A Multi-Method Approach to Molecular Activity Prediction

Carlotta Hölzle*†, Kevin Karim*, Kusumastuti Cahyaningrum*
**KTH Royal Institute of Technology*, Stockholm, Sweden
{csholzle, kkarim, kcah}@kth.se
†*Technical University of Munich*, Munich, Germany

***Abstract*—Predicting molecular activity is a central task in computational chemistry with critical applications in drug discovery and toxicity assessment. Molecules can be naturally described as graphs, enabling both classical graph-theoretic analysis and modern machine learning approaches. In this study we empirically compare three distinct paradigms for molecular representation: (1) global structural descriptors (e.g., graph density, SPID), (2) substructure-based molecular fingerprints (ECFP4), and (3) learned embeddings from Graph Neural Networks (GNNs). Using four TUdataset benchmark datasets, we evaluated the predictive performance of each feature type in a diverse set of classifiers, including Logistic Regression, Random Forests, and different graph convolutional networks. Our results show that fingerprint-based features consistently outperform graph-theoretic metrics in both accuracy and ROC-AUC, particularly when used with linear models. Structural features such as SPID and density exhibit statistically significant differences between classes but provide limited practical separability. Combining fingerprints with graph-theoretic features yields no inherent improvement, suggesting redundancy. GNN models match classical baselines in performance while offering an end-to-end alternative. This study provides actionable guidance for selecting feature representations in molecular classification and highlights the trade-offs between interpretability, scalability, and predictive power.**

***Index Terms*—Molecular Activity Prediction, GNNs, Graph-Theoretic Features, Molecular Fingerprints, ML classifiers**

I. Introduction

Predicting molecular activity is a fundamental challenge in computational chemistry, with direct applications in drug discovery, toxicity prediction, and chemical screening [26, 3, 31]. Molecules can naturally be modeled as graphs, where atoms correspond to nodes and chemical bonds to edges. With graph representations and labels for activity of the molecules, we can formulate the activity prediction as a graph classification problem. The aim then is to project the different graphs to a high dimensional space based on their features, and, with the use of both classical graph theory and modern machine learning techniques, infer bioactivity or toxicity from their proximity to each other. Recent progress has produced two dominant paradigms for graph-based molecular analysis [30]. On one end of the spectrum, handcrafted descriptors, such as molecular fingerprints or global topological metrics, are widely used due to their interpretability, ease of use, and compatibility with traditional machine learning models, such as Logistic Regression or Random Forests. On the other end, Graph Neural Networks, particularly Graph Convolutional Networks (GCNs), have emerged as powerful end-to-end models that learn molecular representations directly from graph structure. However, GNNs often require significant computational resources and lack interpretability [12, 28]. Despite the wide adoption of both approaches, little is known about their comparative strengths across diverse biochemical datasets [30]. Moreover, it remains unclear whether structural graph metrics such as SPID, clustering coefficient, or graph density, which are commonly used in network science but rarely in cheminformatics, can provide competitive predictive power in molecular classification tasks [20]. In this study, we address this gap by conducting a comprehensive empirical comparison of three feature paradigms: (1) global graph-theoretic metrics, (2) substructure based molecular fingerprints, and (3) GNN-based learned embeddings. Using four well-known datasets from the TUdataset benchmark (DHFR, AIDS, PTC, and Mutagenicity) [18], we test the predictive power of feature representation 1 and 2 across a suite of machine learning models and the graph representation on two GNN implementations. We additionally explore the complementarity of feature sets by combining representation 1 and 2. For evaluation, we monitor the classification accuracy and the ROC-AUC curve, and compare our results with previous research as described in Table I. Our results aim to answer the research question: Which type of representation and architecture works best for which dataset characteristics, and how can one avoid exhaustive model testing by selecting effective features early on? This work provides actionable guidelines for researchers in computational biology and cheminformatics seeking efficient modeling strategies.

