UDC: 004.9 https://doi.org/10.15407/jai2025.02.096

I. Khamar[1], **I. Olenych**[2]
[1,2]Ivan Franko National University of Lviv, Ukraine
50, Dragomanov Street, Lviv, 79005
[1]ivan.khamar@lnu.edu.ua
[2]igor.olenych@lnu.edu.ua
[1]https://orcid.org/0009-0000-0514-903X
[2]https://orcid.org/0000-0002-6642-0222

ENHANCING REGRESSION FORECASTING WITH HYBRID ENSEMBLE–NEURAL NETWORK MODELS

Abstract. In regression forecasting problems based on large-scale and noisy datasets, there is often a need to choose between classical machine learning algorithms and modern neural network methods. Classical methods are simpler and more interpretable, while neural networks are better at handling heterogeneous and high-dimensional data, although they require more resources and more difficult fine-tuning. This paper presents a comparative analysis of the Random Forest (RF), XGBoosting, and Dense Neural Network (DNN) regression models for processing large tabular datasets. In particular, the IMDb dataset from the Kaggle platform was analyzed. Special attention was focused on studying the possibility of improving the performance of the prediction by combining RF and XGBoosting ensemble methods with DNN models.

It was found that the RF model demonstrated acceptable predictive quality, namely, a coefficient of determination (R^2) was 0.8640. The XGBoosting-based model showed a considerably better result, with an R^2 of 0.9245. The basic DNN model was characterized by the R^2 value of 0.8990. After optimizing the hyperparameters of the DNN model, the R^2 increased to 0.9179. A hybrid approach has been proposed as an additional way to improve the effectiveness of the DNN model. In particular, the distributions of features according to their impact on the prediction accuracy determined by the RF and XGBoosting methods were used as weighting coefficients for the DNN model feature vector. As a result, the most accurate forecast was obtained. The coefficients of determination R^2 were 0.9283 and 0.9302 for the RF-DNN and XGBoosting-DNN hybrid models, respectively. The obtained results can be used to develop predictive models based on heterogeneous and high-dimensional tabular data.

Keywords: forecasting, model efficiency, machine learning, ensemble methods, dense neural networks, feature engineering.

Introduction

The rapid growth of data volumes generated from many sources, such as various Internet services, financial and medical systems, Internet of Things sensors, etc., is increasing the need for effective methods of data processing, analysis, and forecasting. In particular, when it comes to regression forecasting based on large-scale and often noisy datasets, there is a need to choose between classical forecasting algorithms, optimized machine learning methods, or modern neural network approaches [1].

On the one hand, classical methods (e.g., linear regression and regression trees) and ensemble approaches such as Random Forest (RF) and XGBoosting are characterized by relative ease of implementation, interpretability, and resistance to overfitting [2-5]. At the same time, neural network models, particularly the Dense Neural Network (DNN), demonstrate great potential when handling high-dimensional, heterogeneous, and nonlinear data [6,7]. However, they often require significantly more computational resources, a more complex setup, and optimization [8].

In the context of regression problems on large data sets, some technical challenges arise, the solution of which determines the effectiveness of subsequent analysis. In particular, along with the existing advantages, the complexity of model tuning also increases. For example, one of the key challenges in using deep learning models is their tendency to overfit, high sensitivity to the choice of hyperparameters, and the need for significant computational resources [9]. Besides, a fine-tuning process is necessary, i.e., optimizing the weights of pre-trained networks by additional training on the target dataset to ensure high efficiency of the applied neural network training algorithms. Fine-tuning makes it possible to adapt the model architecture to the specifics of a particular task, reducing the generalization error. Transfer learning, layer

freezing, learning rate scheduling, dropout regularization, and L2-normalization can be highlighted as popular approaches to fine-tuning [10]. In particular, domain adaptation by fine-tuning DNN models in regression problems significantly increases accuracy, even with a limited training sample size [11]. When using large datasets such as IMDb or Kaggle Housing Dataset, neural network methods also provide advantages in accuracy compared to classical models, especially under conditions of high variability of input features [1].

