Linear ensemble model with winner-takes-all aggregation strategy for improved small data classification

Ivan Izonin[1,2,*,†], Roman Tkachenko[2,†], Serhii Chesanov[2,†], Yaroslav Tolstyak[3,4,†] and Myroslav Stupnytskyi[5,†]

[1]*The Bartlett School of Sustainable Construction, University College London, 1-19 Torrington Place, London WC1E 7HB, United Kingdom*

[2]*Lviv Polytechnic National University, S. Bandera str., 12, Lviv, 79013, Ukraine*

[3]*Lviv National Medical University named after Danylo Halytskyi, Pekarska str 69, 79010, Lviv, Ukraine*

[4]*Lviv Regional Clinical Hospital, Chernihivska srt 7, 79010, Lviv, Ukraine*

[5]*Military Medical Clinical Center of the Western Region, Anesthesiology and Intensive Care Department in the Clinic of Neurosurgery and Neurology, Lviv, Ukraine*

Abstract

Classification tasks involving small sample sizes remain particularly challenging and highly relevant in modern machine learning, especially within medical domains where data scarcity prevails. Classical classifiers and standard ensemble methods often perform poorly under such conditions due to overfitting, sensitivity to noise, and high computational complexity. This study addresses the problem of improving classification accuracy on small datasets by bridging the gap between high-performance ensemble learning and the robustness of linear modeling. In this work, we propose a new ensemble of linear machine learning algorithms that constructs an ensemble of linear regressors, where a separate binarized regressor is trained for each class. After normalizing the output of each weak regressor, the final prediction is obtained using a winner-takes-all aggregation strategy. The method is evaluated on a medical dataset containing records from 73 patients with polytrauma. Results show that the Ridge-based ensemble achieves the highest F_1-score on the test sets (92.8%), significantly outperforming both baseline linear models and widely used ensemble methods such as AdaBoost and Gradient Boosting. We conclude that the proposed ensemble classifier offers simplicity, high interpretability, low computational cost, and superior robustness to overfitting, making it uniquely suitable for small-data scenarios in medical, financial, and scientific applications.

Keywords

Classification, small data, ensembles method, linear model, interpretable artificial intelligence, machine learning, aggregation, winner-takes-all, weak predictors, regressors, neural-like structure, SGTM, Ridge

1. Introduction

In modern machine learning, classification with limited sample sizes remains one of the most relevant and challenging problems [1]. Many classical methods, including individual classifiers and ensemble models, demonstrate high performance on large datasets [2, 3]. However, their application to small datasets often leads to significant difficulties. These include an increased risk of overfitting, sensitivity to noise, and challenges in scaling models under limited computational resources [4]. As a result, the search for new approaches that can improve classification performance in small-data scenarios remains an important research direction in both machine learning and statistics.

One promising direction that has attracted considerable attention is the use of ensemble methods to improve classification performance on small datasets [5, 6]. Ensemble models combine the predictions of multiple classifiers, which typically enhances accuracy and robustness. Nevertheless, even ensembles

can face limitations such as computational overhead and the complexity of integrating outputs from multiple models, particularly when the training data is limited. To address these drawbacks, it is necessary to develop new aggregation strategies that preserve model simplicity and efficiency while achieving high classification accuracy.

Classification tasks involving small datasets, especially in medicine, finance, and the natural sciences, often encounter limited training samples, which complicates the construction of reliable models [7, 8]. Traditional ensemble methods such as one-vs-all, gradient boosting, and others do not always yield satisfactory results in such settings, primarily due to their sensitivity to noise and overfitting [9]. In these cases, it becomes essential to design methods that not only improve classification accuracy but also offer robustness and fast training, which is particularly important in low-data environments.

The objective of this study is to improve the effectiveness of classification in small-data scenarios by developing a linear ensemble model with data normalization and a winner-takes-all aggregation strategy.

The main contribution of this work lies in the development of a novel ensemble classifier based solely on linear regressors. Each model in the ensemble is trained using a one-vs-all strategy, where a separate training subset is created for each class with binary target labels. This approach enables the use of simple and interpretable models while achieving high accuracy through their aggregation. It combines the benefits of ensemble learning with the computational efficiency of linear modeling.

A key contribution of the proposed approach is the introduction of two-stage input normalization: first across columns (features), and then across rows (observations). This normalization ensures consistent feature scaling, preserves directional information, and increases the model's robustness to variability in input data. We also modify the aggregation mechanism by applying a winner-takes-all decision rule based on the maximum normalized prediction across all ensemble members. This allows for high classification consistency while maintaining minimal computational cost.

