AIMarkerFinder: AI-Assisted Marker Discovery Based on an Integrated Approach of Autoencoders and Kolmogorov-Arnold Networks

Pavel S. Demenkov *, Timofey V. Ivanisenko, Vladimir A. Ivanisenko

Posted Date: 24 November 2025

doi: 10.20944/preprints202511.1705.v1

Keywords: denoising autoencoder; Kolmogorov-Arnold Network; KAN; DAE; feature selection; metabolomics; bioinformatics

 doi:10.20944/preprints202511.1705.v1

Article

AIMarkerFinder: AI-Assisted Marker Discovery Based on an Integrated Approach of Autoencoders and Kolmogorov-Arnold Networks

Pavel S. Demenkov *, Timofey V. Ivanisenko and Vladimir A. Ivanisenko

The Artificial Intelligence Research Center of Novosibirsk State University, Pirogova Street 1, Novosibirsk 630090, Russia

* Correspondence: demps@bionet.nsc.ru

Abstract

In modern bioinformatics, the analysis of high-dimensional data (genomic, metabolomic, etc.) remains a critical challenge due to the "curse of dimensionality," where feature redundancy reduces classification efficiency and model interpretability. This study introduces a novel method, AIMarkerFinder, for analyzing metabolomic data to identify key biomarkers. The method is based on a denoising autoencoder with an attention mechanism (DAE), enabling the extraction of informative features and the elimination of redundancy. Experiments on glioblastoma and adjacent tissue metabolomic data demonstrated that AIMarkerFinder reduces dimensionality from 446 to 4 key features while improving classification accuracy. Using the selected metabolites (Malonyl-CoA, Glycerophosphocholine, SM(d18:1/22:0 OH), GC(18:1/24:1)), the Random Forest and Kolmogorov-Arnold Network (KAN) models achieved accuracies of 0.904 and 0.937, respectively. The analytical formulas derived by KAN provide model interpretability, which is critical for biomedical research. The proposed approach is applicable to genomics, transcriptomics, proteomics, and the study of exogenous factors on biological processes. The study's results open new prospects for personalized medicine and early disease diagnosis. AIMarkerFinder is avaliable at *https://github.com/dps123/AIMarkerFinder*

Keywords: denoising autoencoder; Kolmogorov-Arnold Network; KAN; DAE; feature selection; metabolomics; bioinformatics

1. Introduction

Recent advances in high-throughput technologies, such as next-generation sequencing (NGS), microarrays, and mass spectrometry (MS), have opened new opportunities for researchers in identifying the genetic causes of diseases [1,2]. Radiomics is a new technology in which medical imaging provides important information about tumor physiology [3]. Metabolomics, as part of systems biology, focuses on the comprehensive study of the metabolome—the collection of low-molecular-weight compounds (metabolites) involved in the biochemical processes of an organism. In recent years, this discipline has gained particular relevance due to its ability to reveal subtle mechanisms of cellular function regulation and predict changes in response to external and internal stimuli, such as diseases, drugs, or dietary changes. Metabolomic data provide a unique opportunity for the early diagnosis of pathologies, the development of personalized treatment methods, and a deeper understanding of the interplay between genetics, environment, and lifestyle [4].

These high-throughput representations suffer from the curse of dimensionality, so appropriate computational methods are required to extract knowledge from them [5]. Microarray data contain many heterogeneity factors because they include the expression of every possible gene in the genome. Scientific studies have proven that genes responsible for certain biological processes are interrelated, and some genes act as activators or inhibitors of others [6]. In high-dimensional data, such as

 doi:10.20944/preprints202511.1705.v1

microarray datasets, irrelevant features can interfere with true features, which in turn introduces heterogeneity into the data and generates dependencies between features. Statistical analysis loses its significance in the case of dependent features. Therefore, we must select features that play an important role in evaluation and are independent.

Identifying such independent genes (features) whose expression models have significant biological associations with phenotypic behavior is important for knowledge discovery. In microarray analysis, the goal for biologists is to detect a small number of features that explain microarray data behavior [7]. Selected significant biomarkers from microarray data are essential for patient stratification and the development of personalized medicine strategies [8]. From a machine learning perspective, controlling the number of features helps reduce overfitting, leading to better prediction of the target variable on training data. The dimensionality of the feature space casts doubt on model construction and the effectiveness of knowledge discovery. Therefore, a recommended ratio of 10:1 samples per feature is suggested for creating reliable classifiers and predictive models [9].

