Project Report

Tan-Thinh Duong
Department of Information Technology
FPT University
Ho Chi Minh City, Vietnam
tanthinhdt24@gmail.com

***Abstract*—This report focuses on optimizing the data pipeline and the TabPFN-v2 model, a state-of-the-art architecture for tabular data, to achieve optimal predictive performance on a test set for both regression and classification tasks. The optimized TabPFN-v2 consistently outperformed other evaluated architectures, achieving an R-squared score of 0.88296 for regression and an MCC score of 0.79255 for classification on the test set, demonstrating its superior predictive capabilities. All code and resources can be found at: https://github.com/tanthinhdt/cbbl-2025.**

***Index Terms*—Nanoparticle cytotoxicity, Tabular dataset, Tabular Prior-data Fitted Network**

I. Introduction

Nanoparticles, with their characteristic dimensions ranging from 1 to 100 nanometers [1], exhibit unique physicochemical properties that have spurred revolutionary applications across diverse fields. These applications span from transforming healthcare through targeted drug delivery and advanced diagnostics to enhancing electronic devices and contributing to effective environmental remediation [2]. However, the nanoscale dimensions of these materials also introduce critical safety considerations. Their size enables interactions with biological entities at the molecular level, potentially leading to adverse effects. Consequently, the assessment of nanoparticle toxicity, particularly their capacity to induce cellular damage or death—a phenomenon termed cytotoxicity—is of paramount importance [3]. A comprehensive understanding of the mechanisms governing nanoparticle interactions with biological systems and the elicitation of cytotoxic responses is fundamental to ensuring their safe and sustainable development and widespread application.

Traditional experimental methodologies employed for toxicity assessment, encompassing both in vitro and in vivo studies, can be associated with substantial costs, extended timelines, and potential ethical concerns [4]. To circumvent these limitations, computational approaches, especially those harnessing the capabilities of machine learning and deep learning, have emerged as promising alternatives for predicting nanoparticle cytotoxicity. These in silico methods offer the potential for cost-effective, rapid, and ethically sound hazard evaluation by analyzing the physicochemical properties of nanoparticles to predict their biological interactions and toxic potential [5]. A significant number of tabular datasets, originating from diverse sources including individual research investigations and specialized nanoinformatics databases [6], have been instrumental in the development of computational models for predicting nanoparticle cytotoxicity. Consequently, data-driven approaches, which primarily operate on tabular data and rely on inductive reasoning, have dominated the field for the past several decades [7].

This report centers its analysis on the TabPFN-v2 model, recognized as a state-of-the-art architecture for tabular data, for both regression and classification tasks. Consequently, I will explore the optimization of the data pipeline and the TabPFN-v2 model itself to achieve optimal predictive performance on the test set.

II. Related Works

For tabular datasets, a range of machine learning methods have been successfully employed in various prediction tasks. Traditional algorithms such as Logistic Regression [8] and Support Vector Machines (SVMs) [9] offer interpretable linear or non-linear decision boundaries and remain valuable baselines, particularly for smaller datasets. Ensemble methods, including Decision Trees [10], Random Forests [11], and Gradient Boosting Machines (GBMs) like XGBoost [12] and LightGBM [13], have consistently demonstrated high predictive power by combining multiple weak learners. These methods excel at capturing complex non-linear relationships and feature interactions within tabular data and are often preferred in real-world applications due to their robustness and performance.

More recently, attention-based deep learning models specifically designed for tabular data have emerged as state-of-the-art contenders. Architectures like TabTransformer [14] and TabPFN (Tabular Prediction using Foundation Models) [7, 15] leverage transformer networks to learn contextual embeddings of categorical features and effectively model complex feature dependencies. TabPFN, in particular, stands out as a zero-shot or few-shot learning approach, often achieving strong performance even with limited data by leveraging pre-training on a vast collection of diverse tabular datasets. These modern techniques offer the potential to further enhance predictive accuracy, especially in scenarios with intricate relationships within the tabular features.

III. Data

A. Data Properties

The dataset utilized for both regression and classification tasks shares identical characteristics, as summarized in Table I, with the sole exception of the target variable. Comprising

TABLE I
STATISTICS OF DATASETS.