II. Related Work

Graph-theoretic metrics such as graph density, clustering coefficient, and node/edge counts are long established tools for capturing structural properties of networks. [27] ”small-world” model and [1]'s work on distance dispersion in social networks motivate the use of dispersion based metrics like the Shortest-Path Index of Dispersion (SPID), which we adopt to quantify intra-graph heterogeneity. [16] demonstrate that global topological features can be used as input to classifiers like SVMs or Random Forests, showing competitive results on graph classification benchmarks. This aligns with our goal to assess the statistical relevance and predictive power of such features in a biochemical context. In early cheminformatics work, datasets like MUTAG and PTC were used to benchmark handcrafted graph features with classical ML models. [25] and [8] report strong performance using topological descriptors, showing that explicit substructure enumeration is not always necessary. Graph kernels, such as the Weisfeiler-Lehman kernel [24], have also shown success, although at the cost of higher computational demands. Our approach relates closely to these studies, but instead compares non fine-tuned GNNs with a broader range of ML models and graph properties. Parallel to these efforts, molecular fingerprints have become a dominant input representation in drug discovery pipelines. Rogers and Hahn [22]'s ECFP encodes substructural features into bit vectors, enabling fast and scalable learning with models like Logistic Regression and gradient boosting. Gadiya et al. [5] use such fingerprints to achieve >90% accuracy in classifying antibacterial activity, illustrating their strong predictive power in real world settings. We extend this line of work by testing fingerprint-based models across multiple datasets and comparing their performance against both GNNs and structural metrics. Finally, Graph Neural Networks have become the state of the art in graph-based learning. GCNs [13], GIN [29], and related architectures learn representations by aggregating neighborhood information, allowing them to capture both local and global dependencies. Compared to traditional machine learning approaches, which often depend on handcrafted features, GNNs offer a significant advantage through their ability to construct rich and expressive representations directly from graph-structured data [35]. An additional motivation comes from the progress in graph representation learning [2, 4, 6, 7, 32], which focuses on encoding nodes, edges, or entire subgraphs into low dimensional vector embeddings. Unlike conventional models, GNNs learn to capture and utilize the relational and topological structure of graphs, making them particularly effective in domains where the connections between entities are crucial, such as in molecular graphs. Previous work specifically on the AIDS, DHFR, PTC_MR and Mutagenicity datasets acts as benchmarks for evaluation our models. Table I shows the top performing ML and GNN models for each dataset.

TABLE I: State of the art ML and GNN models along with their classification accuracy.

Dataset	ML	GNN
AIDS	kNN 97.3% [21]	FIT-GNN: 84.3% [23]
DHFR	SVM 80.8% [14]	GPNN: 82.15% [10]
PTC_MR	SVM 68.61% [11]	$\text{U2GNN}_{\text{(Unsupervised)}}$: 92.67% [19]
Mutagenicity	SVM 84% [9]	$\text{HGP-SL}_{\text{GCN}}$: 82.15% [34]

III. Methodology

A. Data Exploration

To ensure standardized preprocessing and reproducibility, we selected molecular datasets from the `TUDataset` collection[18]. This repository offers a broad range of graph classification datasets with consistent formatting, making it particularly suitable for benchmarking. Importantly, we sought to select datasets with downstream biomedical relevance, where improved molecular classification could potentially support applications such as drug discovery or toxicity screening. We ultimately selected four datasets: DHFR, AIDS, PTC_FM, and Mutagenicity. Each consists of molecules represented as graphs, with binary labels indicating biological activity or toxicity. DHFR was chosen as our primary dataset due to its favorable properties for model prototyping: medium dataset size and a fairly balanced class split of 60:40. It contains 756 drug like molecules labeled based on their inhibitory effect on Dihydrofolate Reductase (DHFR), a key enzyme in DNA and protein synthesis. The AIDS dataset, by contrast, contains over 2000 molecules labeled according to their ability to inhibit HIV replication. It presents a significant class imbalance (80% negative), making it valuable for testing model performance under skewed label distributions. PTC_MR, a smaller dataset with 344 compounds labeled for carcinogenicity in rats, allows us to examine model behavior under limited data availability. Finally, the Mutagenicity dataset comprises 4,337 compounds categorized as mutagenic or non-mutagenic. Its larger size and mild class imbalance make it useful for assessing model scalability. To characterize the datasets quantitatively, we computed statistics such as class ratios, number of samples, and average graph sizes. DHFR, for example, does not have disconnected graphs and features moderate graph complexity with consistent node and edge counts. We visualized representative molecules from each dataset, see Figure 1, to gain intuition about structural motifs, such as aromatic rings or heterocycles, which are relevant to chemical activity. An overview of dataset statistics, including class distributions and structural properties, is provided in Table II.

