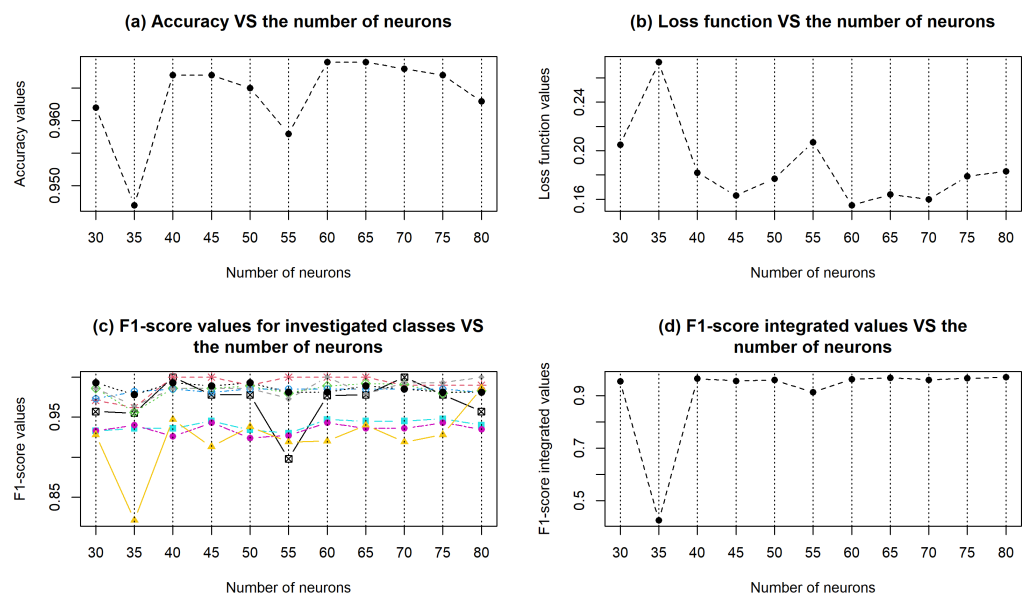


Figure 6. Results of modeling when applying a three-layer GRU recurrent neural network. The different colors in (c) correspond to different classes.

5.3. Calculating the Comprehensive Quality Criterion for the Classification of Gene Expression Data

The analysis of the simulation results, presented hereinbefore, indicates challenges to 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 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:

$$QC_w = \sum_{k=1}^n weight_k \cdot QC_k \tag{12}$$

where $weight_k$ denotes the weight of the corresponding k -th quality criterion (QC_k).

The algorithm for calculating the criterion (6) 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:

$$loss'_k = max(loss) - loss_k \tag{13}$$

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

$$QC_k^{norm} = \frac{QC_k - min(QC)}{max(QC) - min(QC)} \tag{14}$$

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, was half as much as the weights of accuracy and integrated F1-score, which were calculated on the test data subset. Therefore, the weight vector for the criterion 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 (12):

$$QC_k^{compr} = w[1] \cdot ACC_k^{norm} + w[2] \cdot F1_k^{norm} + w[3] \cdot loss_k^{norm} \tag{15}$$

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

The proposed methodology was tested using results obtained in the previous subsections during the simulation of LSTM and GRU recurrent neural networks with various sets of hyperparameters. Figures 7 and 8 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.

