DOI: https://doi.org/10.15276/ict.02.2025.07
UDC 004

Application of a novel TabPFN model for tabular data classification

Andrew L. Mrykhin[1)]
Postgraduate Student of the Department of Information Systems
ORCID: https://orcid.org/0009-0009-1545-632X; amrykhin@gmail.com
Svitlana G. Antoshchuk[1)]
Doctor of Engineering Sciences, Professor of the Department of Information Systems
ORCID: https://orcid.org/0000-0002-9346-145X; asg@op.edu.ua. Scopus Author ID: 8393582500
[1)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

As tabular data remains the most commonly used form of data—ubiquitous across numerous fields such as medicine, finance, manufacturing, economics, public governance, and climate science—the problem of developing new methods for the classification and regression analysis of tabular datasets remains highly relevant. Although deep learning has revolutionized learning from raw data in domains like computer vision and natural language processing, tabular data presents a unique set of challenges that prevent conventional neural network–based models from being immediately effective. In our study, we examine the novel TabPFN v2 (Tabular Prior-Data Fitted Network) model developed by Prior Labs, which promises highly accurate predictions on small- to medium-sized datasets without extensive tuning or data preprocessing. TabPFN is a generative, transformer-based foundation model that leverages the same mechanisms that have driven the remarkable success of large language models to produce a powerful tabular prediction algorithm. It is pre-trained on a large corpus of diverse synthetic tabular datasets and employs in-context learning with a bidirectional attention mechanism to address key limitations of existing deep learning models when analyzing row–column–structured data. Applying TabPFN to a real-world task of classifying supply records for risk assessment, we found that, when used within its specified limits, this model can outperform established state-of-the-art gradient-boosted decision tree models. We also explored the optimization options available in TabPFN and conducted experiments using our real-world data. Overall TabPFN is a powerful example of how transformer model principles can be adapted to row-column organized data. While not being a one-size-fits-all solution, TabPFN is certainly worth including in the toolkit for tabular data analysis.

Keywords: Tabular data; machine learning; classification; regression; gradient boosting decision trees; generative transformer model; in-context learning; two-way attention mechanism

1. THE TABULAR DATA CHALLENGE

Tabular data is the row-column organised values, the kind we see every day in relational databases and spreadsheets. And it is the most commonly used form of data, ubiquitous in a huge number of applications in medicine, finance, manufacturing, economics, public governance, climate science etc. [1].

In the last decade we often celebrate the successes of Deep Learning in domains like vision and language, but tabular data presents a unique set of challenges that prevent those models from being immediately effective [2].

Firstly, tabular data is inherently heterogeneous. We're dealing with a mix of data types – categorical values, dense numerical features, and sparse IDs—all living side-by-side.

Secondly, there's an inherent lack of spatial or temporal structure. Classic deep learning models, like CNNs and standard RNNs, are designed for data with strong local correlation, like images or text sequences. But tabular data is different. It's unstructured in order – meaning the column order doesn't usually matter. A model must learn relationships independent of that column order.

Finally, while the dimensionality (number of features) can be high, the datasets (the number of samples) are often small to medium-sized when compared to typical DL tasks. This scarcity of data makes training large, conventional Deep Learning models prone to catastrophic overfitting.

2. EXISTING SOLUTIONS: THE GRADIENT BOOSTING REIGN

For the past two decades, the tabular data analysis domain is dominated by Gradient Boosting Machines (GBMs) or other name – Gradient Boosted Decision Trees (GBDTs) methods, such as XGBoost, LightGBM, or CatBoost [2].

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

GBMs are ensemble models that sequentially combine hundreds or even thousands of simple, weak decision trees. The mechanism relies on an additive correction principle when the first tree makes an initial prediction, then subsequent trees are trained specifically to predict and correct the errors (or residuals) made by the cumulative ensemble of all previous trees. This iterative, boosting process ensures the model focuses its learning effort precisely on the most difficult data points, rapidly minimizing the overall error.

Such architecture grants next powerful capabilities:

– Handling Non-Linearity and Feature Interaction: decision trees can naturally capture complex, non-linear relationships and interactions between features without requiring manual feature transformations. The ability to partition the feature space recursively allows the model to learn localized patterns that linear models entirely miss;

– Robustness to Heterogeneity: GBDTs inherently handle mixed data types and are highly robust to feature scaling and outliers. Unlike neural networks, they don't require extensive normalization or encoding, significantly simplifying the preprocessing pipeline.