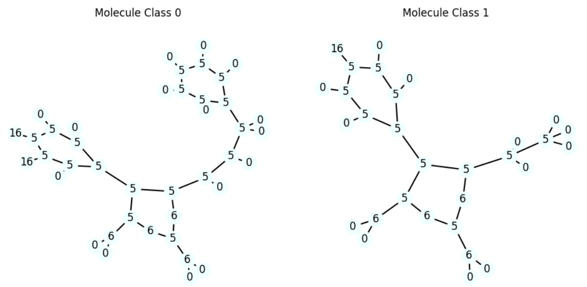


Fig. 1: Example of molecules form class 0 and 1 in the DHFR dataset. Node labels code for atom type.

TABLE II: Summary statistics of selected TUDataset molecular datasets.

Dataset	# Graphs	Class Balance	Avg. Nodes	Avg. Edges
DHFR	756	61% / 39%	295	461
AIDS	2000	20% / 80%	1600	400
PTC_FM	344	41% / 59%	206	143
Mutagenicity	4337	45% / 55%	2401	1936

B. Statistical Analysis of Structural Graph Features

To quantitatively assess structural differences between molecular graphs, a set of global graph-theoretic metrics was computed for each sample. We extracted the number of nodes and edges, together with the average node degree, the density (ratio of existing to possible edges), clustering coefficient and SPID of each graph.

Disconnected graphs were standardized by extracting their largest connected component using the `networkx.connected_components()` function from the NetworkX library. To determine whether feature distributions differed significantly between graph classes, both parametric and non-parametric tests were employed. Specifically, two-sample t-tests (`scipy.stats.ttest_ind`), Mann–Whitney U tests (`scipy.stats.mannwhitneyu`) and Cohen's d were performed for each feature. Only features with non-zero variance were considered, and statistical significance was evaluated under a two-sided alternative hypothesis at a threshold of $p < 0.05$. Cohen's d for standardized effect size was calculated to assess the magnitude of class separation independently of sample size. The effect size was calculated as:

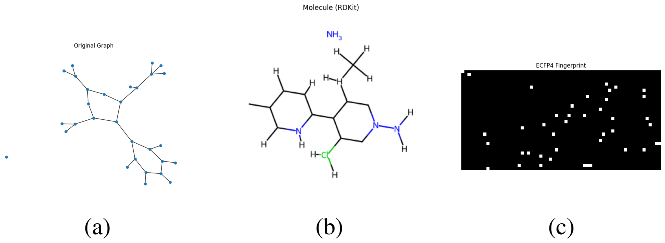
$$d = \begin{cases} \frac{\mu_0 - \mu_1}{s_p} & \text{if} s_p > 0 \\ \text{NaN} & \text{else} \end{cases} \quad \text{with} \quad s_p = \sqrt{\frac{(n_0 - 1)s_0^2 + (n_1 - 1)s_1^2}{n_0 + n_1 - 2}} \tag{1}$$

Here, μ_0 and μ_1 denote the sample means, s_0 and s_1 the sample standard deviations, and n_0, n_1 the number of samples in each class. While p-values indicate whether group differences are statistically significant, Cohen's d offers a more interpretable, scale-invariant measure of effect size. Consequently, Cohen's d is used as the principal metric for comparing structural imbalance across datasets in the subsequent analysis.

C. Standard ML Methods

A broad range of standard classifiers were employed to evaluate predictive performance. Specifically, we tested Logistic Regression, support vector machines (SVM), decision trees, Random Forests, AdaBoost, gradient boosting, k-nearest neighbors (k-NN), Gaussian Naive Bayes, linear discriminant analysis (LDA), and multi-layer perceptrons (MLP). All models were implemented using Scikit-learn, with hyperparameters left at their default settings except for the number of iterations which was set to 1000 for MLP and LR to ensure convergence. Performance was assessed using 5-fold cross-validation, and we report mean accuracy, ROC AUC, and corresponding standard deviations. The classifiers were selected for their ability to handle sparse binary input and to facilitate fair comparison across both linear and nonlinear model families.