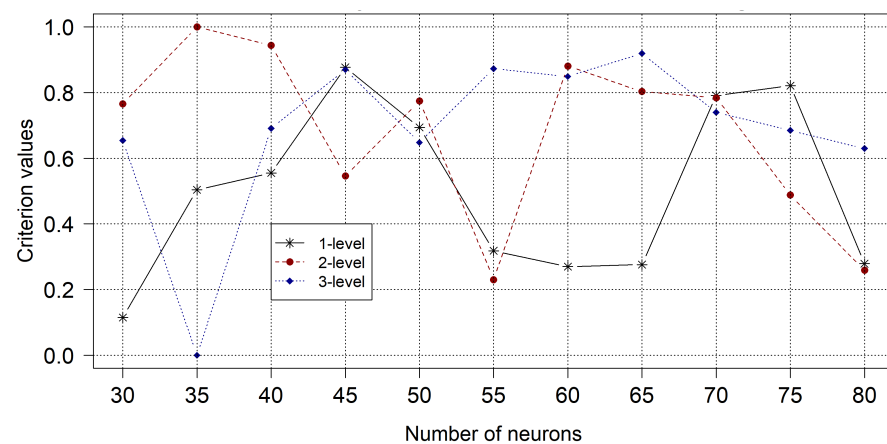


Figure 7. Distribution diagrams of the classification comprehensive quality criterion when using LSTM recurrent neural network.

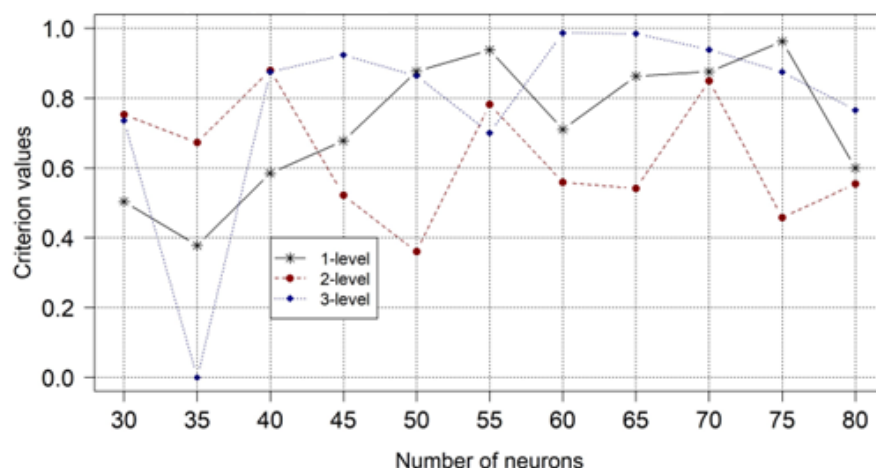


Figure 8. Distribution diagrams of the classification comprehensive quality criterion when using GRU recurrent neural network.

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 is 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.

6. Comparative Analysis of Convolutional and Recurrent Neural Networks with Optimal Hyperparameters

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 previous ones, the data were divided into three subsets: for network training, its validation during the training process, and testing 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 number of filters = 32, kernel size = 3, Dense kernel = 64, maximal pooling = 3; the activation functions for convolutional layer, dense layer, and output layer were *sigmoid*, *selu*, and *softmax*, respectively.

In Figures 9–11, 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 criterion 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 39 s, which was significantly less than that when using LSTM (185 s) and GRU (166 s) recurrent neural networks.

In Figure 12, the diagrams of classification quality criteria based on gene expression data are depicted when applying different types of deep neural networks.

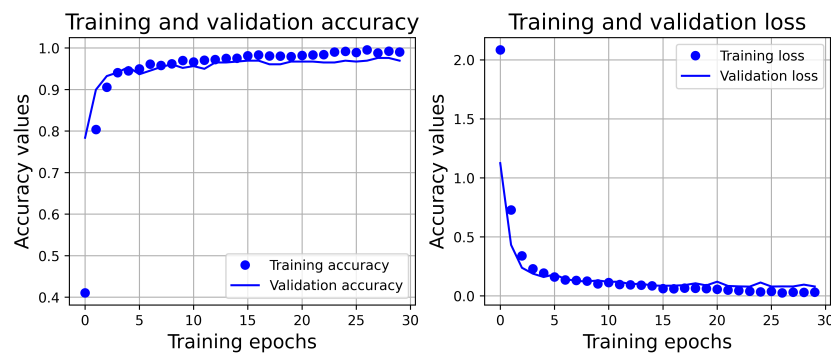


Figure 9. Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the CNN model.