The practical value of the proposed ensemble lies in its universality and ease of adaptation to tasks with limited training data—particularly relevant in medical research and domains where model interpretability is critical. Unlike widely used nonlinear ensemble methods, the proposed approach offers a much simpler structure and lower computational complexity.

2. Related works

Ensemble machine learning methods have gained popularity due to their ability to combine weak models into a stronger one, thereby improving the accuracy and robustness of predictions. These methods help reduce overfitting and enhance the generalization capability of models, making them effective for solving complex classification tasks. However, several limitations may reduce their effectiveness when applied to specific types of data [10]. This section provides an overview of the main ensemble approaches, highlighting their strengths and weaknesses.

The One-vs-All method is one of the most commonly used strategies for multiclass classification problems [11]. It involves training a separate classifier for each class to distinguish it from all other classes. After training, the class with the highest probability is selected as the final prediction. Despite its simplicity and effectiveness, this method has several notable disadvantages. One major issue is that it does not account for inter-class relationships, which can lead to misclassifications, especially when classes have similar features [11].

Gradient Boosting is among the most powerful ensemble methods [12]. It builds models sequentially, where each subsequent model attempts to correct the errors of the previous one. This often leads to significantly improved classification accuracy [12]. However, the method has some limitations. First, it is sensitive to the scale of features—features with large values may dominate the learning process and negatively affect outcomes. To mitigate this, data normalization is typically applied before training, which adds extra preprocessing steps. Additionally, due to the number of models involved and the depth of decision trees used, Gradient Boosting can be computationally intensive, limiting its practicality for large or resource-constrained datasets.

AdaBoost is another widely used ensemble method from the boosting family [13]. It focuses on correcting the errors of previous models, with each new classifier giving more attention to observations that were misclassified in earlier iterations. However, AdaBoost is highly sensitive to noise in the data. Noisy instances may be overly emphasized during training, significantly degrading model performance.

Despite their overall effectiveness, ensemble methods such as One-vs-All, Gradient Boosting, and AdaBoost face significant limitations when applied to small datasets. One of the primary issues is sensitivity to feature scaling. In small-data scenarios, even minor variations in feature values can have a strong impact on model predictions, leading to unstable and inaccurate results. This increases the risk of overfitting, where the model performs well on the training data but fails to generalize to new, unseen data.

Another important challenge is the high computational demand, particularly problematic when data availability is limited. Both Gradient Boosting and AdaBoost require repeated model training based on residual errors, which increases training time and resource consumption—even on relatively small datasets [12]. This can pose a substantial barrier to their use in low-resource environments.

Handling noisy and imbalanced data is also a critical challenge. Small datasets may lack sufficient information for reliable classification, and even small errors or class imbalances can negatively affect model performance [11]. In methods such as One-vs-All, the lack of modeling of inter-class relationships can further reduce classification accuracy—especially when training data is limited. These limitations highlight the need for improving existing approaches or developing new, simpler, more interpretable, and computationally efficient ensemble methods that can better address the challenges of small-data classification.

3. Proposed ensemble of linear machine learning methods

The ensemble of linear machine learning methods developed in this study is based on the principle of combining weak learners to create a more powerful classifier. It is well established that ensemble techniques such as Adaptive Boosting [13] can effectively improve classification accuracy by aggregating weak models into a strong overall predictor [14]. A key feature of the method proposed in this work is the use of linear regressors instead of more complex models, which ensures simplicity, computational efficiency, and high interpretability of the results. Linear regressors produce output values within the range $[0, 1]$ for each class, which is typical in regression-based approaches, particularly in the context of binary classification problems [15].

The winner-takes-all aggregation strategy used to combine the outputs of all models in the ensemble is similar to the approach employed in Support Vector Machines (SVM), where the final class is chosen based on the highest predicted score among all models [16]. This strategy helps maintain classification accuracy and provides consistent results even in the presence of data variability.

The developed ensemble of linear models consists of a set of linear regressors, each trained on a binarized subset of the data corresponding to a specific target class. For each class, the original multiclass problem is converted into a regression task where samples belonging to the class are labeled as 1 (positive), and all others as 0 (negative). Each regressor then predicts a value in the $[0, 1]$ interval. As a result, the total number of models in the ensemble is equal to the number of classes in the classification task.

Once each regressor is trained, the predictions are aggregated using the winner-takes-all principle. This means that the final class assigned to a sample is the one with the highest predicted score among all regressors. In this way, the ensemble makes its final classification decision based on the relative strength of the individual model outputs, selecting the most probable class. The following sections present the key steps of the training and inference algorithms for the proposed ensemble in more detail.