Most popular modern implementations of GBMs are:

– XGBoost (Extreme Gradient Boosting): Mature and widespread implementation, known for its focus on robustness and regularization, offering control mechanisms to prevent overfitting. It has been the dominant choice in competitive data science for years [7];

– LightGBM (Light Gradient Boosting Machine): LightGBM is distinguished by its efficiency and speed, especially on large datasets. It uses a novel leaf-wise tree growth strategy that prioritizes error reduction, making it significantly faster than XGBoost in many scenarios [8];

– CatBoost (Categorical Boosting): CatBoost’s primary advantage is its native handling of categorical features. It employs specialized ordered target encoding and permutation-driven training to avoid target leakage, allowing it to often perform well with minimal preprocessing.

However they discussed TabPFN model is not the first attempt to apply DL to tabular data. Quite a number of models were proposed starting from Multi-Layer Perceptrons with specifically tuned regularizations, Feature tokenization models like TabNet, Neural Additive Models (NAMs), models emulating gradient boosting – Neural Oblivious Decision Ensembles (NODE). While innovative, these models often still required extensive training time, heavy hyperparameter tuning, and large amounts of data to achieve parity with, let alone surpass, highly-tuned GBDTs [3], [4].

The concept of a pre-trained transformer model was also explored, e.g. ExcelFormer [5] model and the earlier v1 version of TabPFN itself. However, these previous attempts also largely failed to dethrone GBDTs.

3. TABPFN OVERVIEW

The model examined in our study – TabPFN v2 or Tabular Prior-Data Fitted Network was released last summer by Prior Labs with the goal of providing a solution that can perform accurate classification or regression on small to medium-sized tabular datasets without dataset-specific training or tuning. At its core, TabPFN v2 applies the idea of in-context learning, similar to what we see in large language models. Instead of learning a fixed mapping from features to outcomes for one dataset, it learns how to learn from examples [6].

The key idea behind TabPFN is to generate a large corpus of synthetic tabular datasets and then train a transformer-based neural network to learn to solve these synthetic prediction tasks. Although traditional approaches require hand-engineered solutions for data challenges such as missing values, TabPFN autonomously learns effective strategies by solving synthetic tasks that include these challenges.

TabPFN addresses two key limitations inherent to use of transformer-based models with tabular data. First, as transformers are designed for sequences, they treat the input data as a single sequence, not using the tabular structure. Second, machine learning models are often used in a fit-predict model, in which a model is fitted on the training set once and then reused for multiple test datasets. Transformer-based ICL algorithms, however, receive train and test data in a single pass and thus

perform training and prediction at once. Thus, when a fitted model is reused, it has to redo computations for the training set.

To better use the tabular structure, TabPFN authors proposed an architecture that uses a two-way attention mechanism, with each cell attending to the other features in its row (that is, its sample) and then attending to the same feature across its column (that is, all other samples). This design enables the architecture to be invariant to the order of both samples and features and enables more efficient training and extrapolation to larger tables than those encountered during training.

To mitigate repeating computations on the training set for each test sample in a fit-predict setting, the model can separate the inference on the training and test samples. This allows performing ICL on the training set once, save the resulting state and reusing it for multiple test set inferences.

The result is a model that works remarkably well on datasets up to around ten thousand samples and a few hundred features, often outperforming widely used tree-based methods such as CatBoost or XGBoost, and doing so in seconds rather than hours of training. It is especially strong when the data is heterogeneous or when users want to avoid lengthy feature engineering and hyperparameter optimization.

However, TabPFN v2 is not a universal replacement for traditional models. Because it relies on transformer attention over all input samples, its computational cost grows quadratically with dataset size. Performance begins to degrade on very large datasets or when the number of features exceeds several hundred. It also struggles when the number of classes is high or when the data distribution changes significantly from what the model was exposed to during pretraining.

4. EMPIRICAL EXAMPLE: SUPPLY RISK CLASSIFICATION

We have performed empirical evaluation of TabPFN performance in a real-world multi-class classification task involving supply chain risk assessment.

The risk evaluation task we would use in our case study is a part of the Model for assessing the impact of tactical material procurement risks on order fulfilment in Make-To-Order Manufacturing presented recently by the authors at applied information systems and technologies in the digital society conference.

The dataset we use as the base for our classification consists of material procurement records extracted from the ERP system of the customer. The initial number of samples is 3381, overall 39 features were extracted, 7 of them were selected as predictors at exploratory data analysis phase 5 of selected predictors are categorials with cardinality from 3 to 698.