The reason for feature selection is that classifiers trained on a reduced feature space are more robust and reproducible than those built on the original large feature space. In feature selection, we particularly look for correlated features. Features that do not provide useful information are called irrelevant features, while features that do not provide additional information beyond already selected features are termed redundant features [10]. Features that are uncorrelated or unassociated with class variables are referred to as noise, which actually introduces bias into prediction and reduces classification efficiency. Therefore, noise must be removed to improve predictive performance, and this can be achieved through dimensionality reduction. This can be done either through feature extraction or feature selection [11].

In feature extraction, new features are derived from the original input data by selecting a new basis for the data. Feature selection helps reduce the impact of high dimensionality on a dataset by identifying a subset of features that effectively define the data. Direct evaluation of feature subsets becomes an NP-hard problem [12]. To address this, we attempt to use suboptimal procedures with tractable computations. We must also address another important issue when a feature depends on the response variable rather than the predictors. Selecting a subset of features allows classifiers to focus on important features while ignoring potentially misleading ones. From a computational complexity perspective, having an economical set of features involved in the classification process helps scale many learning algorithms rapidly with additional features [13].

The best feature selection algorithm should always bring benefits such as data understanding, a better classifier model, improved generalization, and the identification of irrelevant features. It should also assist in understanding the relationships between features and target variables, reduce computational requirements for solving a specific task, enable efficient dimensionality reduction in the case of high-dimensional datasets where the number of observations is less than the number of features, help improve the performance of the predictor used to solve a specific task, and enhance efficiency in terms of cost and time. The feature selection process contributes to knowledge discovery, where the identified features can be directly used in future research. In bioinformatics, the identification of important features can indicate new metabolic pathways and help reveal hidden connections between specific cellular processes [13].

This paper describes the developed method AIMarkerFinder for highlighting important features, based on a denoising autoencoder (DAE) with an attention mechanism. The AIMarkerFinder method highlights important features in the data, enabling more accurate classifier models.

 doi:10.20944/preprints202511.1705.v1

2. Materials and Methods

2.1. Metabolomics Data

Feature extraction was performed on a dataset of metabolomic profiles from glioblastoma and adjacent brain tissue from the study by Basso et al. [14]. The dataset contains 32 samples (16 glioblastoma and 16 adjacent tissues), with the concentration of 446 metabolites measured.

2.2. Data Preparation

During the training phase, the input data underwent preprocessing consisting of two steps: class balancing and noise injection.

To address class imbalance, the RandomOverSampler method from the imbalanced-learn library (abbreviated as imblearn) was used. This method increased the number of instances in underrepresented classes by randomly selecting and duplicating existing samples.

If the dataset size after balancing was less than 3000 samples, each vector was duplicated repeatedly until the total dataset size exceeded 3000.

After class balancing, Gaussian noise was added to each component of the feature vector. The process is described by the formula:

$$C^{noisy}_n = C^{original}_n + \varepsilon, \quad (1)$$

where:

C^{noisy}_n — the noisy value of the n-th component of the vector,

$C^{original}_n$ — the original value of the n-th component,

$\varepsilon \sim N(0, \sigma_n \cdot d)$ — noise sampled from a normal distribution with mean 0 and variance $\sigma n \cdot \sigma_n \cdot d$.

Parameters:

σ_n — the sample variance of the n-th component, calculated separately for each class,

d=0.25 — a coefficient determining the noise level.

The variance σ_n, calculated for each class and vector component, preserved the statistical characteristics of the original data. The chosen coefficient d=0.25 ensured a moderate noise level sufficient to enhance the diversity of training examples without distorting their semantic meaning.

This approach simultaneously addressed class balancing and increased the size of the training data by generating synthetic noisy samples, which improved the model's generalization capability.

2.3. Implementation

The proposed denoising autoencoder (DAE) with an attention mechanism was implemented using the PyTorch framework. The Adam optimizer was used for training with a learning rate of 0.001. The model was trained for 10,000 epochs with a batch size of 2048. Early stopping was allowed if the error stabilized within 0.005 for 100 consecutive epochs. During training, model parameters achieving the best loss values on validation data were saved.

Training of the random forest and C4.5 models was performed using the sklearn library. For the random forest model, hyperparameter optimization was conducted with GridSearchCV using the parameters:

n_estimators: np.arange(10, 201, 20),

max_depth: np.arange(3, 15, 2),

random_state: [0].

For the C4.5 model, the GridSearchCV parameters were set as:

criterion: ['entropy'],

max_depth: np.arange(1, 21),

min_samples_split: np.arange(2, 11),

max_leaf_nodes: np.arange(3, 26),

max_features: ['sqrt'],

 doi:10.20944/preprints202511.1705.v1

random_state: np.arange(0, 10).