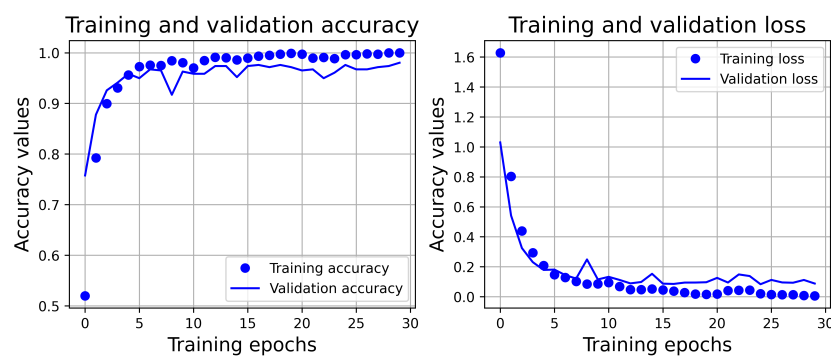


Figure 10. Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the LSTM RNN model.

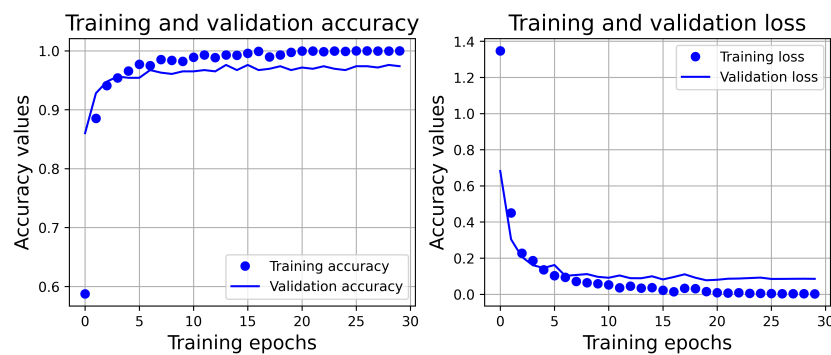


Figure 11. Diagrams of changes in accuracy values and the loss function, calculated on the training subset and during model validation when applying the GRU RNN model.

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, 954 out of 981 objects were correctly identified. In other cases, 952 were correctly identified. 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. The 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 forming the vector of optimal hyperparameters during the model tuning process.

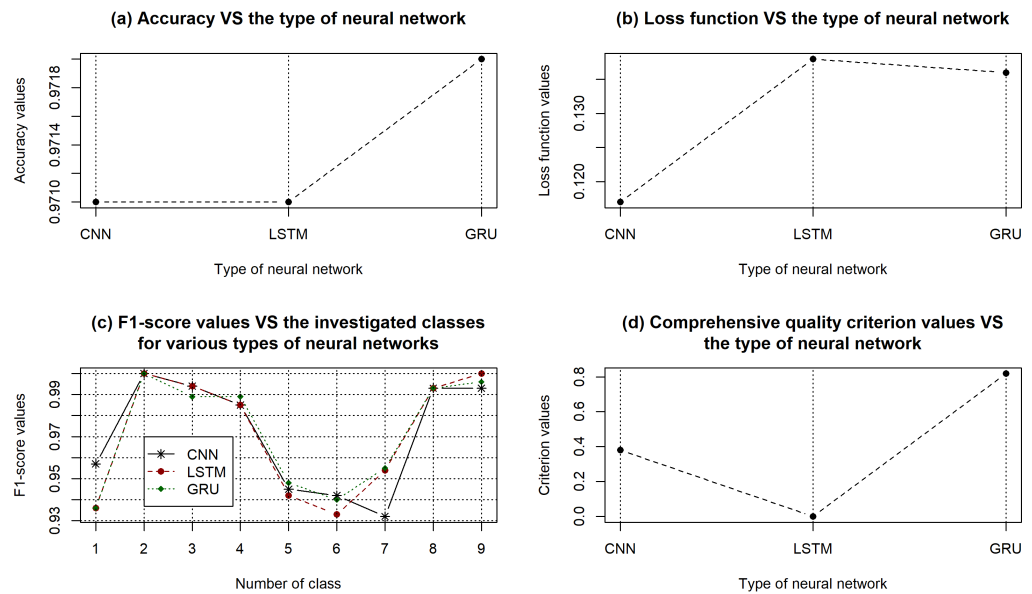


Figure 12. The simulation results when applying the GRU RNN model.