a) Fingerprint Engineering: To enable the application of classical machine learning models, each input graph was transformed into a chemically meaningful molecular fingerprint. These fingerprints were generated using the RDKit library [15], which enables the extraction of structural descriptors from molecular graphs. Prior to fingerprint generation, we reconstructed chemically valid molecules by assigning atomic numbers to node labels and enforcing basic valence rules to eliminate invalid bond configurations. This sanitization step was essential, as the raw graph data occasionally contained chemically implausible structures. For instance, hydrogen atoms with more than one bond, due to oversimplified graph construction. Atoms were pruned to respect maximum allowed valences (e.g., carbon: 4, nitrogen: 3, hydrogen: 1), and molecules failing valence correction were excluded from fingerprint computation. Valid molecular graphs were then converted into Extended Connectivity Fingerprints (ECFP4) with a radius of 2 and 2048-bit dimensionality. These circular fingerprints encode the presence or absence of specific atom-centered substructures by applying iterative hashing of local neighborhoods. The resulting bit vectors are sparse: each bit position denotes a particular hashed substructure pattern (e.g., ”a carbon bonded to an amine”), and only a small subset of bits are activated per molecule, white squares in 2, reflecting its unique chemical motifs. The resulting matrix is sparse, however according to Zhang et al. [33] sparse binary vectors can be used as input features to classical machine learning techniques. To train the classifiers we constructed a feature matrix $X \in 0, 1^{n \times 2048}$, where each row corresponds to a molecule and each column represents a fingerprint bit.



(a) (b) (c)

Fig. 2: Conversion of the molecular graphs to fingerprints. (a) Shows the graph of one molecule as provided by TUDataset. (b) is the conversion into a molecular structure, note that this molecule is flipped compared to the graph in (a). Finally, (c) shows a 2D representation of the fingerprint of the molecule.

b) Graph Theoretic Descriptors: In addition to chemically informed fingerprints, we computed an extended set of graph-theoretic descriptors to capture the topological structure of molecular graphs. These included the number of nodes and edges, graph density, and summary statistics of the degree distribution (mean, standard deviation, minimum, maximum). Furthermore, we calculated the average clustering coefficient and several shortest-path-based metrics such as diameter, radius, average shortest path length, and SPID. All graph-theoretic features were extracted using NetworkX, and molecular graph construction was performed with RDKit.

c) Combined Features: To assess whether topological and chemical information provide complementary predictive signals, we additionally evaluated a combined feature set by concatenating the graph-theoretic descriptor matrix with the fingerprint vectors.

D. Graph Neural Networks

For graph-level classification we evaluate two Graph Convolutional Network (GCN) architectures. GCNConv is a GCN which uses spectral-based convolution layers that apply graph Laplacian normalization as proposed by Kipf and Welling [13]. GraphConv is a GCN version, which uses a spatial operator derived from the Weisfeiler-Lehman graph isomorphism test, introduced by Morris et al. [17]. Both models are implemented using the PyTorch Geometric library and consist of three convolutional layers with 64 hidden units and ReLU activations, followed by global mean pooling and a fully connected MLP classifier.

a) Graph Convolutional Network Model Architecture: The first model uses GCNConv layers, which implement a spectral formulation of graph convolution. Node features are updated by aggregating neighbor features, weighted by a symmetrically normalized adjacency matrix to account for node degree. This normalization reduces the risk that high-degree nodes dominate the aggregation and helps prevent instabilities such as exploding or vanishing gradients in graphs with irregular degree distributions [13]. GCNConv applies a renormalization trick to further stabilize training. This, however, introduces computational overhead due to matrix operations, particularly in large or dense graphs. We picked GCNConv because it remains a standard and widely adopted baseline in graph learning [19, 34].

b) GraphConv-Based Model Architecture: The second model replaces GCNConv with GraphConv layers, which follow a spatial message-passing framework. Unlike GCNConv, GraphConv does not apply degree-based normalization. Instead, it updates each node by separately transforming its own features and summing the unnormalized features of its neighbors. This design emphasizes raw information flow, improving the model's capacity to capture structural variations that might be suppressed by normalization. To address the absence of inherent normalization, batch normalization layers are added after each GraphConv layer, promoting stable training and faster convergence at the cost of slightly increased model complexity.

c) Training and Evaluation: Both GNN models are trained using the Adam optimizer with a learning rate of 0.0001. To ensure a robust and unbiased evaluation, 10-fold stratified cross-validation is performed using the `StratifiedKFold` function from sklearn, which maintains class distribution across folds. In addition, the datasets were shuffled before creating the batches. Both models are evaluated using accuracy and ROC AUC, with results averaged across all folds to provide reliable performance comparisons.