	Training set		**Test set**	
	Regression	**Classification**	**Regression**	**Classification**
Features	20			
Examples	1775		762	
Missing-value examples	1102		475	
High-viability examples	-	910	-	
Low-viability examples	-	865	-	

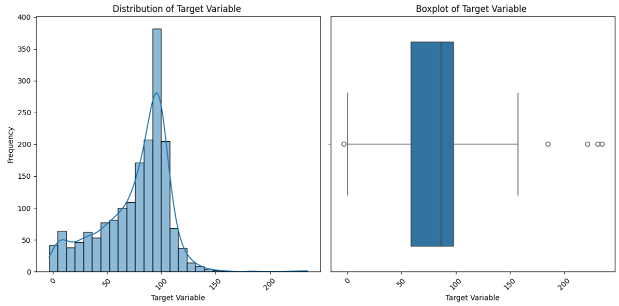


Fig. 1. Histogram and box plot of 'Target' variable of regression dataset.

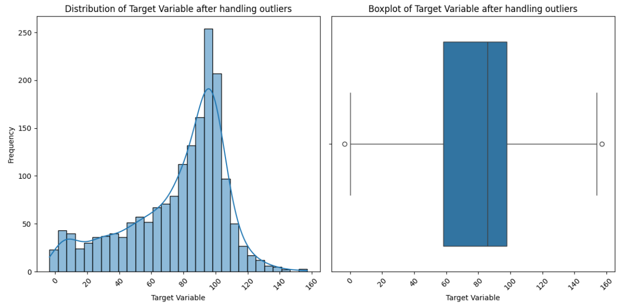


Fig. 2. Histogram and box plot of 'Target' variable of regression dataset after handling outliers.

2,573 instances, the dataset provides a reasonable number of examples for meaningful model training and was strategically split into a training set (approximately 70%) and a test set (approximately 30%), reflecting a standard partitioning approach. As further detailed in Table I, a substantial proportion of the data (around 62% in both training and test sets) exhibited missing values, exclusively concentrated within the 'Coat/Functional Group' feature.

For the regression task, the labels are continuous values representing cell viability. The distribution of the training set's target variable, visualized in Fig. 1, spans a range from -3.5573% to 234.6300%. Notably, the boxplot within this visualization indicates the presence of potential outliers in the target variable, necessitating the consideration of appropriate handling methods.

Conversely, for the classification task, the labels are binary: 1 denotes high cell viability (viability $\geq$ 85%), while 0 represents lower viability. This classification target is directly inferred from the regression viability by applying the specified threshold. Furthermore, as depicted in Table I, the classification target variable's distribution revealed a relatively balanced representation of high and low viability examples, thereby obviating the need for specific sampling techniques to address class imbalance.

B. Data Processing

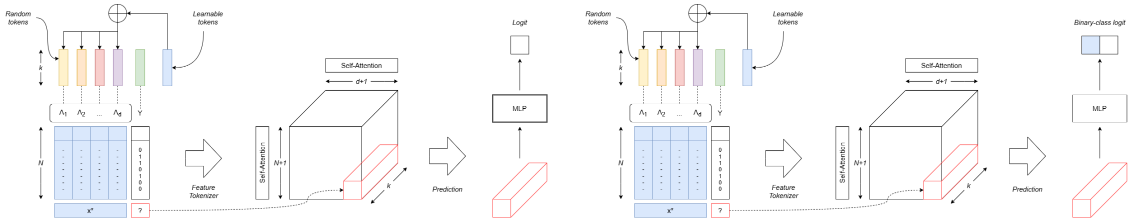
This section outlines the optimized data pipelines for the TabPFN-v2 regressor and classifier, whose performance is detailed in Section V and whose components were selected based on ablation studies in Section V-C.

The regression pipeline begins with mode imputation for missing values in the 'Coat/Functional Group' column (using the training set's mode). This is followed by Min-Max scaling for numerical features, Label encoding for categorical features, and Principal Component Analysis (PCA) to reduce 'Surface_Charge', 'Human_Animal', and 'Cell_Age' into two features. The final step involves Z-score trimming to handle outliers in the target variable. The classification pipeline is simpler, involving only mode imputation for 'Coat/Functional Group' and Label encoding for categorical features.