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 classification 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, they are 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, our proposed methodology, based on recurrent neural networks, offers advantages in hyperparameter optimization compared with 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 the selection of the most suitable model by considering various classification quality criteria and assigning appropriate significance weights.

However, the limitation of our 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 could also pave the way for a comparative analysis of different deep learning model types for gene expression data processing.

7. Conclusions

The manuscript presents research results regarding the application of a recurrent neural network (RNN) for processing gene expression data. Two types of RNNs, LSTM and GRU, were investigated. An algorithm for optimizing the architecture and hyperparameter values of the RNN, which involves calculating both the accuracy of sample classification and the evaluation of the F1-score, the value of which allows for the assessment of the quality of sample distribution into respective classes, is proposed. An integral criterion of the F1-score based on the desirability method by Harrington was calculated to enhance the objectivity of decision making regarding model efficiency. Also proposed is a comprehensive data classification quality criterion using the relevant type of deep learning network, calculated as a weighted sum of partial quality criteria determined during the simulation process.

The modeling of various RNN architectures was carried out, and as a result, optimal hyperparameter values for each type of network were determined. The simulation results enabled the conclusion regarding the higher appeal of the single-layer GRU recurrent network with 75 neurons in the recurrent layer. A comparative analysis of convolutional and recurrent neural networks with optimal hyperparameters was also performed. It is shown 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%, and in other cases, 97.1%. In the first case, 954 out of 981 objects were correctly identified. In other cases, 952 were correctly identified. Although the convolutional neural network is more attractive in terms of the loss function value and training time, the GRU recurrent neural network is more appealing based on a set of criteria.

The authors' further research perspective involves investigating the Bayesian optimization algorithm for optimizing the model's hyperparameter values with a comparative analysis of different types of models and methods for optimizing their parameters.

Author Contributions: Conceptualization, formal analysis, resources, writing—review and editing: S.B., I.K., and I.L.; methodology, software (R and Python programming), validation, statistical analysis and investigation, writing—original draft preparation: S.B.; results visualization: I.K. and I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval for this study were waived due to the type of study (retrospective study). The datasets from the freely available TCGA database were used during the simulation procedure implementation.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available from the corresponding author upon request.

Acknowledgments: We express our gratitude to the research team from the Center for Cancer Genomics at the National Cancer Institute, National Institutes of Health, and The Cancer Genome Atlas (TCGA) for providing the opportunity to download and utilize the gene expression datasets from patients investigated for various types of cancer diseases.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
RNN	Recurrent neural network
TCGA	The Cancer Genome Atlas
DL	Deep learning
LOAD	Late-onset Alzheimer's disease
RCO	Rider Chicken Optimization
CPM	Count Per Million
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit

References

1. Shukla, V.; Rani, S.; Mohapatra, R.K. A New Approach for Leaf Disease Detection using Multilayered Convolutional Neural Network. In Proceedings of the 2023 3rd International Conference on Artificial Intelligence and Signal Processing, AISP 2023, Vijayawada, India, 18–20 March 2023.
2. Wang, H.-Q.; Li, H.-L.; Han, J.-L.; Feng, Z.P.; Deng, H.X.; Han, X. MMDAE-HGSOC: A novel method for high-grade serous ovarian cancer molecular subtypes classification based on multi-modal deep autoencoder. *Comput. Biol. Chem.* **2023**, *105*, 107906. [[CrossRef](#)] [[PubMed](#)]
3. Yuan, M.; Feng, Y.; Zhao, M.; Xu, T.; Li, L.; Guo, K.; Hou, D. Identification and verification of genes associated with hypoxia microenvironment in Alzheimer's disease. *Sci. Rep.* **2023**, *13*, 16252. [[CrossRef](#)] [[PubMed](#)]