The aim of our task is to predict on-time delivery or shipping delays for manufacturing materials. We identified four target classes representing In-time delivery and three levels of delay: short, moderate, and long. The sample distribution across target classes is highly imbalanced, with the IN_TIME delivery class being dominant.

Overall, due to its limited number of samples, the predominance of categorical features, and the imbalance of target classes, our dataset promises to be suitable but quite challenging for both the TabPFN and GB models.

The first model to examine is the proved baseline in the domain of GBMs – XGBoost [7].

We tried both one-hot plus target encodings and native handling of categorial features, with later delivering the better results. We have configured early stopping to control overfitting. And we use class weighting to counter target classes imbalance. The SMOTE was also tested, but it produced worse results than weighting. At last we applied randomized search to improve the hyper parameters set.

The XGBoost model achieved a usable overall accuracy of 73 % and just above average 61 % Macro-F1 score. Corresponding confusion matrix is presented in Fig. 1.

The second model to test is LightGBM that is widely acknowledged as a more modern and optimized implementation of GBDTs [8]. LightGBM is highly reputed for its native efficient handling of categorical data. And we applied early stopping and class weighting to our model as well as randomized search on hyper parameters.

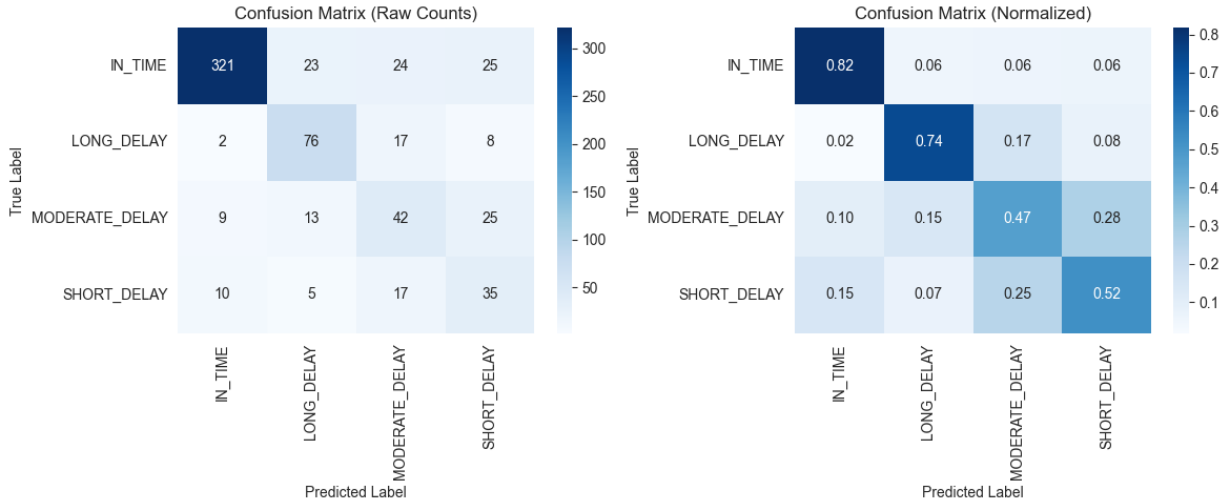

Fig 1. **Confusion matrix for XGBoost classification results**

The LightGBM model yielded the better scores, particularly on the critical minority classes. The model showed a significant 6 percentage point Macro-F1 improvement over XGBoost, reaching a stable 67 % Macro-F1 Score with general accuracy of 77 %. Corresponding confusion matrix is presented in Fig. 2.