Therefore, research into the effectiveness of neural network algorithms and the feasibility of using traditional approaches in real-world conditions is gaining practical importance. This is especially true in cases where there is a trade-off between accuracy, speed, and interpretability of solutions. Balancing accuracy and speed is especially important where system response time is critical. This applies, in particular, to online recommendations, dynamic pricing, financial transaction monitoring, and medical applications, where the speed of prediction can impact the quality of life for patients. In such conditions, optimizing the model architecture, reducing the input features (feature selection), and using incremental learning can significantly reduce resource consumption without losing the quality of the forecast [12].

In this paper, the effectiveness of the classical ensemble RF and XGBoosting models, as well as the DNN architecture for regression analysis of large amounts of data, is investigated through empirical comparison. Particular attention is focused on improving the performance of deep learning models through fine-tuning and a hybrid approach application. According to the hybrid approach, feature engineering that considers the feature impact, determined by the ensemble methods, on the model accuracy is proposed. The obtained results can improve the understanding of each approach's applicability limits and formulate recommendations for the choice of models in the practical problems of regression on big data.

Methods and means of implementation

The study is based on the IMDb Top 5000 Movies Dataset from the Kaggle open platform, which contains structured information about the most popular movies [13]. Each record in the dataset includes various characteristics, such as the movie title, start year, genre, information about the director and main actors, movie length, country of origin, language, production company, IMDb rating, number of votes, and gross revenue. The target variable for the regression problem is the IMDb rating, represented as a real continuous value in the 1.0–10.0 range. The resulting dataset consisted of 5000 complete records containing input characteristics and the target value. The dataset provides a diverse basis for investigating regression models on heterogeneous tabular data because it includes structured numerical values and categorical attributes that reflect movie metadata.

A classic scheme for dividing the initial data set was applied: 80 % of the total data was used for training models, and the remaining 20 % was used for testing to ensure objective comparison of models and reproducibility of experiments. Using the train_test_split function with the random_state=42 fixed parameter at the distribution eliminates random deviations in sample formation during re-runs. An additional 20 % of the training sample, which amounts to 16 % of the full dataset, was allocated to the validation sample to control the learning process, select the optimal number of epochs, and prevent overtraining of the DNN-based models.

Two different approaches to supervised learning, the ensemble RF and XGBoosting regressors, as well as DNN-based models, were studied. The models were implemented in Python using powerful open libraries, including OpenDatasets, Pandas, SKLearn, NumPy, Torch, and TensorFlow. The scikit-learn library, which supports parallel execution thanks to the n jobs parameter, was used to implement the ensemble models. The basic and optimized models based on DNN were implemented using the PyTorch and TensorFlow libraries, respectively. In

addition, regularization techniques, including Dropout, Batch Normalization, and Early Stopping, were applied to reduce overfitting and improve generalization ability [14].

Three common regression metrics, namely Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2), which reflects the degree of correspondence between the predicted values and the actual ones, were used to evaluate the performance of the models quantitatively. RMSE is sensitive to anomalies and serves as an important complement to MAE in detecting model instability. In general, the use of these metrics provides a balanced assessment of the prediction quality and generalization ability of the models.

RMSE, MAE, and R^2 were calculated based on a single test sample not used during model training. The chosen methodology for data distribution and model evaluation follows common machine learning practices to ensure the validity of the results [15,16]. A fixed seed avoids distortion of the results due to sample fluctuations, and the same test sample is a key condition for a fair comparison of models [17].

Data analysis and pre-processing

Exploratory data analysis was an important research stage that ensured the formation of a feature matrix for machine learning models. The analysis revealed four groups of features:

- identifiers and links (tconst, primaryTitle, IMDbLink, Title_IMDb_Link);
- numeric metadata (startYear, runtimeMinutes, numVotes, averageRating, rank);
- categorical features with low cardinality (directors, writers);
- categorical features with potentially high cardinality and multi-labels (genres).