4. Liu, H.; Arsie, R.; Schwabe, D.; Schilling, M.; Minia, I.; Alles, J.; Boltengagen, A.; Kocks, C.; Falcke, M.; Friedman, N.; et al. SLAM-Drop-seq reveals mRNA kinetic rates throughout the cell cycle. *Mol. Syst. Biol.* **2023**, *19*, e11427. [[CrossRef](#)]
5. Mohamed, T.I.A.; Ezugwu, A.E.; Fonou-Dombeu, J.V.; Ikotun, A.M.; Mohammed, M. A bio-inspired convolution neural network architecture for automatic breast cancer detection and classification using RNA-Seq gene expression data. *Sci. Rep.* **2023**, *13*, 14644. [[CrossRef](#)]
6. Zheng, P.; Zhang, G.; Liu, Y.; Huang, G. MultiScale-CNN-4mCPred: A multi-scale CNN and adaptive embedding-based method for mouse genome DNA N4-methylcytosine prediction. *BMC Bioinform.* **2023**, *24*, 21. [[CrossRef](#)]
7. Davri, A.; Birbas, E.; Kanavos, T.; Ntritsos, G.; Giannakeas, N.; Tzallas, A.T.; Batistatou, A. Deep Learning for Lung Cancer Diagnosis, Prognosis and Prediction Using Histological and Cytological Images: A Systematic Review. *Cancers* **2023**, *15*, 3981. [[CrossRef](#)] [[PubMed](#)]
8. Chuang, Y.H.; Huang, S.H.; Hung, T.M.; Lin, X.Y.; Lee, J.Y.; Lai, W.S.; Yang, J.M. Convolutional neural network for human cancer types prediction by integrating protein interaction networks and omics data. *Sci. Rep.* **2021**, *11*, 20691. [[CrossRef](#)] [[PubMed](#)]
9. Li, J.; Sun, W.; Feng, X.; Xing, G.; von Deneen, K.M.; Wang, W.; Zhang, Y.; Cui, G. A dense connection encoding–decoding convolutional neural network structure for semantic segmentation of thymoma. *Neurocomputing* **2021**, *451*, 1–11. [[CrossRef](#)]
10. Gholami, H.; Mohammadifar, A.; Golzari, S.; Song, Y.; Pradhan, B. Interpretability of simple RNN and GRU deep learning models used to map land susceptibility to gully erosion. *Sci. Total. Environ.* **2023**, *904*, 166960. [[CrossRef](#)]
11. 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. *Appl. Sci.* **2013**, *13*, 6022. [[CrossRef](#)]
12. Yasinska-Damri, L.; Babichev, S.; Durnyak, B.; Goncharenko, T. Application of Convolutional Neural Network for Gene Expression Data Classification. *Lect. Notes Data Eng. Commun. Technol.* **2023**, *149*, 3–24.
13. 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. *Appl. Sci.* **2022**, *12*, 11795. [[CrossRef](#)]
14. Zhang, J.; Liu, B.; Wu, J.; Wang, Z.; Li, J. DeepCAC: A deep learning approach on DNA transcription factors classification based on multi-head self-attention and concatenate convolutional neural network. *BMC Bioinform.* **2023**, *24*, 345. [[CrossRef](#)]
15. Shigemizu, D.; Akiyama, S.; Suganuma, M.; Furutani, M.; Yamakawa, A.; Nakano, Y.; Ozaki, K.; Niida, S. Classification and deep-learning-based prediction of Alzheimer disease subtypes by using genomic data. *Transl. Psychiatry* **2023**, *13*, 232. [[CrossRef](#)] [[PubMed](#)]
16. Busaleh, M.; Hussain, M.; Aboalsamh, H. Breast mass classification using diverse contextual information and convolutional neural network. *Biosensors* **2022**, *11*, 419. [[CrossRef](#)] [[PubMed](#)]
17. Cao, X.; Pan, J.S.; Wang, Z.; Sun, Z.; ul Haq, A.; Deng, W.; Yang, S. Application of generated mask method based on mask r-cnn in classification and detection of melanoma. *Comput. Methods Programs Biomed.* **2021**, *207*, 106174. [[CrossRef](#)] [[PubMed](#)]