IV. ARCHITECTURE

As depicted in Fig. 3, TabPFN-v2 [7] is constructed upon a Transformer architecture. A defining characteristic of TabPFN-v2 is its use of in-context learning. This paradigm allows the model to learn from a given dataset, consisting of labeled training examples and unlabeled test instances, within a single forward pass of the network, without requiring any explicit gradient-based fine-tuning on that specific dataset. To effectively handle the diverse nature of tabular data, a key innovation in TabPFN-v2 is the introduction of Randomized Feature Tokens [16]. This mechanism is designed to manage the inherent heterogeneity present in tabular datasets, where the number of features and the meaning of each feature can vary significantly. Furthermore, the model employs alternating self-attention mechanisms that operate at two distinct levels: across different samples (inter-sample) and across different features within each sample (inter-feature).

The selection of a transformer architecture enables TabPFN-v2 to capture intricate dependencies that exist within tabular data. This capability extends beyond simply examining individual features in isolation, allowing the model to understand how different features interact with each other and how these interactions might vary across different data samples. This approach marks a significant shift from traditional tabular



(a) (b)

Fig. 3. Architecture of TabPFN-v2 for: (a) regression task and (b) classification task.

TABLE II
HYPERPARAMETERS FOR OPTIMIZED TABPFN-V2 MODEL.

Hyperparameter	Regressor	Classifier
Number of estimators	10	10
Softmax temperature	0.5	0.5
Average before softmax	True	True

models, which often rely on techniques that treat features as independent entities or use tree-based structures to model interactions in a more localized manner. The in-context learning strategy adopted by TabPFN-v2 suggests that the model has undergone a pre-training process that has equipped it with a general understanding of how to learn from tabular data. This pre-training allows the model to rapidly adapt to new datasets presented in the input context, making predictions without the need for the often time-consuming and computationally expensive process of fine-tuning model parameters for each specific task.

V. EXPERIMENTS

A. Implementation Details

All experiments were conducted on a P100 16GB GPU hosted on Kaggle and a local NVIDIA GTX1650Ti 4GB GPU, with the random seed consistently set to 42 to ensure reproducibility. While the primary focus of this report is the TabPFN-v2 model, leveraging the implementation and pre-trained weights provided by Prior Labs [17], a comparative analysis was also performed against other machine learning methods. The models included in this analysis can be broadly categorized into two types: data-driven models, which predominantly employ inductive reasoning (comprising traditional machine learning algorithms and TabPFN-v2), and knowledge-driven models, which often utilize deductive reasoning (specifically Large Language Models - LLMs) as described in contexts like [18]. Implementations for Logistic Regression, Decision Tree, Random Forests, and Gradient Boosting were sourced from the Scikit-learn library [19]. GPT2 [20] and BioGPT [21] were utilized via their official pre-trained models available on HuggingFace. For data-driven methods, 5-fold cross-validation was applied to measure performance prior to evaluation on the final independent test set.

In terms of data preprocessing, the ”optimized” TabPFN-v2 models employed the comprehensive data processing pipelines detailed in Section III-B. In contrast, the machine learning baseline models were evaluated using a simplified data processing approach that consisted solely of label encoding for categorical features. For the LLMs, inspired by the methodology in [18], training data required direct parsing of each row into an input text format (examples provided in Table III), followed by tokenization and padding to the longest sequence. Regarding hyperparameters, the ”optimized” TabPFN-v2 models utilized the optimized settings detailed in Table II. Hyperparameters for the baseline models were maintained at their default settings. Key training hyperparameters for the LLMs are provided in Table IV. In the ablation studies presented in Section V-C, a simplified data processing and hyperparameter setup, akin to that used for the baseline models, was applied to the TabPFN-v2 model. This was done to isolate the impact of specific data processing factors on its performance during comparative analysis.

All results presented in subsequent sections are evaluated using R-squared (R^2) for regression tasks and the Matthews Correlation Coefficient (MCC) for classification tasks. In certain instances, results are intentionally marked with a hyphen (”-”) to denote inapplicability of a particular metric (e.g., the MCC metric is not applicable to models exclusively designed for regression, such as Linear Regression). Additionally, some experiments could not be fully completed due to time constraints within the competition period.