The accuracy of the C4.5 and Random Forest (RF) models was evaluated using 5-fold cross-validation.

The KAN method implementation was based on the pykan library [15]. The network structure consisted of 3 layers: the first and last layers had dimensions matching the input and output data, respectively, while the intermediate layer had a dimension one-third that of the input data. The model was trained with parameters:

grid=5,
k=3,
scale_base_mu=1,
scale_base_sigma=2,
noise_scale=0.3.

The optimizer was LBFGS, and the loss function was CrossEntropyLoss. Training was repeated 100 times with different initial values of the seed parameter (0–99). The model with the highest accuracy was selected.

3. Results

3.1. Denoising Autoencoder

Autoencoders are a powerful tool for noise suppression tasks, allowing efficient extraction of clean signals from noisy data. However, to achieve high performance in real-world conditions, autoencoders must be trained on diverse types of noise. The study employs a denoising autoencoder (DAE) model with an attention mechanism for the classification task. The network architecture is based on the article Guan et al.[16] and was modified to detect the most informative features.

The general scheme of the AIMarkerFinder method is presented in Figure 1. It consists of four main components: an attention module, an encoder, a decoder, and a classifier. The attention module automatically identifies the most significant components of the input vector. The encoder compresses the input data to the hidden layer dimension. The decoder restores the original data dimension from the compressed representation. The classifier predicts the object category.

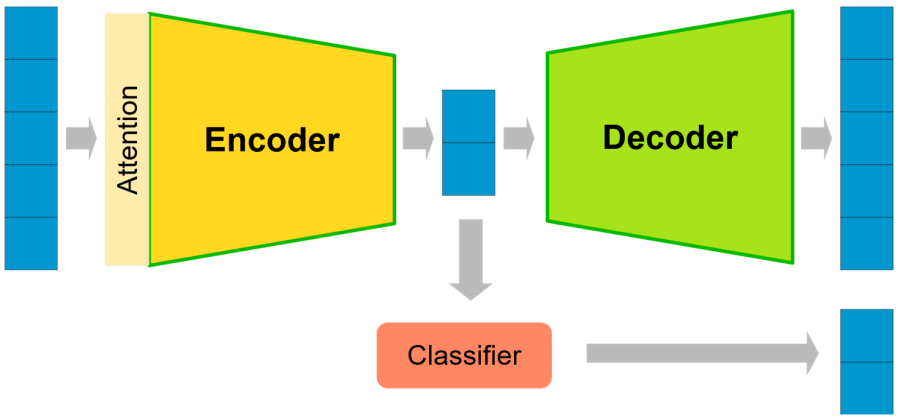

Figure 1. General scheme of the denoising autoencoder with an attention mechanism used in the study.

During training, the data first pass through the attention module for "weighting" the features. They are then fed into the encoder and projected into a low-dimensional hidden feature space. The classifier is trained on compressed representations using category labels, enhancing the discriminative ability of latent features. Simultaneous decoder training ensures the preservation of maximum information necessary for data reconstruction.

The attention module is implemented as a fully connected layer with a combination of Softmax and Sigmoid activation functions. The Sigmoid function provides a clearer boundary between

 doi:10.20944/preprints202511.1705.v1

important and secondary features. The obtained coefficients are used to weight the input data fed into the encoder.

The encoder consists of 2 layers, where the size of the first layer matches the input vector size, and the hidden layer size is determined by the number of principal components explaining 99% of the data variance. The decoder has a symmetric architecture to the encoder.

The classifier receives data from the encoder output. Its architecture also consists of 2 fully connected layers: the first layer is half the size of the autoencoder's hidden layer, and the output layer corresponds to the number of recognizable classes.

All model components are trained jointly, enabling the encoder to find features that simultaneously separate data into classes and reflect data properties.

The loss function includes a weighted sum of three components: Mean Squared Error (MSE) and the coefficient of determination (R^2) for evaluating data reconstruction quality, as well as Binary Cross-Entropy (BCE) for classification quality:

$$\text{Loss} = \text{MSE} + \lambda_1*(1 - R^2) + \lambda_2*\text{BCE}. \quad (2)$$

Parameters λ_1 and λ_2 in equation (2) are set to 0.1 and 0.3, respectively.