The cardinality of categorical variables was assessed to convert them into numerical data. The features "directors" and "writers" had more than 2 and 4 thousand values, respectively. Therefore, these categorical features were encoded using median coding, as applying direct one-hot coding here would have resulted in the “curse of dimensionality” [18]. In this way, the qualitative "reputation" of the film's creators has become a quantitative compact feature with real prognostic content. The “genres” predictor contains about 5 thousand unique combinations, so a direct division into 23 atomic genres ('Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Family', 'Fantasy', 'Film-Noir', 'History', 'Horror', 'Music', 'Musical', 'Mystery', 'News', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War', 'Western') was employed for coding. In addition, the "PrimaryTitle" predictor was converted to two numeric attributes: the length of the title in characters and the number of words in the title.

The correlation matrix was constructed to identify linear relationships between input data. Fig. 1 illustrates the most statistically significant relationships between the data. As a result of data analysis, it was found that the key predictors are the reputation of the director and screenwriter. Besides, a moderate positive relationship was found between the movie rating and the number of votes, and a weak negative relationship with the start year. The duration and genre of the film can also be used as informative predictors.

Rank expectedly had a strong inverse correlation with the IMDb rating and was removed together with irrelevant attributes to avoid information noise and data leakage.

Results and discussion

The performance of non-optimized and optimized models was analyzed using various metrics to explore ways to improve the predictive models. Optimization of models based on the RF and XGBoosting methods includes using pre-prepared data, which reduces noise, collinearity, and the risk of data leakage, and setting the optimal parameters for each model to ensure the highest accuracy and stability of the results.

The results of testing the models based on ensemble methods for predicting the movie rating are presented in Table 1. It can be concluded that RF is a reasonably strong base model that has demonstrated high stability and good accuracy without complex tuning. The XGBoosting-based model showed

significantly better results according to the MAE, RMSE, and R^2 metrics.