B. Main Results

As detailed in Table V, the comparative analysis revealed that knowledge-driven methods (LLMs) achieved competitive results against data-driven approaches, albeit performing less effectively than TabPFN-v2 and ensemble methods like Random Forests. Despite their competitive accuracy, LLMs presented significant practical drawbacks, particularly much longer training times (ranging from several minutes to hours, compared to seconds for data-driven methods) and inference times that were at least three times longer than their data-driven counterparts. This suggests that data-driven methods generally retain dominance in tabular data applications due to their superior efficiency and often comparable accuracy.

Focusing on the leading data-driven model, the TabPFN-v2 architecture consistently outperformed other evaluated architectures in both cross-validation and on the final test set across both regression and classification tasks. Notably, the optimized TabPFN-v2 model achieved the highest R-squared score of 0.86839 and an MCC score of 0.79255 on the test

TABLE III
EXAMPLES OF INPUT TEXT FOR LLMS.

Training	Inference
Predict the viability of a cell. The material of the nanoparticle is Pt. The nanoparticle is inorganic. The morphology of the nanoparticle is Sphere. The fabrication method is Chemical Reduction. The surface coating is PVP. The cell type is IMR90. The number of cells (cells/well) is 5000.0. The origin species of the cell is human. The source of the cell line is Human. The type of cell tissue is Lung. The morphology of the cell is Fibroblast. The cell is in Adult stage. The cell is cell line. The exposure time is 24 hours. The exposure concentration is 25.0 ug/ml. The type of cytotoxicity test is CellTiterGlo. The test mechanism is LuciferaseEnzyme. The size of the nanoparticle is 4.0 nm. The zeta potential indicating surface charge stability of the nanoparticle is -8.0 mV. The surface charge is Negative. Viability (%): 98.293	Predict the viability of a cell. The material of the nanoparticle is Ag. The nanoparticle is inorganic. The morphology of the nanoparticle is Sphere. The fabrication method is Commercial. The surface coating is Citrate. The cell type is CCL-110. The number of cells (cells/well) is 5000.0. The origin species of the cell is human. The source of the cell line is Human. The type of cell tissue is Skin. The morphology of the cell is Fibroblast. The cell is in Fetus stage. The cell is primary. The exposure time is 24 hours. The exposure concentration is 0.5 ug/ml. The type of cytotoxicity test is MTS. The test mechanism is TetrazoliumSalt. The size of the nanoparticle is 39.94 nm. The zeta potential indicating surface charge stability of the nanoparticle is -23.5 mV. The surface charge is Negative. Viability (%):

TABLE IV
TRAINING HYPERPARAMETERS FOR LLMS.

Hyperparameter	GPT2	BioGPT
Training batch size	72	45
Number of epochs	40	
Warm-up ratio (%)	10	
Learning rate	1e-4	
Weight decay	1	
BF-16	True	

set, strongly supporting its superior predictive performance in this context. However, a significant drawback of TabPFN-v2 was its substantially longer inference time, attributed to its Transformer-based architecture, requiring at least 100 times the prediction time on the test set compared to other data-driven models. Furthermore, the inference time of TabPFN-v2 appeared to increase with the size of the training data, potentially due to its need to refit the entire dataset for each prediction. Conversely, TabPFN-v2 exhibited remarkable data efficiency, achieving comparable or even superior results to other methods while using only 90% of the training data.

C. Ablation Studies

This section evaluates different data pipeline and modeling settings to select the strategy yielding the best test set performance.

1) Missing Values: As highlighted in Section III-A, the 'Coating/Functional Group' column in both the training and test sets contains a substantial proportion of missing values (62%), prompting an investigation into their appropriate treatment. While missing data can often negatively impact model performance, the high frequency in this categorical feature suggested it might represent a genuine characteristic—the absence of a coating or functional group. Given the nature of the column and the extent of missingness, I only experimented with the common imputation strategy of filling missing values with the mode.

The results presented in Table VI indicate that mode imputation directly led to superior performance across all R-squared metrics compared to retaining the missing values, leading to its incorporation into the data pipeline for the optimized regression model. For the classification task, the initial assessment was more complex. While mode imputation showed improved performance compared to no imputation on the test set, it resulted in a decrease in MCC during cross-validation. However, through further experimentation combining it with other processing steps, mode imputation ultimately contributed to an improvement in overall classification results within the integrated pipeline. Therefore, mode imputation was adopted in the final data pipeline for the classification task.