For regularisation, weight decay of $5e^{-4}$ and dropout with probability $p = 0.5$ are incorporated in the models. To further reduce potential overfitting, early stopping with a patience of 10 epoch is applied, monitoring the validation loss in each fold. The loss function used is Cross-Entropy, which is suitable for binary classification. Training is conducted with a mini-batch size of 64, for a maximum of 200 epochs per fold during cross-validation.

IV. Results

A. Structural Signal and Statistical Separation in Graph Features

Initial statistical analyses focused on three global graph-theoretic metrics; SPID, density, and clustering coefficient, within the DHFR dataset. Density was significantly higher in active compounds (t = 5.242, p = 2.08e-07), while SPID was elevated in inactive compounds (t = -4.218, p = 2.8e-05), suggesting structural differences in connectivity. Clustering coefficient showed no variation and was therefore excluded from further consideration. Despite statistical significance, classification based solely on SPID and density yielded limited performance, with strong recall for active graphs (98%) but poor recall for inactive ones (7%), resulting in an overall accuracy of approximately 59%. These results underscore the distinction between statistical group differences and practical class separability.

To better understand structural difference between the binary classes per dataset the analysis was extended to a broader set of global structural features. Table III reports Cohen's d values for key graph metrics. The AIDS dataset exhibited strong class separation across multiple features, including node and edge count, density, and SPID. DHFR displayed moderate separation, particularly in size related features, while Mutagenicity and PTC_MR showed only weak or negligible structural imbalance.

Although individual graph features are typically insufficient for accurate classification, their statistical distributions provide valuable insights into dataset specific structural patterns and may help explain variation in model performance.

Dataset	Nodes	Edges	Density	Avg. Degree	Clustering	SPID
AIDS	2.14	2.15	-3.11	0.95	-0.19	1.87
DHFR	0.35	0.34	-0.38	-0.18		0.31
Mutagenicity	-0.10	-0.06	-0.05	0.54	0.26	-0.06
PTC_MR	0.11	0.09	-0.32	0.09	-0.24	0.15

TABLE III: Cohen's d for structural features with highest class separation (positive values indicate higher mean in class 0, negative in class 1).

B. Performance Classical ML Models

Figure 3 shows the average classification accuracy of all evaluated models on the fingerprint-only feature matrix from the DHFR dataset. Among all classifiers, Logistic Regression achieved the highest accuracy on fingerprints (0.798 ± 0.033), closely followed by MLP, Random Forest, and SVM. The best ROC-AUC was observed for Logistic Regression (0.875 ± 0.030), suggesting that linear separation is surprisingly effective in this high-dimensional sparse space, consistent with previous findings that molecular fingerprints perform well with linear models [33]. In contrast, when using only graph-theoretic features, accuracy dropped considerably, see Figure 8. Logistic Regression on structural features reached only 0.705 ± 0.040 accuracy and a ROC-AUC of 0.737 ± 0.053. Naive Bayes performed particularly poorly (accuracy: 0.562 ±

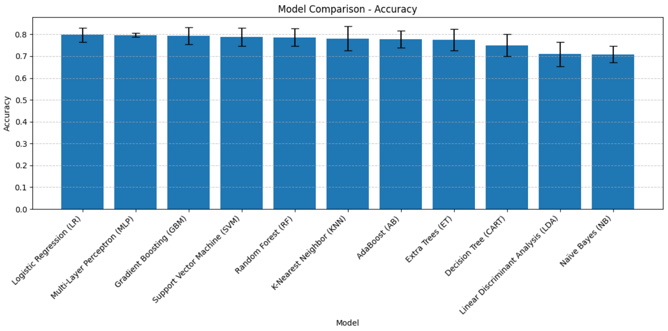


Fig. 3: Mean classification accuracy with standard deviation for each model using ECFP4 fingerprints as input.

0.037), highlighting the limited discriminative power of graph-only descriptors in the absence of chemical semantics. The combined feature space, merging structural and chemical information, did not uniformly improve performance. For Logistic Regression, accuracy decreased by 0.02 points while ROC-AUC dropped by 0.34 points, additionally for both evaluation metrics, we saw an increase in standard variation. Beyond predictive performance, we analyzed the computational cost associated with training each model (see Figure 6). As expected, tree-based ensemble models (e.g., Gradient Boosting, Random Forest) and SVMs required significantly more time, with MLPs being the slowest overall, over 100% increase of training time on fingerprints compared to Logistic Regression or Naïve Bayes. Notably, the models that received the graph features as input trained faster across the board due to their low dimensional input. Overall, these results confirm that chemically informed fingerprints are superior to topology-only descriptors for molecular activity prediction, especially when combined with robust linear or ensemble models.