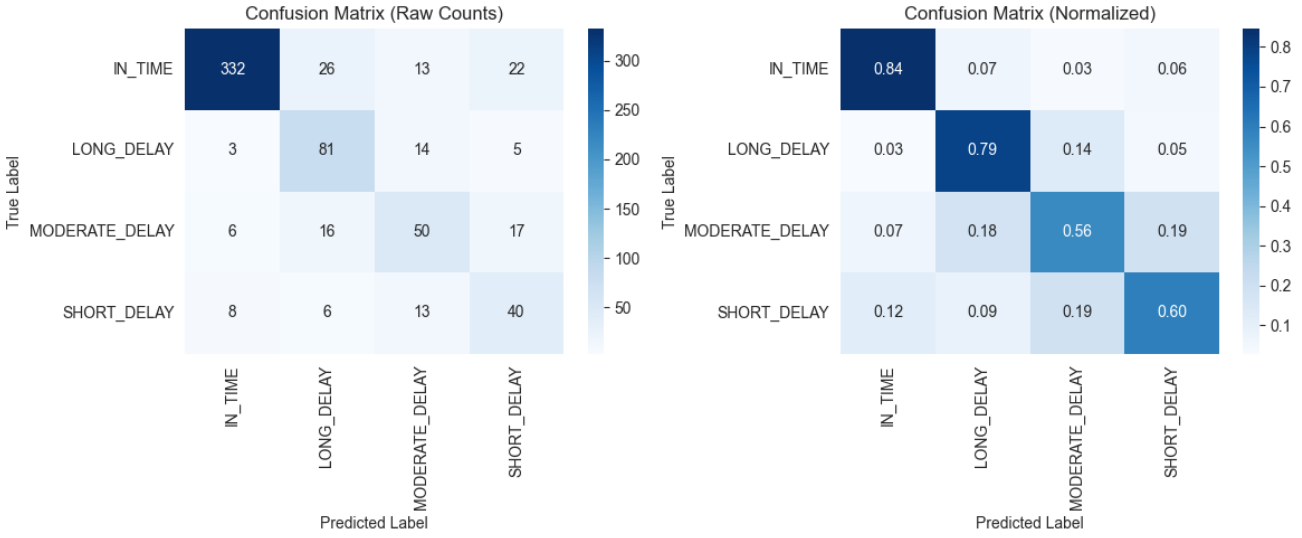

Fig 2. **Confusion matrix for LightGBM classification results**

And the last is our title model – the TabPFN. TabPFN can handle categorial features and target classes imbalance natively and is internally protected from overfitting.

TabPFN achieved a 79 % Accuracy and a new high of 72 % Macro-F1 Score. Corresponding confusion matrix is presented in Fig. 3. These are the best classification results of all models reviewed, with TabPFN actually outperforming gradient-boosting models in our case.

5. TABPFN OPTIMIZATION

TabPFN authors stress that their model provides strong performance out of the box without extensive tuning. Also since TabPFN is a pretrained transformer model it doesn't have common hyperparameters like rate, depth, tree splitting or sampling options.

Nevertheless several tuning options are available for TabPFN.

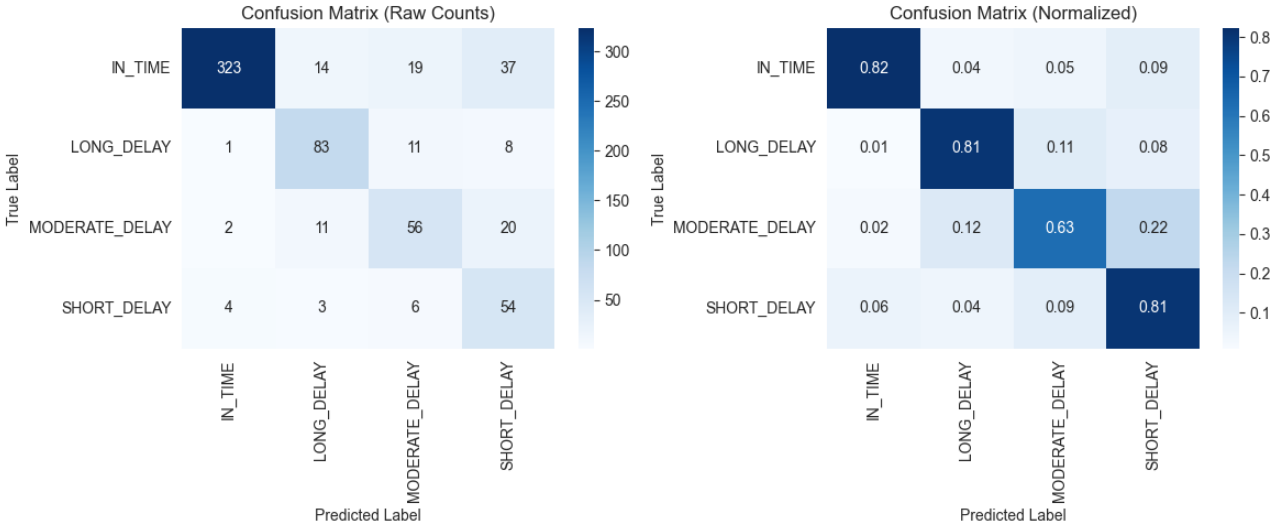

Fig. 3. **Confusion matrix for TabPFN classification results**

First is TunedTabPFN class that provides automatic tuning capabilities using Bayesian optimization of both model and preprocessing parameters? Parameters tuned include model subtype, pretrained data checkpoint selection, number of estimators and so on as well as preprocessing options like scaling, encoding, outliers handling, etc.