2) Feature Scaling: To optimize the preprocessing of numerical features, this section investigates the application of Min-Max scaling and Standard scaling. Min-Max scaling transforms values according to the equation

$$\frac{\text{Value} - \text{Min value}}{\text{Max value} - \text{Min value}},$$

resulting in a normalized range of [0, 1]. In contrast, Standard scaling standardizes features by applying the formula

$$\frac{\text{Value} - \text{Mean value}}{\text{Standard deviation}},$$

yielding distributions with a mean of zero and a standard deviation of one.

As shown in Table VII, Min-Max scaling proved to be the most effective strategy across all R-squared metrics for the regression task, leading to its adoption for integration into the optimized data pipeline. For the classification task, initial evaluations also suggested that Min-Max scaling was more effective than no scaling, particularly evidenced by a wider gap in MCC on the test set compared to the performance gap observed during cross-validation when no scaling was applied. However, subsequent experiments integrating Min-Max scaling with other preprocessing steps revealed a decrease in overall MCC performance. Consequently, to maintain optimal performance for the classification task, no scaling was ultimately adopted for the numerical features in its final data pipeline.

3) Feature Encoding: To effectively represent categorical features for the optimized TabPFN-v2 model, this section investigates three distinct encoding methods: Label encoding, One-hot encoding, and Target encoding. Label encoding assigns a unique non-negative integer per category. One-hot encoding creates sparse binary vectors, significantly increasing

TABLE V
MAIN RESULTS ON CROSS-VALIDATION AND TEST SET.

Drive	Method	Data (%)	Cross-validation		Regression test set		Classification test set	
			Avg. R^2	Avg. MCC	R^2	Inference time (s)	MCC	Inference time (s)
Data	Logistic Regression	100	-	0.2750	-	-	0.16989	0.0041
	Linear Regression	100	0.1816	-	0.16989	0.0048	-	-
	Decision Tree	100	0.6677	0.6460	0.76673	**0.0008**	0.71634	**0.0012**
	Random Forests	100	0.8044	0.6966	0.84581	0.0086	0.76629	0.0062
	Gradient Boosting	100	0.6607	0.6656	0.65157	0.0026	0.69814	0.0017
	TabPFN-v2	90	0.8051	0.7245	0.85896	0.6432	0.76895	0.3127
	TabPFN-v2	100	0.8122	0.7237	0.86839	0.6862	0.76104	0.3397
	TabPFN-v2 (Optimized)	100	**0.8480**	**0.7300**	**0.88296**	0.8713	**0.79255**	0.8338
Knowledge	GPT2	100	-	-	0.78022	2.4049	-	-
	BioGPT	100	-	-	0.79314	4.1377	-	-

TABLE VI
RESULTS OF MISSING VALUES HANDLING ON CROSS-VALIDATION AND TEST SET.

Method	Cross-validation		Test set	
	Avg. R^2	Avg. MCC	R^2	MCC
None	0.8122	**0.7237**	0.86839	0.76104
Mode	**0.8200**	0.7173	**0.87879**	**0.78216**

TABLE VII
RESULTS OF FEATURE SCALING ON CROSS-VALIDATION AND TEST SET.

Method	Cross-validation		Test set	
	Avg. R^2	Avg. MCC	R^2	MCC
None	0.8122	**0.7237**	0.86839	0.76104
Standard	0.8115	0.7216	0.86839	0.76625
Min-Max	**0.8131**	0.7186	**0.86908**	**0.76629**

dimensionality. Target encoding replaces categories with the mean target value.

As shown in Table VIII, Label encoding consistently outperformed other methods across all R-squared metrics for the regression task, leading to its adoption. For the classification task, Label encoding showed a higher MCC score during cross-validation (0.023 higher than One-hot encoding). While One-hot encoding provided a marginal MCC advantage on the test set (0.01842 higher), its substantial increase in data size (from 20 to over 240 features) was a significant drawback. Therefore, Label encoding was selected for the classification pipeline due to its cross-validation MCC superiority and its more efficient data representation.