3.2. Selection of the Most Significant Features

After training the model, a set of parameters with the best loss function performance on validation data is selected. The original data are input into the model, and the output includes latent layer vectors (embedding) and attention layer vectors (attention). Embedding vectors are used for visualization and can further be employed to build classification models. The attention layer vectors are averaged component-wise. The components of the resulting averaged attention vector (Matt) are sorted in descending order. The index of the component where the maximum change in values occurs is determined by the formula:

$$k_{max} = \underset{k}{argmax} \left(\frac{Matt[k-1] - Matt[k]}{Matt[k]}\right). \quad (3)$$

Attention components with indices not exceeding k_{max} are considered the most significant for the model. However, since the model simultaneously solves two tasks (compression and classification), the selected features include both classification-relevant features and those reflecting data structure. Therefore, in the next step, a Random Forest method was applied to select features with high classification ability. Training was conducted on a dataset containing only the previously selected features. As a result of Random Forest model training, a feature importance vector (FI) was obtained. The components of the FI vector are sorted in descending order and normalized by the median value. To account for model quality, all components of the vector are multiplied by the average accuracy of cross-validated models raised to the power of Alpha. The exponent Alpha is critical and determines the method's sensitivity. At Alpha = 2, the model selects only features enabling high-accuracy classification, while at Alpha = 1, it allows lower-quality classification. In the implemented method, Alpha was evaluated as a function of the number of selected DAE features and the size of the DAE latent layer:

$$\text{Alpha} = (\text{latent layer size} + \text{number of selected features}) / \text{latent layer size} \quad (4)$$

For the obtained sorted and normalized vector, the inflection point of the graph is found using the Piecewise Linear Regression algorithm [17].

Normalized values greater than 1.0 are considered a necessary condition for the presence of classification-capable features. If no such values exist in the vector, the algorithm stops.

Data structures can be complex, and the solution to the feature selection problem for classification may be ambiguous. To address this, AIMarkerFinder proposes iteratively extracting important features by removing already selected features from the data.

As a result of iterative algorithm runs, a list of important features is identified. For this list, a classifier is built using the Random Forest (RF) method, and a feature importance vector is computed.

 doi:10.20944/preprints202511.1705.v1

Feature weights are normalized by the median value, and the inflection point is determined using the Piecewise Linear Regression algorithm. This results in a reduced set of important features used in the classifier model.

3.3. Data Classification

For building classifiers on the selected features, the following models were used: C4.5 [18], Random Forest (RF) [19], and Kolmogorov-Arnold Network (KAN) [15]. The C4.5 method constructs a decision tree, generating a tree where internal nodes represent features (or attributes), and leaf nodes represent classes or labels. Each path from root to leaf represents a classification rule. The method is easily interpretable as it results in a set of logical rules.

Random Forest is an ensemble method that builds multiple decision trees and aggregates their predictions to improve accuracy and model stability. Each tree is trained on a random subsample of data (with replacement) and uses randomly selected features for splitting. The method has high accuracy, especially on complex data structures, and is robust to overfitting due to the ensemble approach.

The KAN model is an approach to constructing multidimensional functions based on Kolmogorov's theorem. This theorem states that any continuous function of multiple variables can be represented as a superposition of functions of one variable and sums. The KAN method uses this idea to build models that can produce analytical formulas describing dependencies.

To evaluate the quality of models built using C4.5 and RF, a 5-fold cross-validation procedure was employed. For the KAN model, training was conducted on a noisy dataset, and testing was performed on real data.

3.4. Analysis of Metabolomic Profile Samples

The data of metabolomic profiles of glioblastoma and adjacent brain tissue [14] have a complex structure. The results of Principal Component Analysis (PCA) on the original dataset are shown in Figure 2.