3.1. Training algorithm

The training algorithm of the proposed ensemble involves the sequential execution of the following steps:

1. Splitting the available dataset A into training A_{train} and testing A_{test} subsets. All subsequent steps are performed solely on the training subset.

2. Creation of K separate training subsets, one for each of the K classes, $k = \overline{1, K}$. The number of subsets corresponds to the number of classes in the classification task. Each subset has the same size but a different target attribute y_i constructed as follows:

$$y_i = \begin{cases} 1, & \text{if } x_i \in C_k; \\ 0, & \text{if } x_i \notin C_k, \end{cases} \tag{1}$$

where x_i is an observation, C_k is the set of observations belonging to class k, and y_i is the target attribute (1 for class k, 0 for all other classes).

3. Normalization of each constructed training subset column-wise (feature-wise), independently from other features [17]:

$$x'_j = \frac{x_j}{\max(|x_j|)}, \tag{2}$$

where x_j denotes the values of the j-th feature for all observations, and $\max(|x_j|)$ is the maximum absolute value of the j-th feature among all observations. This approach preserves the sign of the features and ensures uniform scaling across all features.

4. Normalization of each training subset row-wise [17]:

$$x''_i = \frac{x'_i}{|x'_i|}, \tag{3}$$

where x'_i is the feature vector for the i-th observation normalized at step 3, and $|x'_i|$ is the norm (usually Euclidean) of the vector x'_i. Row-wise normalization accounts for the relationships between complex nonlinear dependencies among all attributes of the data vector. Alternative normalization methods at this step, e.g., as proposed in [18], can also consider the absolute value of features and increase the dimensionality of the input data space.

5. Training each corresponding training subset on its respective regressor in the ensemble. Each subset A_k is used to train the corresponding regressor R_k from the ensemble:

$$R_k \leftarrow Train(A_k), \tag{4}$$

where R_k is the regressor for class k, and $Train(A_k)$ is the training process on subset A_k.

6. Saving the minimum and maximum predicted values for each ensemble member. After each regressor R_k generates predictions for every observation x_i, the minimum and maximum predicted values are stored:

$$y_k^{min} = \min_k(R_k(x_i)), \quad y_k^{max} = \max_k(R_k(x_i)), \tag{5}$$

where y_k^{min} and y_k^{max} are the minimum and maximum predictions among all regressors for the specific observation x_i.

A simplified flowchart illustrating the key steps of the training algorithm (for the binary classification case) is shown in Figure 1.

3.2. Application algorithm

The application algorithm of the developed ensemble involves the sequential execution of the following steps:

1. Sequential normalization of the input vector u with an unknown output (target) attribute according to Equations (2) and (3).
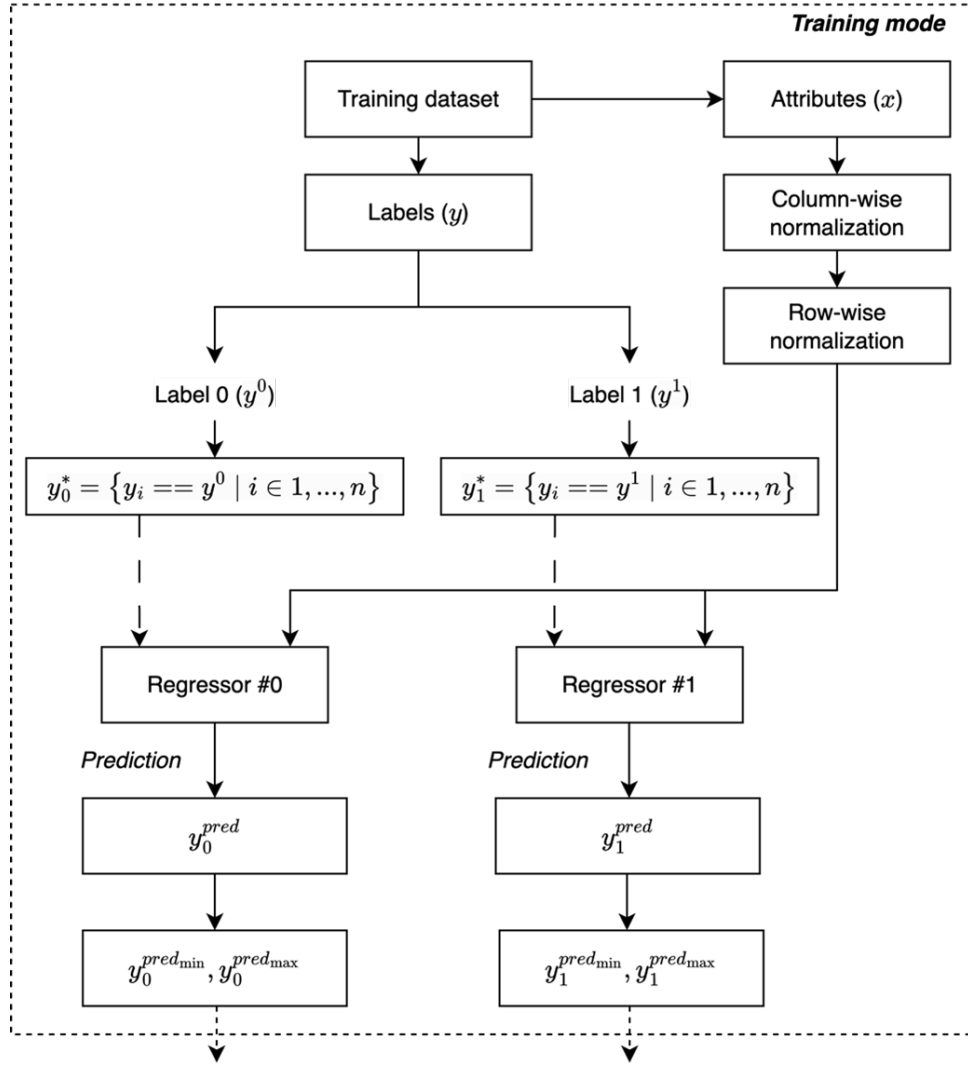
Figure 1: Flowchart for visualizing the key steps of the training algorithm, illustrating the parallel processing of class-specific regressors and normalization steps.