4) Dimensionality Reduction: Identifying and mitigating the impact of irrelevant or low-impact features is crucial for both model accuracy and resource efficiency. To this end, I trained a Random Forests model and analyzed the resulting feature importance scores, as visualized in Fig. 4. The three features exhibiting the lowest impact—namely 'Surface_Charge', 'Human_Animal' and 'Cell_Age'—were targeted for dimensionality reduction or removal. My experimentation involved two main strategies: first, I independently removed each of these low-impact features and evaluated the resulting model performance. Second, I explored the use of Principal Components Analysis (PCA) to compress these features. For each combination of these low-impact features, PCA with (number of features - 1) components was applied with the aim of reducing dimensionality while retaining as much information as possible.

TABLE VIII
RESULTS OF FEATURE ENCODING ON CROSS-VALIDATION AND TEST SET.

Method	Cross-validation		Test set	
	Avg. R^2	Avg. MCC	R^2	MCC
Label	**0.8122**	**0.7237**	**0.86839**	0.76104
One-hot	0.8096	0.7007	0.86474	**0.77946**
Target	0.7869	0.6960	0.85966	0.76125

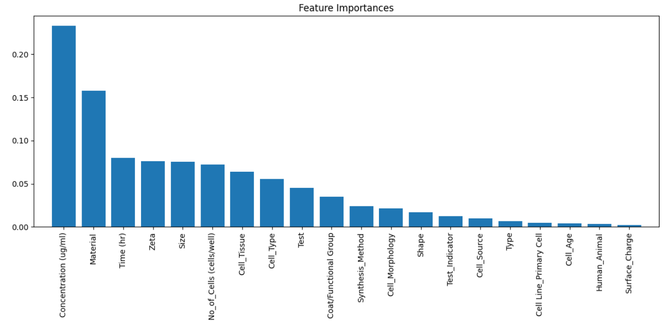


Fig. 4. Feature importance in regression task.

As illustrated in Table IX, removing each of these three features individually generally led to a decrease in R-squared score during cross-validation, but an increase in R-squared score on the test set. Conversely, applying PCA to combinations of these features often improved model performance across the R-squared metrics. Notably, performing PCA on all three features, retaining two components, yielded the best R-squared results, surpassing other configurations in all R-squared metrics. Consequently, this PCA approach on these three features with two components was adopted in my data pipeline for regression task.

5) Outliers: This section evaluates various strategies for handling outliers in the 'Target' variable in regression task through the combination of different outlier detection and treatment methods. Two outlier detection techniques were employed: the Interquartile Range (IQR) method, which identifies outliers as values below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR, and the Z-score method, which flags data points

TABLE IX
RESULTS OF DIMENSIONALITY REDUCTION ON CROSS-VALIDATION AND TEST SET.

Reduced features				**Cross-validation**		**Test set**	
Surface_charge	**Human_Animal**	**Cell_Age**	**Cell Line_Primary Cell**	**Avg. R^2**	**Avg. MCC**	R^2	**MCC**
				0.8122	0.7237	0.86839	0.76104
✓				0.8130	-	0.86905	-
	✓			0.8107	-	0.87177	-
		✓		0.8072	-	0.86846	-
			✓	0.8099	-	0.87034	-
✓	✓			0.8163	-	0.86997	-
✓	✓	✓		0.8172	-	**0.87186**	-
✓	✓	✓	✓	**0.8177**	-	0.87095	-

TABLE X
RESULTS OF OUTLIERS HANDLING ON CROSS-VALIDATION AND TEST SET.

Detection	**Handling**		**Cross-validation**		**Test set**	
	Trim	**Clip**	**Avg. R^2**	**Avg. MCC**	R^2	**MCC**
None			0.8122	0.7237	0.86839	0.76104
IQR	✓		0.8408	-	0.86913	-
		✓	0.8288	-	0.86816	-
Z-score	✓		**0.8418**	-	0.86737	-
		✓	0.8291	-	**0.86928**	-

with a Z-score exceeding 3 or falling below -3. For outlier treatment, I experimented with clipping (capping outlier values at predefined boundaries) and trimming (removing outlier data points). As shown in Table X, addressing outliers generally improved performance compared to leaving them untreated. Furthermore, Z-score and trimming tended to be more effective for most cases, leading the adoption of the Z-score trimming method.