Next is the AutoTabPFN class that implements Portfolio Hyperparameter Ensembling (PHE) – the AutoML – style pipeline-level tuning.

AutoTabPFN creates a post-hoc ensemble of multiple TabPFN configurations using AutoGluon. It randomly samples many preprocessing + model configuration combinations, evaluates each on a validation split. Then selects a portfolio (subset) of the best-performing configurations and creates a post-hoc ensemble of multiple TabPFN configurations, and finally combines them via greedy ensemble selection (weighted averaging).

We have applied AutoTabPFN to our supply dataset classification problem varying max_time and eval_metric parameters. Best matching our risk assessment task results were obtained with eval_metric set to “recall_macro”, we reached 74 % Macro-F1 Score with general accuracy of 81%. Corresponding confusion matrix is presented in Fig. 4.

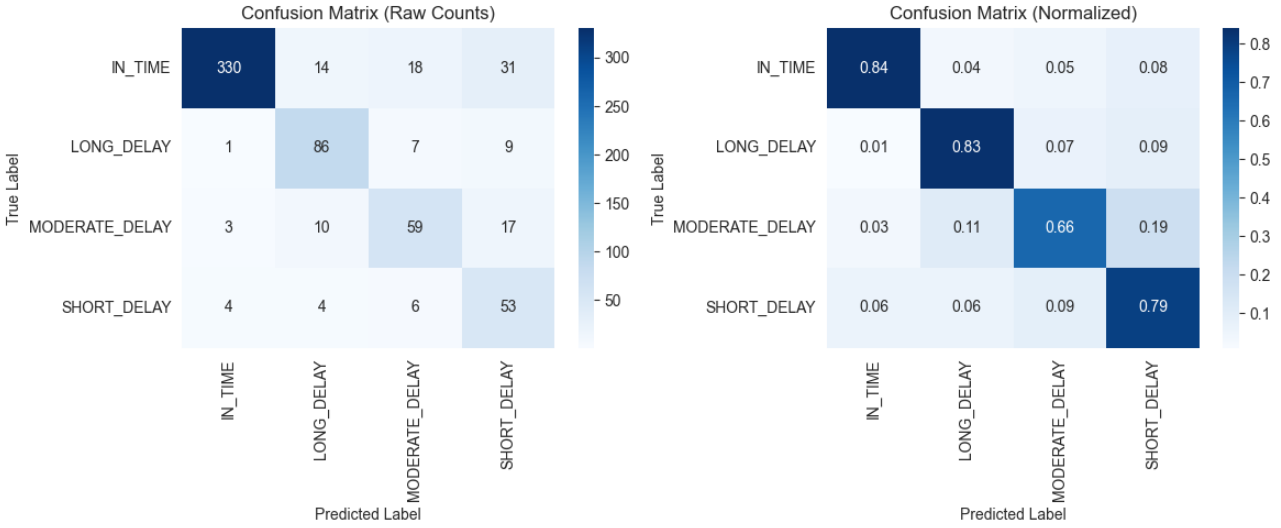

Fig. 4. **Confusion matrix for AutoTabPFN PHE optimized classification results**

As we can see, tuning delivered marginal improvements of prediction accuracy in critical target classes.

At last there is an option for refining the pre-trained weights of the transformer model (fine-tuning TabPFN in model author's terminology). During fine-tuning Adam optimizer is being used to adapt the general, meta-learned "algorithm" inside the transformer to the quirks of user specific dataset, making the internal learned relationships between data points more precise.

Fine-tuning is especially effective on datasets that exceed recommended for model size and require subsampling. In that case the model can be iteratively refined at every split.

6. CONCLUSION

We have obtained empirical evidence that the TabPFN model really achieves performance levels claimed by authors and when applied within the specified limits it can outperform established leaders in the class. While it is definitely not a one-size-fits-all solution and scalability or interpretability issues may hamper its performance on some datasets, TabPFN is certainly worth including into a tabular data analysis toolkit.

REFERENCES

1. Borisov V., Leemann T., Seßler K., Haug J. "Pawelczyk M., Kasneci G. "Deep Neural Networks and Tabular Data: A Survey". *IEEE Transactions on Neural Networks and Learning Systems*. 2024; 35 (6): 7499–7519. DOI: https://doi.org/10.1109/TNNLS.2022.3229161.