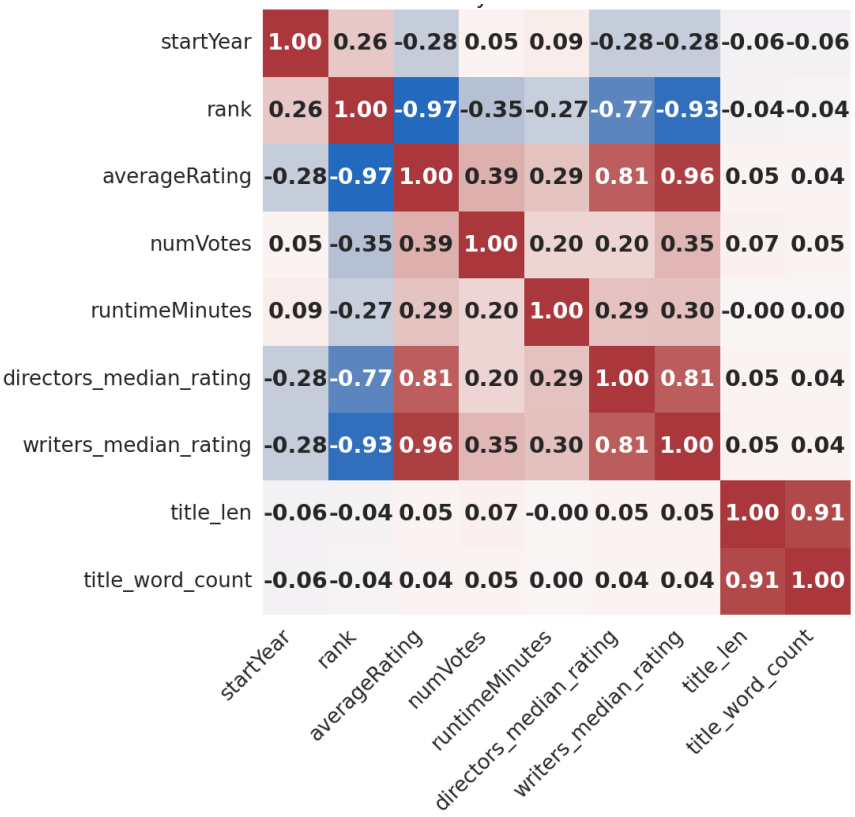

Fig. 1. Correlation matrix of the pre-prepared dataset

feature	importance
writers_median_rating	0.929276
numVotes	0.017492
directors_median_rating	0.014308
startYear	0.010163
runtimeMinutes	0.009439
title_len	0.005903
is_Documentary	0.002298
title_word_count	0.001951
is_Romance	0.001409
is_Crime	0.000749
is_Comedy	0.000712
is_Drama	0.000669
is_War	0.000632
is_Horror	0.000603
is_Action	0.000579
is_Music	0.000532
is_Sci-Fi	0.000480
is_Thriller	0.000460
is_Fantasy	0.000405
is_Mystery	0.000394
is_Adventure	0.000391
is_Western	0.000316
is_Biography	0.000298
is_History	0.000175
is_Family	0.000162
is_Musical	0.000093
is_Sport	0.000067

Fig. 2. The distribution of features according to their impact on the RF model accuracy

An important aspect of ensemble methods is their ability to estimate the influence of each feature on prediction accuracy. In particular, Fig. 2 and Fig. 3 show the normalized distribution of features on their importance for the developed movie rating prediction models. The resulting distributions will be used in the hybrid forecasting approach.

In general, the DNN-based models demonstrate high forecasting efficiency (see Table 1). However, the accuracy of the models significantly depends on the architecture and parameters of the DNN. The lowest accuracy was demonstrated by the basic model, whose architecture consisted of a limited number of layers and neurons (Fig. 4).

Table 1. The IMDb rating forecasting accuracy

Model	Metric		
	MAE	RMSE	R^2
RF	0.0916	0.2257	0.8640
XGBoosting	0.0870	0.1670	0.9245
Basic DNN	0.1311	0.1945	0.8990
Fine-tuned DNN	0.1162	0.1754	0.9179
RF-DNN	0.0828	0.1638	0.9283
XGBoosting-DNN	0.0883	0.1606	0.9302

feature	importance
writers_median_rating	0.578267
directors_median_rating	0.111363
is_Documentary	0.088144
is_Biography	0.029181
numVotes	0.018188
is_Drama	0.014412
is_Sci-Fi	0.012732
is_Romance	0.011661
startYear	0.010850
title_word_count	0.009344
is_Crime	0.008537
is_Horror	0.008363
is_Thriller	0.008007
is_Action	0.007930
is_Fantasy	0.007796
runtimeMinutes	0.007716
title_len	0.007382
is_Family	0.007318
is_Mystery	0.007031
is_Comedy	0.006956
is_Music	0.006605
is_Sport	0.006535
is_Adventure	0.005918
is_History	0.005392
is_War	0.005366
is_Musical	0.004678
is_Animation	0.003818
is_Western	0.000510

Fig. 3. The distribution of features according to their impact on the XGBoosting model accuracy

Baseline Dense NN (PyTorch) Params

	Layer (type)	Output Shape	Param #
0	Linear	(None, 128)	46,592
1	ReLU	(None, 128)	—
2	Linear	(None, 64)	8,256
3	ReLU	(None, 64)	—
4	Linear	(None, 1)	65
5			
6	Total: 54,785	Trainable: 54,785	Non-trainable: 0

Fig. 4. Basic DNN model parameters

To improve performance and provide more accurate predictions, the basic DNN model has been refined using hyperparameter optimization. A comprehensive approach to fine-tuning the structure and hyperparameters of models, which combines manual and automated settings, has been applied [19]. The optimal choice of parameters, including the number of layers, the number of neurons in each layer, types of activations, the optimizer, and the initial learning rate value, was determined through experimental selection on the validation sample. In particular, the number of layers was increased to 4 with a gradual decrease in neurons: 256 → 128 → 64 → 32 (Fig. 5). A larger number of layers and neurons makes it possible to capture non-linear interactions between many features.