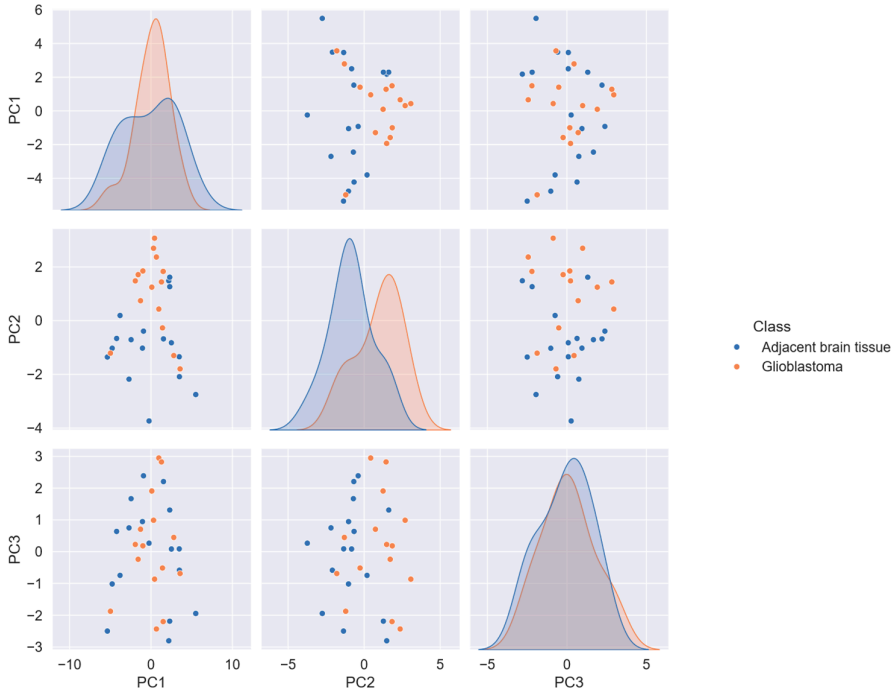

Figure 2. Results of PCA analysis on the full sample of glioblastoma and adjacent brain tissue metabolomic profiles.

 doi:10.20944/preprints202511.1705.v1

The first three principal components explain 27%, 10%, and 9% of the total data variance, respectively.

We hypothesized that using vector representations of the original data for dimensionality reduction could improve data separation. To obtain vector representations of the original data, an NSA model with an attention mechanism was trained on data including all 446 features. The length of the vector representation was set to 30. The results of PCA analysis on the 30-dimensional vector representation obtained by the trained NSA model are presented in Figure 3.

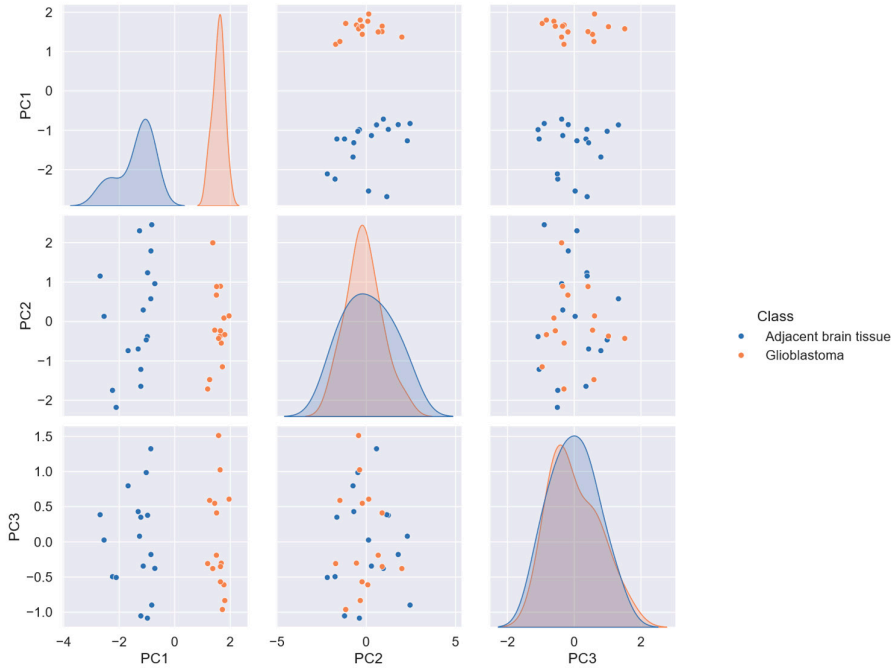

Figure 3. Results of PCA analysis on the 30-dimensional vector representation of data obtained by the trained NSA model with an attention mechanism. The first three components explain 44%, 25%, and 8% of the total data variance, respectively.

From the figure, it is evident that linear separation of groups is possible using the vector representation obtained by the NSA model with an attention mechanism. However, the constructed model does not provide an interpretable result.

To obtain an interpretable result, we applied the iterative AIMarkerFinder method to analyze the metabolomic profile sample, enabling the selection of the most important features from the original data. During the iterative feature selection process, 31 important features were identified (Table 1). Using the Piecewise Linear Regression algorithm, 4 out of the 31 most important features were identified (Figure 4).

Table 1. List of selected metabolites with their classification weights.