2. Applying the normalized vector u'' to each of the K, $k = \overline{1, K}$ pre-trained linear members of the ensemble:

$$y_k^{pred} = R_k(u''), \tag{6}$$

where y_k^{pred} is the prediction from the k-th regressor in the ensemble.

3. Normalization of the output signals y_k^{pred} from each ensemble member using the stored values y_k^{max} and y_k^{min} from Equation (5):

$$y_k^{result} = \frac{y_k^{pred} - -y_k^{min}}{y_k^{max} - -y_k^{min}}. \tag{7}$$

This approach eliminates dependence on the scale of the output data and makes the algorithm more robust and generalizable to new data.

4. Determination of the final class label for the current observation according to the "winner-takes-all" principle:

$$y^{final} = \arg \max_k y_k^{result}, \tag{8}$$

where y_k^{result} is the normalized prediction from the k-th regressor, and $\arg \max(\cdot)$ selects the index k corresponding to the maximum value among all y_k^{result}. As a result, the class with the
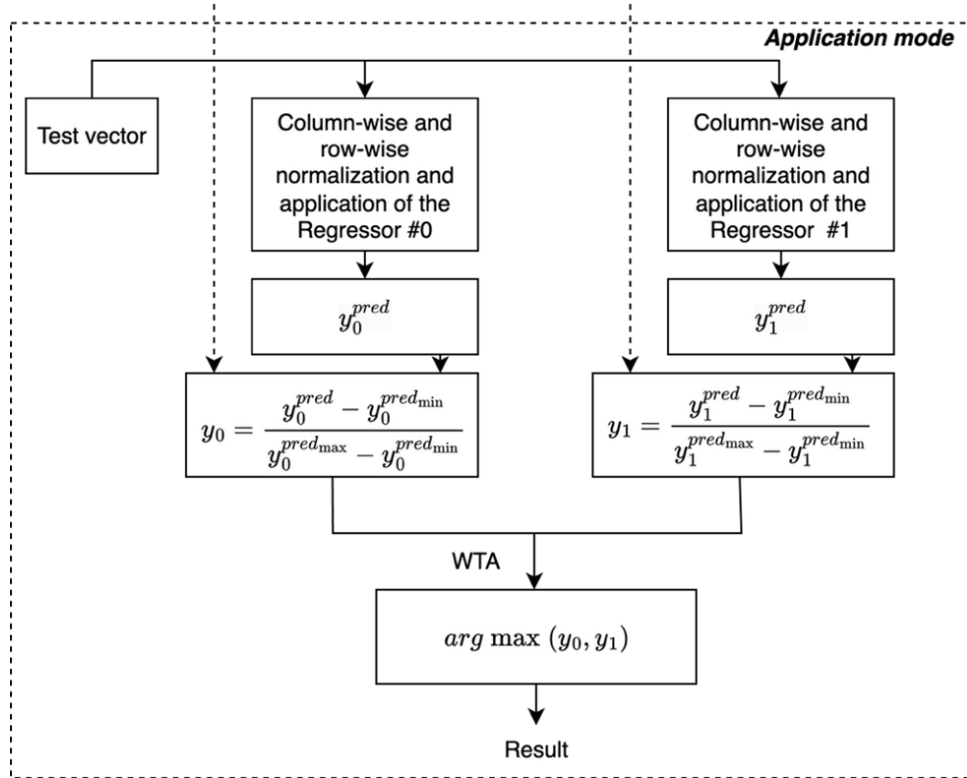
Figure 2: Flowchart for visualizing the key steps of the application algorithm, showing the normalization of new inputs, prediction generation, and the winner-takes-all aggregation.

highest normalized prediction value is chosen as the final classification for the current observation u.