2. Shwartz-Ziv R., Armon A. "Tabular data: Deep learning is not all you need." *Information Fusion.* 2022; 81: 84–90. DOI: https://doi.org/10.48550/arXiv.2106.03253.

3. Kadra A., Lindauer M., Hutter F., Grabocka J. "Well-tuned simple nets excel on tabular datasets". *Advances in Neural Information Processing Systems*. 2021; 34: 23928–23941. DOI: https://doi.org/10.48550/arXiv.2106.11189.

4. Ruan Y., Lan X., Ma J., Dong Y., He K., Feng M. "Language modelling on tabular data: A survey of foundations, techniques and evolution". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2408.10548.

5. Chen J., Yan J., Chen Q., Chen D., Wu J., Sun, J. "Excelformer: A neural network surpassing GDBTs on tabular data". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2301.02819.

6. Hollmann N., Müller S., Purucker L., et al. "Accurate predictions on small data with a tabular foundation model". *Nature*. 2025; 637: 319–326. DOI: https://doi.org/10.1038/s41586-024-08328-6.

7. Chen T., Guestrin C. "Xgboost: A scalable tree boosting system". In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM Press).* 2016. p. 785–794. DOI: https://doi.org/10.48550/arXiv.1603.02754.

8. Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., Liu T. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". *Advances in Neural Information Processing Systems (Curran Associates, Inc.).* 2017; 30: 3146–3154. DOI: https://dl.acm.org/doi/10.5555/3294996.3295074.

DOI: https://doi.org/10.15276/ict.02.2025.07
УДК 004

Застосування новітньої моделі TabPFN для класифікації табличних даних

Мрихін Андрій Львович[1)]
Аспірант каф. Інформаційних систем
ORCID: https://orcid.org/0009-0009-1545-632X; amrykhin@gmail.com
Антощук Світлана Григорівна[1)]
Д-р техніч. наук, професор каф. Інформаційних систем
ORCID: https://orcid.org/0000-0002-9346-145X; asg@op.edu.ua. Scopus Author ID: 8393582500
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

На сьогодні табличні дані залишаються найпоширенішою формою представлення інформації — вони повсюдно використовуються в таких галузях, як медицина, фінанси, виробництво, економіка, державне управління та кліматологія. Тому проблема розроблення нових методів класифікації та регресійного аналізу табличних наборів даних залишається надзвичайно актуальною. Хоча методи глибокого навчання здійснили справжню революцію в аналізі вихідних даних у таких сферах, як комп'ютерний зір і обробка природної мови, табличні дані становлять унікальний набір викликів, що не дозволяє традиційним нейромережевим моделям бути безпосередньо ефективними. У нашому дослідженні розглядається новітня модель TabPFN v2 (Tabular Prior-Data Fitted Network), розроблена компанією Prior Labs, яка обіцяє забезпечити високу точність прогнозування на малих і середніх вибірках без потреби в трудомісткому налаштуванні гіперпараметрів і попередній обробці даних. TabPFN є генеративною моделлю - трансформером, що використовує ті самі механізми, що забезпечили видатні успіхі великих мовних моделей, для створення потужного алгоритму прогнозування табличних даних. Модель попередньо навчена на великому корпусі різноманітних синтетичних табличних наборів даних і застосовує навчання в контексті (in-context learning) з двонаправленим механізмом уваги для подолання ключових обмежень існуючих моделей глибокого навчання під час аналізу даних, організованих у вигляді рядків і стовпців. Застосовуючи TabPFN до реального завдання класифікації записів про постачання виробничих матеріалів для оцінювання ризиків, ми з'ясували, що за умови використання в межах її визначених обмежень ця модель може перевершувати визнані найсучасніші рішення, засновані на градієнтному бустингу над деревами рішень. Ми також дослідили доступні в TabPFN можливості оптимізації та провели експерименти з нашими реальними даними. Загалом, TabPFN є яскравим прикладом того, як принципи моделей-трансформерів можуть бути успішно адаптовані для аналізу табличних даних. Хоча TabPFN не є універсальним рішенням, вона безперечно варта того, щоб бути включеною до інструментарію аналізу табличних даних.

Ключові слова: табличні дані; машинне навчання; класифікація; регресія; градієнтний бустинг над деревами рішень; генеративна модель-трансформер; навчання в контексті; двонаправлений механізм уваги