In addition, a stochastic regularization technique with the parameter dropout = 0.2 and L2-normalization was applied. Using Batch Normalization has made learning faster, and LeakyReLU activation improved stability. Reducing the batch size (from 64 to 32) helps minimize error by updating the weights more frequently. The optimal learning rate = 0.001 was determined using the Adam optimizer.

Optimizing the architecture of the DNN-based model provides better forecasting results. In particular, a decrease in the MAE and RMSE values and an increase in the R^2 were found.

Tuned Dense NN (Keras) Params

	Layer (type)	Output Shape	Param #
0	Dense	(None, 256)	93,184
1	BatchNormalization	(None, 256)	1,024
2	Dropout	(None, 256)	—
3	Dense	(None, 128)	32,896
4	BatchNormalization	(None, 128)	512
5	Dropout	(None, 128)	—
6	Dense	(None, 64)	8,256
7	BatchNormalization	(None, 64)	256
8	Dense	(None, 1)	65
9			
10	Total: 406,789	Trainable: 135,297	Non-trainable: 896

Fig. 5. Fine-tuned DNN model parameters

The next step to improve the performance of the DNN-based model was to apply a hybrid approach. In particular, the feature importance values obtained by the RF and XGBoosting methods were used as weighting coefficients for the feature vector of the neural network models. As a result, a noticeable reduction in MAE and RMSE values was observed in the hybrid models, indicating a decrease in the number or absence of significant errors (see Table 1). The values of the coefficient of determination R^2 were 0.9283 and 0.9302 for the RF-DNN and XGBoosting-DNN models, respectively. The obtained results confirm the high potential of neural networks in regression problems with heterogeneous tabular data.

Conclusions

The paper studies the effectiveness of classical ensemble and neural network regression methods applied to tabular datasets. In particular, the performance of the RF, XGBoosting, and DNN models in default and additionally tuned configurations was compared using the IMDb dataset from the Kaggle platform. Besides, the possibility of combining ensemble methods with the DNN model to improve prediction performance was investigated.

The analysis highlighted the sensitivity of neural networks to architectural design and hyperparameter tuning. The improvement in DNN performance was achieved by increasing the number of layers, adjusting the batch size, and applying appropriate regularization. The hybrid RF-DNN and XGBoosting-DNN models, the feature vectors of which are weighted using the distribution of the importance of features, provide the highest accuracy of forecasting with the coefficient of determination R^2 of about 0.93.