A simplified flowchart illustrating the key steps of the application algorithm (for the binary classification case) is shown in Figure 2.

3.3. Dataset descriptions

For testing and validation of the developed method, modeling was conducted on an extremely small dataset [19] collected by specialists from the Department of Anesthesiology and Intensive Care at the Kharkiv City Clinical Hospital of Emergency and Urgent Medical Care named after Prof. O. I. Meshchaninov. The task consists of predicting mortality in patients with polytrauma based on 56 clinical and laboratory tests to assess the risk of death, optimize triage in intensive care settings, monitor treatment effectiveness, and support medical decision-making [20].

The dataset contains 73 samples of male patients hospitalized due to polytrauma. It includes 56 independent attributes covering a variety of clinical and laboratory indicators. The dependent variable is mortality, labeled as 1 (fatal case) and 0 (survival). Out of the total 73 cases, 31 were fatal, and 42 survived [19].

4. Modeling and results

4.1. Modeling

The modeling of the developed ensemble classifier based on linear machine learning methods was carried out using proprietary software implemented in Python 3.11. Additionally, the libraries NumPy, Pandas, scikit-learn, Matplotlib, and Seaborn were employed for machine learning implementation, data visualization, and processing. The synthesis and training of the ensemble classifier, as well as the

implementation of its individual components, linear regression-based regressors, were performed using custom modules adapted to support multi-class binarization and data normalization according to the steps described in Section 3.

To evaluate classification quality, five-fold stratified cross-validation was utilized, which accounts for class imbalance and reduces the risk of model overfitting. Each iteration of the cross-validation involved a new split into training and validation sets, maintaining the proportional class distribution (stratification).

For an objective analysis of classification performance during modeling, a diverse set of metrics was employed to assess the effectiveness of the ensemble classifier from multiple perspectives. In particular, for each model, the following metrics were calculated:

- Accuracy (for both training and test sets) – the overall classification accuracy;
- Precision – the proportion of correctly classified positive instances among all instances predicted as positive;
- Recall – the model's ability to identify all relevant instances of the target class;
- F_1-score – the harmonic mean of precision and recall, particularly important in tasks with class imbalance;
- Matthews Correlation Coefficient (MCC) – a balanced metric that takes into account true positives, true negatives, false positives, and false negatives;
- Cohen's Kappa – a measure of classification agreement that accounts for the probability of chance agreement.

The use of a comprehensive set of metrics, rather than relying solely on standard accuracy, ensures a thorough and statistically sound evaluation of the model's performance. In particular, the F_1-score, MCC, and Cohen's Kappa metrics are recommended for assessing models on imbalanced datasets, as they take into account all elements of the confusion matrix and are not prone to overestimating performance when one class dominates. This combination of metrics allows for more balanced and objective conclusions regarding the robustness, generalizability, and effectiveness of the developed ensemble classifier compared to existing methods.

4.2. Results

To quantitatively evaluate the performance of the proposed ensemble algorithms, a series of experiments were conducted using various linear machine learning methods as weak predictors within the developed ensemble framework:

- Algorithm 1 – proposed ensemble via SGTM neural-like structure.
- Algorithm 2 – proposed ensemble via SVM with linear kernel.
- Algorithm 3 – proposed ensemble via Ridge regression.

The comparative results of these three approaches are summarized in Table 1, which presents the mean values and standard deviations for each evaluation metric, obtained via 5-fold cross-validation to ensure the robustness of the results. Experiments were conducted in both training and testing modes, allowing for a comprehensive assessment of the models' ability to fit the training data and their generalization performance on unseen data.

5. Comparison and discussion

The evaluation of the proposed ensemble of linear machine learning methods was carried out in two stages. The first stage involved comparing three algorithmic implementations of the developed ensemble with the baseline linear regressors used as the foundation for training the ensemble. The baseline methods included: Ridge regression; Support Vector Regressor (SVR) with linear kernel; and Linear SGTM neural-like structure.