VI. CONCLUSION

In this study, I evaluated the TabPFN-v2 model for predicting nanoparticle cytotoxicity. My experiments demonstrate the effectiveness of the TabPFN-v2 architecture, particularly with an optimized data pipeline, achieving strong predictive performance. While TabPFN-v2 exhibited longer inference times compared to traditional methods, its accuracy and data efficiency are notable advantages. Future work could focus on reducing the inference time of TabPFN-v2 and exploring its applicability to larger, high-dimensional datasets.

REFERENCES

[1] J. Khalili Fard, S. Jafari, and M. A. Eghbal, "A Review of Molecular Mechanisms Involved in Toxicity of Nanoparticles," *Advanced Pharmaceutical Bulletin*, vol. 5, no. 4, pp. 447–454, Nov. 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4729339/

[2] M. Garcés, L. Cáceres, D. Chiappetta, N. Magnani, and P. Evelson, "Current understanding of nanoparticle toxicity mechanisms and interactions with biological systems," *New Journal of Chemistry*, vol. 45, no. 32, pp. 14 328–14 344, Aug. 2021, publisher: The Royal Society of Chemistry. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2021/nj/d1nj01415c

[3] M. Awashra and P. Młynarz, "The toxicity of nanoparticles and their interaction with cells: an in vitro metabolomic perspective," *Nanoscale Advances*, vol. 5, no. 10, pp. 2674–2723, 2023, publisher: Royal Society of Chemistry. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2023/na/d2na00534d

[4] I. Yousaf, "AI and Machine Learning Approaches for Predicting Nanoparticles Toxicity The Critical Role of Physiochemical Properties," Sep. 2024, arXiv:2409.15322 [physics]. [Online]. Available: http://arxiv.org/abs/2409.15322

[5] I. Furxhi and F. Murphy, "Predicting In Vitro Neurotoxicity Induced by Nanoparticles Using Machine Learning," *International Journal of Molecular Sciences*, vol. 21, no. 15, p. 5280, Jul. 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7432486/

[6] M. Ahmadi, S. M. Ayyoubzadeh, and F. Ghorbani-Bidkorpeh, "Toxicity prediction of nanoparticles using machine learning approaches," *Toxicology*, vol. 501, p. 153697, Jan. 2024.

[7] N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirrmeister, and F. Hutter, "Accurate predictions on small data with a tabular foundation model," *Nature*, vol. 637, no. 8045, pp. 319–326, Jan. 2025, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-024-08328-6

[8] J. S. Cramer, "The Origins of Logistic Regression," Rochester, NY, Dec. 2002. [Online]. Available: https://papers.ssrn.com/abstract=360300

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[10] W. A. Belson, "Matching and Prediction on the Principle of Biological Classification," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 8, no. 2, pp. 65–75, 1959, publisher: [Royal Statistical Society, Oxford University Press]. [Online]. Available: https://www.jstor.org/stable/2985543

[11] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794, arXiv:1603.02754 [cs]. [Online]. Available: http://arxiv.org/abs/1603.02754

[13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree."

[14] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," Dec. 2020, arXiv:2012.06678 [cs]. [Online]. Available: http://arxiv.org/abs/2012.06678

[15] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter, "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second," Sep. 2023, arXiv:2207.01848 [cs]. [Online]. Available: http://arxiv.org/abs/2207.01848

[16] H.-J. Ye, S.-Y. Liu, and W.-L. Chao, "A Closer Look at TabPFN v2: Strength, Limitation, and Extension," Feb. 2025, arXiv:2502.17361 [cs]. [Online]. Available: http://arxiv.org/abs/2502.17361

[17] P. Labs, "Prior Labs: Leading Tabular Foundation Model (TabPFN)." [Online]. Available: https://priorlabs.ai

[18] T. Li, S. Shetty, A. Kamath, A. Jaiswal, X. Jiang, Y. Ding, and Y. Kim, "CancerGPT for few shot drug pair synergy prediction using large pretrained language models," *npj Digital Medicine*, vol. 7, no. 1, pp. 1–10, Feb. 2024, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41746-024-01024-9

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[20] "microsoft/biogpt · Hugging Face." [Online]. Available: https://huggingface.co/microsoft/biogpt

[21] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, "BioGPT: generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, p. bbac409, Nov. 2022. [Online]. Available: https://doi.org/10.1093/bib/bbac409