C. Generalization and Comparative Performance Across Datasets

To assess the generalizability of our multi-method framework, we extended fingerprint- and graph-based classification experiments to three additional biochemical datasets, Mutagenicity, AIDS, and PTC_MR, each varying in size, chemical diversity, and graph complexity. Across all datasets, fingerprint-based models consistently outperformed those relying solely on graph-theoretic descriptors, with an average accuracy gain of 8.3 percentage points (pp), see TableVI. This trend corroborates our findings on DHFR and reinforces the superior capacity of substructural fingerprints to encode biochemically relevant information. Model rankings were largely stable across datasets: SVMs and ensemble methods (Random Forest, Extra Trees, Gradient Boosting) dominated performance, while simpler models like Naïve Bayes and LDA lagged, particularly on high dimensional inputs. Logistic Regression, although competitive with fingerprints, often failed to benefit from additional structural features. Figure 4 summarizes the performance of top models using either fingerprints (FP) or combined features (FP + graph-theoretic). In Mutagenicity, combining features improved accuracy for ensemble models (e.g., Extra Tree accuracy: 0.7549 → 0.7782). Similarly, in AIDS, the optimal model shifted from SVM with FP (accuracy: 0.9895) to AdaBoost on combined input (accuracy: 0.9985), accompanied by a reduction in prediction variance. The AIDS dataset offered a unique advantage: access to node level attributes (e.g., atom types, charges). These features, as already available in numerical form, could be used as ML input features without much preprocessing. Models using node attributes achieved competitive accuracies as FP trained models, reaching >99% accuracy, see Figure 9. PTC_MR, being the smallest dataset with the lowest structural difference between classes displayed different behaviour. While fingerprint models still led, performance margins over graph-based inputs narrowed, and combined features only modestly improved outcomes for specific models (e.g., GBM). In fact, SVM performance slightly declined with combined features (accuracy: 0.6250 → 0.5902), though prediction stability improved (std.: 0.0569 → 0.0439). Random Forest outperforms SVM on the combined feature input (accuracy: 0.6193 vs. 0.5902), however, with an increased variance of 0.3 points.

Regarding runtime, models trained on low dimensional graph metrics were consistently faster, while fingerprint-based models incurred higher training times, especially for SVMs (nearly 400-times longer on Mutagenicity), as shown in Figure 6.

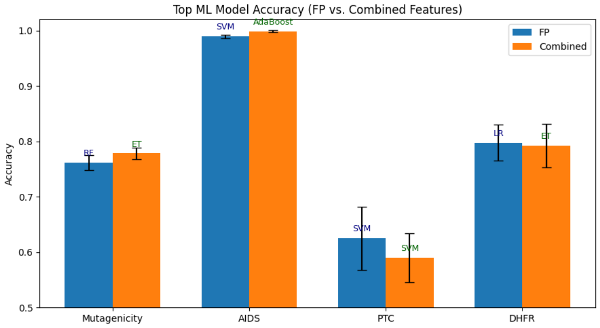


Fig. 4: Top ML Model Accuracy for Fingerprints (FP) and Combined Features (FP + Graph-Theoretic). Error bars indicate one standard deviation.

D. Performance GNN Models

As for the graph neural networks, the GraphConv based model achieved a significantly higher accuracy then GCNconv on the Mutagenicity, AIDS and DHFR, datasets, as seen in Table IV. The performance gain ranged from +5.5 pp in mutagenicity to +17.0 pp in DHFR. Only in PTC did GCNConv score the higher accuracy by 3.4 pp, however, both GCNConv and GraphConv delivered unsatisfactory results in the dataset, with classification accuracies of 57.8% and 54.4% respectivly. In terms of AUC, GraphConv consistently outperformed GCNConv across all datasets. On AIDS, the 14.2% accuracy improvement was accompanied with a 22.3 pp increase in AUC. In DHFR, the improvement was even more substantial, with 31.9 pp in AUC. Mutagenicity only showed a moderate improvement accuracy and similarly in AUC , by an increase of 5.0 pp. However, while GCNConv outperformed GraphConv in the PTC_MR datase in terms accuracy, GraphConv still yielded a slightly higher AUC by 4.7 pp.