Table 1
Performance results of 3 ensemble algorithms in training and test modes using 5-fold cross-validation (average values and deviations).

| Performance indicator / Algorithm	Algorithm 1	Algorithm 2	Algorithm 3
Accuracy (train)	96.2 ± 2.1	93.7 ± 1.8	97.2 ± 2.1
Accuracy (test)	88.7 ± 8.6	91.5 ± 7.0	92.9 ± 6.4
Precision (train)	95.9 ± 2.1	93.4 ± 1.8	96.9 ± 2.2
Precision (test)	89.0 ± 8.8	91.4 ± 7.0	93.2 ± 6.1
Recall (train)	96.5 ± 2.0	94.3 ± 1.8	97.5 ± 2.1
Recall (test)	88.8 ± 8.3	92.2 ± 6.9	93.3 ± 6.1
F_1-Score (train)	96.1 ± 2.1	93.6 ± 1.8	97.2 ± 2.1
F_1-Score (test)	88.5 ± 8.6	91.5 ± 7.1	92.8 ± 6.5
MCC (train)	0.924 ± 0.04	0.876 ± 0.036	0.944 ± 0.042
MCC (test)	0.777 ± 0.17	0.836 ± 0.139	0.865 ± 0.122
Cohen Kappa (train)	0.922 ± 0.042	0.873 ± 0.036	0.943 ± 0.043
Cohen Kappa (test)	0.771 ± 0.171	0.83 ± 0.14	0.857 ± 0.128
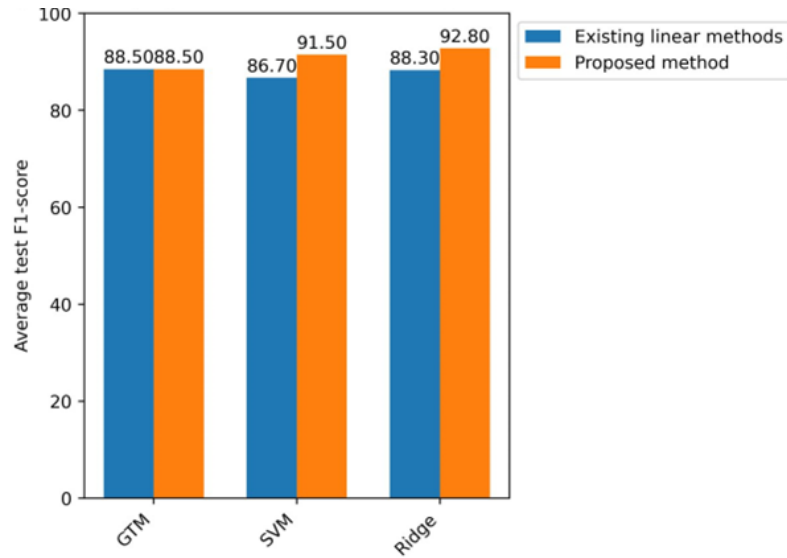
Figure 3: Average F_1-score comparison of 3 proposed linear ensemble algorithms against their existing baseline linear methods.

The results of this comparison are presented in Figure 3, showing the average F_1-scores on the test datasets. The F_1-score was chosen as the primary evaluation metric because, according to [21], it better reflects the balance between precision and recall than overall classification accuracy, especially in cases of imbalanced class distributions. This makes the F_1-score more suitable for correctly assessing the generalization quality of ensemble methods, where it is crucial to avoid the dominance of more frequent classes in the final outcome.

As shown by the results presented in Figure 3, using the linear SGTM neural-like structure as the base regressor for implementing the training procedures of the proposed ensemble (Algorithm 1) does not demonstrate an improvement in its overall accuracy. In contrast, for the SVR, applying the proposed ensemble led to an increase in the F_1-score by nearly 5%. This improvement can be attributed to the additional row-wise normalization, which better aligns the feature space for forming an optimal separating hyperplane.

For Ridge regression, which is a linear model with L2 regularization, the developed ensemble also showed a significant increase in the F_1-score by 4.5%. It is known that Ridge regression is sensitive to multicollinearity and feature scaling; therefore, the proposed preprocessing steps improve the stability,
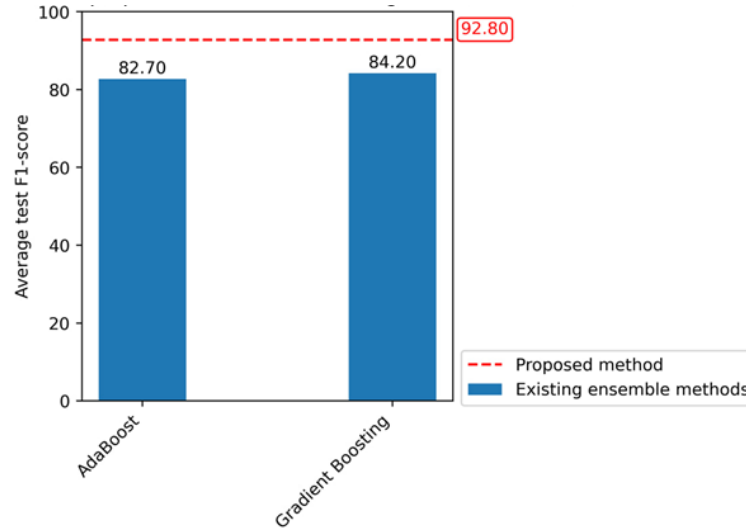
Figure 4: Average F_1-score of the best proposed algorithm (Algorithm 3) of the developed linear ensemble model vs. existing non-linear ensembles (AdaBoost and Gradient Boosting).