18. Mostavi, M.; Chiu, Y.C.; Huang, Y.; Chen, Y. Convolutional neural network models for cancer type prediction based on gene expression. *BMC Med. Genom.* **2020**, *13*, 44. [[CrossRef](#)]
19. Ramires, R.; Chiu, Y.; Horerra, A.; Mostavi, M.; Ramirez, J.; Chen, Y.; Huang, Y.; Jin, Y.-F. Classification of cancer types using graph convolutional neural networks. *Front. Phys.* **2020**, *8*, 203. [[CrossRef](#)]
20. Choi, J.M.; Chae, H. moBRCA-net: A breast cancer subtype classification framework based on multi-omics attention neural networks. *BMC Bioinform.* **2023**, *24*, 169. [[CrossRef](#)]
21. Tapak, L.; Ghasemi, M.K.; Afshar, S.; Mahjub, H.; Soltanian, A.; Khotanlou, H. Identification of gene profiles related to the development of oral cancer using a deep learning technique. *BMC Med. Genom.* **2023**, *16*, 35. [[CrossRef](#)]
22. Mustafa, E.; Jadoon, E.K.; Khaliq-uz-Zaman, S.; Humayun, M.A.; Maray, M. An Ensembled Framework for Human Breast Cancer Survivability Prediction Using Deep Learning. *Diagnostics* **2023**, *13*, 1688. [[CrossRef](#)]
23. Monti, M.; Fiorentino, J.; Milanetti, E.; Gosti, G.; Tartaglia, G.G. Prediction of Time Series Gene Expression and Structural Analysis of Gene Regulatory Networks Using Recurrent Neural Networks. *Entropy* **2022**, *24*, 141. [[CrossRef](#)] [[PubMed](#)]
24. Majji, R.; Nalinipriya, G.; Vidyadhari, C.; Cristin, R. Jaya Ant lion optimization-driven Deep recurrent neural network for cancer classification using gene expression data. *Med. Biol. Eng. Comput.* **2021**, *59*, 1005–1021. [[CrossRef](#)] [[PubMed](#)]
25. Aher, C.N.; Aher, A.K. Rider-chicken optimization dependent recurrent neural network for cancer detection and classification using gene expression data. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2021**, *9*, 174–191. [[CrossRef](#)]
26. The Cancer Genome Atlas Program (TCGA). El. Resource. Available online: <https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga> (accessed on 15 March 2021).
27. Illumina. El. Resource. Available online: <https://www.illumina.com/> (accessed on 15 March 2021).
28. Vural, S.; Wang, X.; Guda, C. Classification of breast cancer patients using somatic mutation profiles and machine learning approaches. *BMC Syst. Biol.* **2016**, *10*, 264–276. [[CrossRef](#)]
29. Phoa, F.K.H.; Chen, H.-W. Desirability function approach on the optimization of multiple Bernoulli-distributed response. In Proceedings of the ICPRAM 2013-Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods, Barcelona, Spain, 15–18 February 2013; pp. 127–131.

30. Zhao, Y.; Chen, Z.; Dong, Y.; Tu, J. An interpretable LSTM deep learning model predicts the time-dependent swelling behavior in CERCER composite fuels. *Mater. Today Commun.* **2023**, *37*, 106998. [[CrossRef](#)]
31. Amendolara, A.B.; Sant, D.; Rotstein, H.G.; Fortune, E. LSTM-based recurrent neural network provides effective short term flu forecasting. *BMC Public Health* **2023**, *23*, 1788. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.