References

1. Bjerre, L.M., Peixoto, C., Alkurd, R., Talarico, R., & Abielmona, R. (2024). Comparing AI/ML approaches and classical regression for predictive modeling using large population health databases: Applications to COVID-19 case prediction. *Global Epidemiology*, 8, 100168. https://doi.org/10.1016/j.gloepi.2024.100168
2. Ha, S., Park, J. & Jo, K, (2025) Comparative analysis of regression algorithms for drug response prediction using GDSC dataset. *BMC Res Notes*, 18 (Suppl 1), 10. https://doi.org/10.1186/s13104-024-07026-w
3. Olenych, I., Demchyk, D., Babiak, S., Futey O. (2025). Air pollution prediction using machine learning. Artificial Intelligence, No 1(102), 141–146. https://doi.org/10.15407/jai2025.01.141
4. Shivashankar, S. K., Prajwal, M. D., Likith Raj, K. R., Priyadarshini, T. A. R., & Manvitha, S. M. (2024). Forest fire prediction using random forest regressor: A comprehensive machine learning approach. *International Journal of Innovative Science and Research Technology*, 9(9), 2063–2071. https://doi.org/10.38124/ijisrt/IJISRT24SEP1290
5. Sharma, A.K., Li, LH., & Ahmad, R. (2023). Default Risk Prediction Using Random Forest and XGBoosting Classifier. 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications. Smart Innovation, Systems and Technologies, 314. https://doi.org/10.1007/978-3-031-05491-4_10
6. Bhattacharya, S., Liu, Z., & Maiti, T. (2024). Comprehensive study of variational Bayes classification for dense deep neural networks. *Statistics and Computing*, 34, 17. https://doi.org/10.1007/s11222-023-10338-9
7. Hegde, R. S. (2019). Deep neural network (DNN) surrogate models for the accelerated design of optical devices and systems: moving beyond fully-connected feed forward architectures. *Proc. SPIE,* 11105, 1110508. https://doi.org/10.1117/12.2528380
8. Elsayed, A., Levison, J., Binns, A., Larocque, M., & Goel, P. (2025). Regression-based machine learning models for nitrate and chloride prediction in surface water in a small agricultural sand plain sub-watershed in southwestern Ontario, Canada. *Front. Environ. Sci.* 13, 1543852. https://doi.org/10.3389/fenvs.2025.1543852
9. Li, H., Rajbahadur, G. K., Lin, D., Bezemer, C.-P., & Jiang, Z. M. (2024). Keeping deep learning models in check: A history-based approach to mitigate overfitting. *IEEE Access*, 12, 70676–70689. https://doi.org/10.1109/ACCESS.2024.3402543
10. Ha, S., Jeong, S., & Lee, J. (2024). Domain-aware fine-tuning: Enhancing neural network adaptability. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11), 12261–12269. https://doi.org/10.1609/aaai.v38i11.29116
11. Chen, C.-H., Lai, J.-P., Chang, Y.-M., Lai, C.-J., & Pai, P.-F. (2023). A Study of Optimization in Deep Neural Networks for Regression. *Electronics*, 12(14), 3071. https://doi.org/10.3390/electronics12143071
12. Khan, M. A., Azim, A., Liscano, R., Smith, K., Chang, Y.-K., Seferi, G., & Tauseef Q. (2024). On the effectiveness of feature selection techniques in the context of ML-based regression test prioritization. *IEEE Access*, 12, 131556–131575. https://doi.org/10.1109/ACCESS.2024.3459656
13. IMDb Top 5000 Movies [Electronic resource]. - Mode of access: https://www.kaggle.com/datasets/tiagoadrianunes/imd

b-top-5000-movies/data

14. Xu, Y. (2025). Deep regularization techniques for improving robustness in noisy record linkage task. *Advances in Engineering Innovation*, 15, 9–13. https://doi.org/10.54254/2977-3903/2025.20435

15. Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., & Goldstein, T. (2021). SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. *Advances in Neural Information Processing Systems*, 34, 11237–11250. https://doi.org/10.48550/arXiv.2106.01342

16. Zhang, Y., Xiong, F., Xie, Y., Fan, X., & Gu H. (2020). The impact of artificial intelligence and blockchain on the accounting profession. *IEEE Access*, 8, 110461–110477. https://doi.org/10.1109/ACCESS.2020.3000505

17. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

18. Altman, N., & Krzywinski, M. (2018). The curse(s) of dimensionality. Nature Methods, 15, 399–400. https://doi.org/10.1038/s41592-018-0019-x

19. Tiep, N. H., Jeong, H.-Y., Kim, K.-D., Xuan Mung, N., Dao, N.-N., Tran, H.-N., Hoang, V.-K., Ngoc Anh, N., & Vu, M. T. (2024). A New Hyperparameter Tuning Framework for Regression Tasks in Deep Neural Network: Combined-Sampling Algorithm to Search the Optimized Hyperparameters. *Mathematics*, 12(24), 3892. https://doi.org/10.3390/math12243892

The article has been sent to the editors 31.05.25.
After processing 10.06.25.
Submitted for printing 30.06.25.