robustness, and accuracy of the entire ensemble.

In summary, it should be noted that the classifier based on the developed ensemble with individually trained regressors using target class binarization and two-stage feature normalization significantly enhances classification accuracy compared to existing linear methods.

The next stage of the comparison involved evaluating the performance of the proposed ensemble against well-known ensemble methods. For this purpose, Algorithm 3, as the best implementation of the developed ensemble classifier on the studied dataset, was compared with popular nonlinear ensemble methods such as AdaBoost and Gradient Boosting. The results of this comparison, based on the F_1-score for all methods in the application phase, are shown in Figure 4.

As shown in Figure 4, most existing methods achieve an F_1-score above 80%, but do not surpass higher thresholds. This performance level can be attributed to the limited size of the training dataset combined with a large number of features, which complicates building an effective model. In contrast, the proposed ensemble approach (Algorithm 3) reaches an F_1-score exceeding 90%, demonstrating superior generalization capabilities even with limited data. It is important to highlight that this ensemble is built upon linear models, which not only reduces computational complexity but also enhances interpretability compared to most existing nonlinear ensemble methods.

In summary, the following advantages of the developed ensemble of linear machine learning methods should be highlighted:

- Since the base model is a linear model, the ensemble can be easily adapted to various types of tasks. Traditional linear regressors as well as more complex or heuristic models can be used if necessary to improve classification accuracy.
- Linear regressors are less computationally intensive compared to other complex methods such as neural networks or decision trees. This significantly reduces memory and computational resource requirements, which is particularly important when processing large datasets or operating in real-time environments.
- Linear models are highly transparent, allowing one to understand exactly how each feature influences the final classification outcome. This is crucial when model interpretability is required, for example, in medical or financial applications where explanation of the model's decisions may be critical.
- By aggregating weak learners, the method achieves strong classification performance even with limited training data. The boosting-inspired approach effectively combines weak predictors into a powerful and accurate ensemble.

- Linear regressors are less prone to overfitting compared to more complex models, resulting in more stable and robust predictions, especially when the training data are scarce or noisy.

6. Conclusion

This paper addresses the challenge of classifying complications and predicting mortality based on medical data with a limited number of observations. The authors propose a novel ensemble classifier that combines linear regressors, class binarization, and a two-stage feature normalization process. The "winner takes all" aggregation strategy enhances the model's robustness to sample variability and improves overall decision accuracy.

The modeling was conducted on a real-world medical dataset characterized by an extremely small sample size and significant class imbalance. Comparative analysis was performed against baseline linear machine learning methods as well as well-known nonlinear ensemble algorithms. The results demonstrate that the proposed ensemble classifier significantly outperforms competing methods across multiple performance metrics, particularly in terms of the F_1-score. The highest performance was achieved using Ridge regression as the base learner within the ensemble. The achieved high classification accuracy ($F_1 > 90\%$) supports the applicability of the proposed method for medical diagnostic tasks under constrained data conditions.

Moreover, the developed ensemble exhibits several key advantages: ease of implementation, high interpretability, low computational complexity, and flexibility for adaptation to multi-class problems and novel data types. These findings have potential applications in building decision support systems in healthcare and related domains.

Acknowledgements

Declaration on Generative AI

During the preparation of this work, the authors used Chart-GPT-5 in order to Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] E. B. Hekler, P. Klasnja, G. Chevance, N. M. Golaszewski, D. Lewis, I. Sim, Why we need a small data paradigm, BMC Medicine 17 (2019) 133. doi:10.1186/s12916-019-1366-x.

[2] O. V. Kovalchuk, O. V. Barmak, Method of arrhythmia classification on ECG signal, Optoelectronic Information-Power Technologies 48 (2024) 34–44. doi:10.31649/1681-7893-2024-48-2-34-44.