Dataset	Accuracy			AUC		
	GCNConv	GraphConv	△ Acc. (pp)	GCNConv	GraphConv	△ AUC (pp)
AIDS	0.787	**0.929**	14.2	0.738	**0.961**	22.3
DHFR	0.610	**0.780**	17.0	0.545	**0.864**	31.9
Mutagenicity	0.751	**0.806**	5.5	0.828	**0.878**	5.0
PTC_MR	**0.578**	0.544	–3.4	0.517	**0.565**	4.8

TABLE IV: classification accuracy and AUC of the GCNConv and GraphConv models, along with percentage point(pp) difference, across datasets.

E. Model Performance Comparison Across Datasets

Table V summarizes the classification accuracy for the best performing classical ML models (using fingerprints and graph-theoretic features), GCNConv, and GraphConv across the four benchmark datasets. Key values of Cohen's d on the SPID properties and main structural imbalance drivers are also reported.

Dataset	Best ML (FP)	Best ML (CB)	GCNConv	GraphConv	Cohen's d (SPID)	Imbalance Drivers
AIDS	0.990	**0.999**	0.787	0.929	1.87	#Nodes, Density, SPID
DHFR	**0.798**	0.792	0.609	0.780	0.31	Nodes/edges, SPID
Mutagenicity	0.762	0.778	0.751	**0.806**	-0.06	Degree, Clustering
PTC_MR	**0.619**	0.590	0.578	0.544	0.15	Density

TABLE V: Classification accuracy for best ML models on fingerprints (FB), Fingerprints and Graph-theoretic features combined (CB), GCNConv, GraphConv. Along with Cohen's d on SPID and main imbalance drivers. Bold values indicate best performing model.

Across all datasets, the relative performance of GNNs, and classical ML models was closely associated with the degree of structural imbalance (quantified by Cohen's d), and dataset size. In AIDS, exhibiting extreme structural and class imbalance, classical ML models using fingerprints (SVM: 99.0%) and graph-theoretic features (Gradient Boosting: 99.9%) outperformed both GNNs, with GraphConv achieving 92.9% and `GCNConv` lagging behind at 78.7%. In DHFR, despite moderate imbalance (Cohen's $d = 0.31$), GraphConv achieved competitive performance (78.0%), only slightly trailing classical ML (FP LR: 79.8%), while GCNConv lagged behind at 61.0%, indicating greater sensitivity to dataset structure. On the Mutagenicity dataset, where structural imbalance was negligible, GraphConv outperformed all models, including classical ML, reaching 80.6%. Finally, in PTC_MR, the smallest and most balanced dataset, classical ML (SVM: 63.0%) again outperformed deep models, reaching accuracies of only 54.4% and 57.8% for GraphConv and GCNConv respectively. For all datasets, high mean ROC-AUC for classical models confirmed that their superior accuracy did not arise from majority class prediction alone.

V. Discussion

A. Feature Effectiveness and Model Robustness in Classical ML Setting

Our results confirm that chemically aware fingerprint representations provide the most reliable foundation for molecular activity prediction across diverse biochemical datasets. Their consistent superiority, particularly when used with ensemble methods, underscores the strength of substructural descriptors in encoding biochemically meaningful patterns. Tree-based models, such as Random Forest and Extra Trees, as well as linear models like Logistic Regression and SVM, consistently leveraged fingerprint features to achieve high predictive accuracy and robust performance.

The added value of graph-theoretic features was found to be highly context dependent. Ensemble algorithms demonstrated a notable ability to leverage the complex and potentially redundant feature spaces resulting from the combination of fingerprints and topological descriptors. In contrast, simpler linear models, such as Logistic Regression, often showed diminished performance with feature augmentation, likely due to overfitting or difficulty in isolating informative signals in high-dimensional spaces. These findings indicate that naïvely concatenating structural descriptors with fingerprints may introduce noise, underscoring the need for feature selection or dimensionality reduction to mitigate redundancy.

From a practical standpoint, the trade-off between accuracy and computational cost remains an important consideration. Models trained on low dimensional node attributes, which were only available for the AIDS dataset, delivered substantial speed advantages, often completing in under a second, making them appealing for large scale screening or rapid prototyping. Fingerprint-based models, by contrast, incurred higher computational demands, while delivering comparable classification performance.