Metabolite Name	Score
Malonyl-CoA	**3.23**
Glycerophosphocholine	**2.17**
SM(d18:1/22:0 OH)	**1.95**
GC(18:1/24:1)	**1.73**
Pyroglutamic acid	1.54
Ceramide(d18:1/16:0 OH)	1.39
Ceramide(d18:1/16:0)	1.38
3-Phosphoglyceric acid	1.25

 doi:10.20944/preprints202511.1705.v1

4-phosphopantothenate	1.17
Hexose Disaccharide Pool	1.11
2-Octenoylcarnitine	1.11
GC(18:2/16:0)	1.03
THC 18:1/20:0	0.91
Suberylcarnitine	0.8
Ethanolamine	0.79
SM(d18:1/24:0 OH)	0.75
Phenyllactic acid	0.73
Thiamine	0.71
Coenzyme A	0.69
Cysteine-S-sulfate	0.59
2-Hydroxy-3-methylbutyric acid	0.56
Pyridoxal	0.53
Ceramide(d18:1/24:2 OH)	0.51
cysteine sulfinate	0.46
Ceramide(d18:1/26:0 OH)	0.46
1-Pyrroline-5-carboxylic acid	0.46
Ceramide(d18:1/26:0)	0.39
Citric+isocitric acid	0.38
Guanosine	0.33
Dodecanoylcarnitine	0.25
Deoxyribose phosphate	0.24

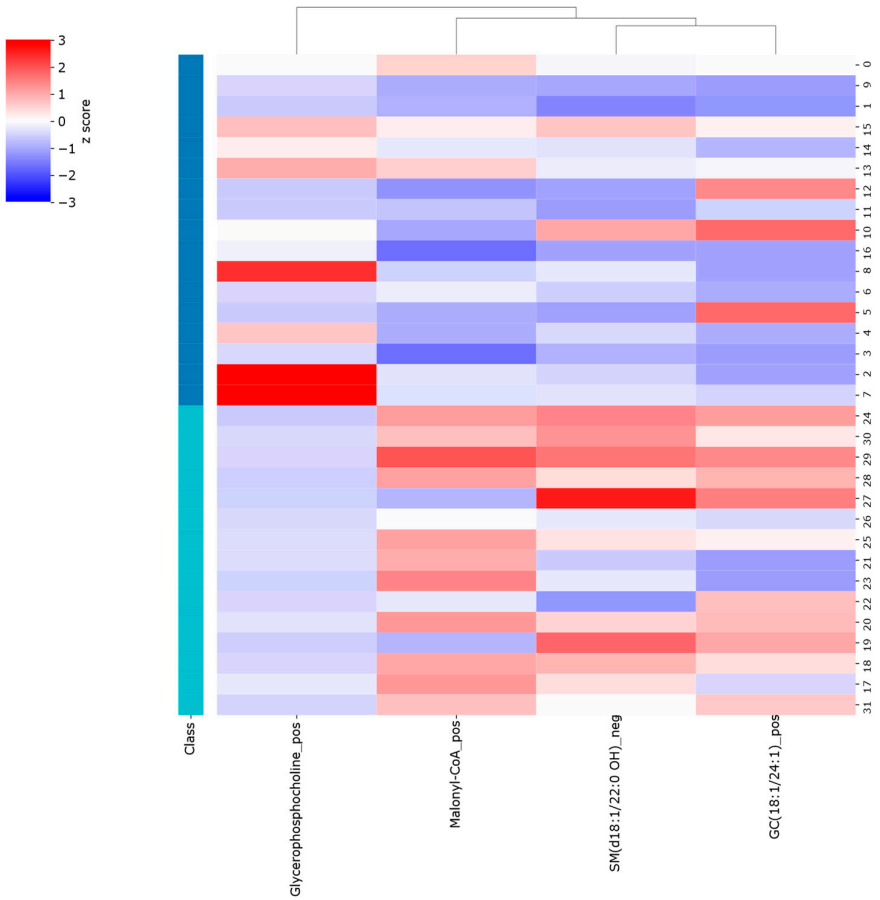

Figure 4. Heatmap built on the sample with the 4 selected features using the AIMarkerFinder method.

 doi:10.20944/preprints202511.1705.v1

Classification accuracy estimates obtained by Random Forest, C4.5, and KAN methods using both the full dataset and the selected features are presented in Table 2.

Table 2. Results of classifier model accuracy evaluation using 5-fold cross-validation.

Input Data Dimension	Random Forest	C4.5	KAN
446	0.738	0.809	-
4	0.904	0.714	0.937

Note: Training the KAN model on 446-dimensional vectors required significant computational resources and was not conducted.

As seen in the table, reducing the number of considered features significantly increases the accuracy of classifiers obtained by the RF method. The C4.5 method did not show good separation on either the full dataset or the selected features. For the KAN model, no classifier was obtained for the full dataset, but for the selected features, KAN provided analytical formulas for data classification, achieving an accuracy (0.937) higher than RF (0.904):

F1 = 12.835****Malonyl-CoA*** + 5.169****SM(d18:1/22:0 OH)*** - 5.076

F2 = 9.330****Glycerophosphocholine*** - 0.004****SM(d18:1/22:0 OH)*** - 0.001****GC(18:1/24:1)*** + 1.389

If F1 > F2, the first class is selected; otherwise, the second class.