[3] S. Popov, Large-scale data visualization with missing values, Technological and Economic Development of Economy 12 (2006) 44–49. doi:10.3846/13928619.2006.9637721.

[4] S. Popov, Nonlinear visualization of incomplete data sets, in: D. Grigoriev, J. Harrison, E. A. Hirsch (Eds.), Computer Science – Theory and Applications, volume 3967 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2006, pp. 524–533. doi:10.1007/11753728_53.

[5] Y. V. Bodyanskiy, O. K. Tyshchenko, A hybrid cascade neural network with ensembles of extended neo-fuzzy neurons and its deep learning, in: Information Technology, Systems Research, and Computational Physics, Springer, Cham, 2018, pp. 164–174. doi:10.1007/978-3-030-18058-4_13.

[6] N. Shakhovska, V. Yakovyna, V. Chopyak, A new hybrid ensemble machine-learning model for severity risk assessment and post-COVID prediction system, Mathematical Biosciences and Engineering 19 (2022) 6102–6123. doi:10.3934/mbe.2022285.

[7] D. Chumachenko, P. Piletskiy, M. Sukhorukova, T. Chumachenko, Predictive model of Lyme disease epidemic process using machine learning approach, Applied Sciences 12 (2022) 4282. doi:10.3390/app12094282.

[8] S. Subbotin, G. Tabunshchyk, P. Arras, D. Tabunshchyk, E. Trotsenko, Intelligent data analysis for individual hypertensia patient's state monitoring and prediction, in: 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2021, pp. 1–4. doi:10.1109/SIST50301.2021.9465989.

[9] I. Krak, O. Sobko, O. Mazurets, I. Tymofiiev, M. Molchanova, O. Barmak, Method for detecting and classifying cyberbullying in text content using neural networks, in: O. Lytvynov, V. Pavlikov, D. Krytskyi (Eds.), Integrated Computer Technologies in Mechanical Engineering - 2024, volume 1473 of *Lecture Notes in Networks and Systems*, Springer Nature Switzerland, Cham, 2025, pp. 486–498. doi:10.1007/978-3-031-94845-9_40.

[10] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Frontiers of Computer Science 14 (2020) 241–258. doi:10.1007/s11704-019-8208-z.

[11] X. Gao, Y. He, M. Zhang, X. Diao, X. Jing, B. Ren, W. Ji, A multiclass classification using one-versus-all approach with the differential partition sampling ensemble, Engineering Applications of Artificial Intelligence 97 (2021) 104034. doi:10.1016/j.engappai.2020.104034.

[12] I. D. Mienye, Y. Sun, A survey of ensemble learning: Concepts, algorithms, applications, and prospects, IEEE Access 10 (2022) 99129–99149. doi:10.1109/ACCESS.2022.3207287.

[13] K. W. Walker, Exploring adaptive boosting (AdaBoost) as a platform for the predictive modeling of tangible collection usage, The Journal of Academic Librarianship 47 (2021) 102450. doi:10.1016/j.acalib.2021.102450.

[14] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1997) 119–139. doi:10.1006/jcss.1997.1504.

[15] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, Springer Series in Statistics, Springer, New York, NY, 2009. doi:10.1007/978-0-387-84858-7.

[16] A. Tarr, K. Imai, Estimating average treatment effects with support vector machines, Statistics in Medicine 44 (2025) e70006. doi:10.1002/sim.70006.

[17] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, M. Gregus, C. Strauss, Towards data normalization task for the efficient mining of medical data, in: Proceedings of the 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), Ruzomberok, Slovakia, 2022, pp. 480–484. doi:10.1109/ACIT54803.2022.9913112.

[18] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, K. K. Singh, A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain, Mathematics 10 (2022) 1942. doi:10.3390/math10111942.

[19] M. Stupnytskyi, O. Biletskyi, Outcome prediction criteria for multiple trauma patients with combined cranio-thoracic injuries, European Journal of Clinical and Experimental Medicine 23 (2025) 110–116. doi:10.15584/ejcem.2025.1.17.

[20] I. Izonin, M. Stupnytskyi, R. Tkachenko, M. Havryliuk, O. Biletskyi, G. Melnyk, Mortality risk prediction for multiple trauma patients admitted to the hospital via machine learning algorithms, in: Z. Hu, F. Yanovsky, I. Dychka, M. He (Eds.), Advances in Computer Science for Engineering and Education VII, volume 242 of *Lecture Notes on Data Engineering and Communications Technologies*, Springer Nature Switzerland, Cham, 2025, pp. 218–228. doi:10.1007/978-3-031-84228-3_18.

[21] D. M. W. Powers, Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, 2020. arXiv:2010.16061.