3.5. Comparison with Analogues

To compare AIMarkerFinder with known methods, the study by Chen et al. [20] was selected, which analyzed the quality of methods (varImp(), Boruta, and Recursive Feature Elimination (RFE)) for selecting significant features in a dataset. The comparative analysis of feature selection and classifier accuracy conducted by the authors showed that the best results were achieved by the Recursive Feature Elimination (RFE) method combined with the RF classifier.

Therefore, this model was selected for comparison with AIMarkerFinder. The results of the analysis of the dependence of the RF model classification accuracy on the number of features selected by the Recursive Feature Elimination (RFE) method on the metabolomic profile dataset are presented in Figure 5.

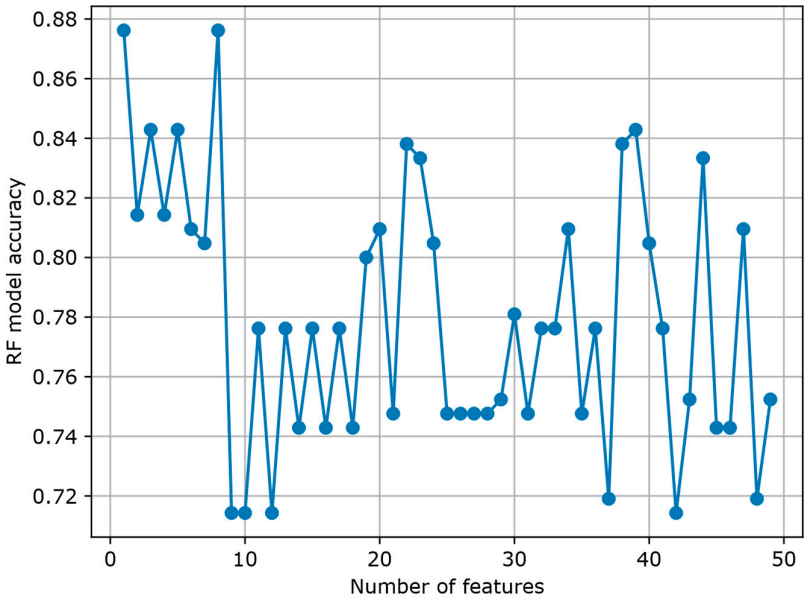

Figure 5. Dependence of RF model accuracy on the number of features selected by RFE.

 doi:10.20944/preprints202511.1705.v1

The highest accuracy scores were achieved with 1 and 8 RFE-selected features, reaching 0.876.

Unlike RFE, AIMarkerFinder automatically selects the number of important features. It is important to note that the RF accuracy on the 4 features selected by AIMarkerFinder was 0.904, which exceeded the accuracy on the 8 best RFE-selected features (0.876).

4. Discussion

The study demonstrated that the application of the denoising autoencoder with an attention mechanism effectively reduces the dimensionality of metabolomic data. In the case of analyzing metabolomic data with 446 features, the DAE with the attention mechanism enabled vector representations of 30 features. The obtained vector representations revealed a structure allowing linear separation of groups (glioblastoma and adjacent tissue). However, these vector representations require additional analysis for interpretation. To address the task of obtaining interpretable features, the AIMarkerFinder method, based on the DAE with an attention mechanism, was employed. This method selected 4 metabolites (Malonyl-CoA, Glycerophosphocholine, SM(d18:1/22:0 OH), and GC(18:1/24:1)) from the 446 features.

Classification of metabolomic data using the Random Forest and Kolmogorov-Arnold Network methods based on these 4 metabolites showed accuracy of 0.904 and 0.937, respectively. The analytical functions derived by KAN from the selected metabolites not only provide high accuracy but also allow model interpretation through linear dependencies, which is critical for biomedical applications.

The developed approach can be applied to analyze high-dimensional data in various biological fields (genomic, transcriptomic, proteomic, and metabolomic data), medical data, and for studying the impact of environmental factors (e.g., environmental pollution, diet, physical activity, or chemical exposure).

Author Contributions: Conceptualization, I.V.A.; writing—original draft preparation, P.S.D.; software, P.S.D.; validation, I.T.V; supervision, I.V.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant for research centers, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement with the Novosibirsk State University dated April 17, 2025 No. 139-15-2025-006: IGK 000000C313925P3S0002.

Data Availability Statement: AIMarkerFinder is avaliable at https://github.com/dps123/AIMarkerFinder.

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

Abbreviations

The following abbreviations are used in this manuscript:

DAE	Denoising autoencoder
RF	Random forest
KAN	Kolmogorov-Arnold Networks

References

1. Mohammadi, M., Sharifi Noghabi, H., Abed Hodtani, G., & Rajabi Mashhadi, H. (2016). Robust and stable gene selection via Maximum–Minimum Correntropy Criterion. Genomics, 107(2–3), 83–87. https://doi.org/10.1016/j.ygeno.2015.12.006
2. Taylor, A., Steinberg, J., Andrews, T. S., & Webber, C. (2014). GeneNet Toolbox for MATLAB: a flexible platform for the analysis of gene connectivity in biological networks. Bioinformatics, 31(3), 442–444. https://doi.org/10.1093/bioinformatics/btu669
3. Parmar, C., Grossmann, P., Bussink, J., Lambin, P., & Aerts, H. J. W. L. (2015). Machine Learning methods for Quantitative Radiomic Biomarkers. Scientific Reports, 5(1). https://doi.org/10.1038/srep13087

 doi:10.20944/preprints202511.1705.v1

4. Nicholson, J. K., Connelly, J., Lindon, J. C., & Holmes, E. (2002). Metabonomics: a platform for studying drug toxicity and gene function. Nature Reviews Drug Discovery, 1(2), 153–161. https://doi.org/10.1038/nrd728
5. Hinrichs, A., Prochno, J., & Ullrich, M. (2019). The curse of dimensionality for numerical integration on general domains. Journal of Complexity, 50, 25–42. https://doi.org/10.1016/j.jco.2018.08.003
6. Perthame, É., Friguet, C., & Causeur, D. (2015). Stability of feature selection in classification issues for high-dimensional correlated data. Statistics and Computing, 26(4), 783–796. https://doi.org/10.1007/s11222-015-9569-2
7. Kumar, A. P., & Valsala, P. (2013). Feature Selection for high Dimensional DNA Microarray data using hybrid approaches. Bioinformation, 9(16), 824–828. https://doi.org/10.6026/97320630009824
8. G.T. Huang, I. Tsamardinos, V. Raghu, N. Kaminski, P.V. (2015) Benos T-RECS: stable selection of dynamically formed groups of features with application to prediction of clinical outcomes Pac. Symp. Biocomput., 20, 431-442
9. L. Kanal, B. Chandraskekaran (1971) On dimensionality and sample size in statistical pattern classification Pattern Recognit., 3, 225-234
10. Kumar, V. (2014). Feature Selection: A literature Review. The Smart Computing Review, 4(3). https://doi.org/10.6029/smartcr.2014.03.007
11. Drotár, P., Gazda, J., & Smékal, Z. (2015). An experimental comparison of feature selection methods on two-class biomedical datasets. Computers in Biology and Medicine, 66, 1–10. https://doi.org/10.1016/j.compbiomed.2015.08.010
12. Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024
13. K. Dunne, P. Cunningham, F. Azuaje (2002) Solutions to Instability Problems with Sequential Wrapper-Based Approaches to Feature Selection Tech rep, Trinity College
14. Basov, N. V., Adamovskaya, A. V., Rogachev, A. D., Gaisler, E. V., Demenkov, P. S., Ivanisenko, T. V., … Pokrovsky, A. G. (2025). Investigation of metabolic features of glioblastoma tissue and the peritumoral environment using targeted metabolomics screening by LC-MS/MS and gene network analysis. Vavilov Journal of Genetics and Breeding, 28(8), 882–896. https://doi.org/10.18699/vjgb-24-96
15. Liu, Z., Ma, P., Wang, Y., Matusik, W., & Tegmark, M. (2024). KAN 2.0: Kolmogorov-Arnold Networks Meet Science (Version 1). Version 1. arXiv. https://doi.org/10.48550/ARXIV.2408.10205
16. Guan, H., Yue, L., Yap, P.-T., Xiao, S., Bozoki, A., & Liu, M. (2023). Attention-Guided Autoencoder for Automated Progression Prediction of Subjective Cognitive Decline With Structural MRI. IEEE Journal of Biomedical and Health Informatics, 27(6), 2980–2989. https://doi.org/10.1109/jbhi.2023.3257081
17. Arin, P., Minniti, M., Murtinu, S., & Spagnolo, N. (2021). Inflection Points, Kinks, and Jumps: A Statistical Approach to Detecting Nonlinearities. Organizational Research Methods, 25(4), 786–814. https://doi.org/10.1177/10944281211058466
18. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
19. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/a:1010933404324
20. Chen, R.-C., Dewi, C., Huang, S.-W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00327-4