The integration of graph-theoretic and fingerprint-based features improved predictive accuracy for the best-performing model on two out of four datasets, and reduced performance variability (standard deviation) in 7 out of 11 models on the DHFR dataset. This suggests that feature diversity can stabilize learning by mitigating overfitting to dataset specific noise and amplifying generalizable signals. Indeed, the observed shift in top performing model types, for instance, from SVM to AdaBoost in AIDS and from Logistic Regression to Extra Trees in DHFR, demonstrates the particular suitability of ensemble methods for navigating rich, redundant feature landscapes.

Yet, feature augmentation is not universally beneficial. In PTC, combining features led to lower accuracy, despite a reduction in variance. This pattern points to a regularization effect: while the model generalized more consistently, the added structural features may have introduced irrelevant or conflicting information, ultimately lowering the accuracy ceiling.

Taken together, these findings suggest a rational modeling workflow: using node augemented features when available, else prioritize fingerprints as input features. Ensemble or linear models should be trained as reliable general purpose baselines. Graph-theoretic features should be considered, when performance is low or standard deviation exceeds an acceptable threshold. Rather than exhaustive model testing, a targeted, chemically informed selection of features and architectures can achieve both high accuracy and stability while managing computational cost.

B. Performance Differences Between Models Across Datasets

The observed differences in model performance between ML and GNN models can be attributed to dataset specific factors such as the magnitude of structural imbalance, sample size, and class balance. Cohen's d was chosen as metric informing structural difference between classes over significance measures like t-test p-value or U statistic because it provides an interpretable and sample size independent measure of practical class separability.

In AIDS, both structural and class imbalances are extreme (Cohen's $d > 1.8$ for multiple features), creating a regime where simple linear models with chemically informative fingerprints are close to solving the task. GraphConv substantially outperforms GCNConv, which can be attributed to its architectural design. By removing neighborhood normalization and incorporating skip connections, GraphConv is able to preserve both discriminative local and global signals that are otherwise oversmoothed in GCNConv. However, feature-based ML still outperforms GraphConv in this regime suggesting that, where structure is highly separable, the extra complexity of deep models is unnecessary.

In DHFR, moderate structural imbalance (Cohen's d for SPID = 0.31) and balanced classes allow the flexibility of GraphConv to approach, but not surpass, the performance of classical models with fingerprints. GCNConv's poorer results could be attributed to the same smoothing effects, especially in the context of moderate heterogeneity.

For Mutagenicity, minimal class separability (Cohen's $d <$ 0.1 for SPID) can potentially explain why all models display a similar performance, with GraphConv having a small edge that may reflect its ability to capture higher order or multihop patterns not present in linear features.

Finally, PTC_MR, characterized by low sample size and near structural balance (Cohen's d for SPID = 0.15), demonstrates the limits of deep learning architectures when meaningful structural signals are absent and data is scarce. Here, classical models outperform Graph models in accuracy and ROC-AUC.

Taken together, these results reinforce that careful model selection and feature engineering must be aligned to dataset specific characteristics. GNNs excel when there is moderate, non trivial structure to exploit, but classical ML dominates when class separation is easy or data is scarce. Further, experiments suggest that GraphConv is better suited for the type of graph classification that is examined in this study. This is indicated by a modest to high increase in discriminative power in all datasets where the GNNs are able to generate meaningful predictions.

C. Comparison to State Of The Art Models

Figure 5 compares our best ML and GNN models to the State Of The Art (SOTA) models, see Table I, on all four datasets. While the SOTA models outperform our models, our models show competitive classification power on Mutagenicity and DHFA, specifically considering that we did not finetune our models. Surprisingly, the classical machine learning model outperformed the state-of-the-art method, achieving an impressive accuracy of 99.85%. However, further investigation is required to determine whether this result may be attributed to an error in the experimental pipeline.

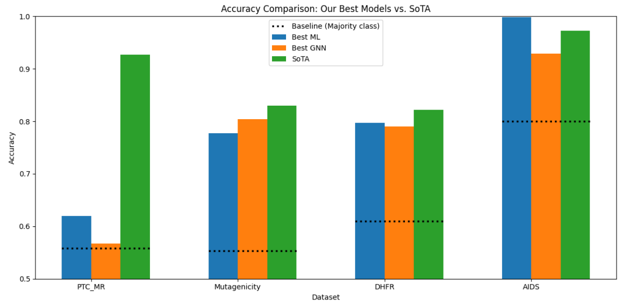


Fig. 5: Accuracy of our best ML and GNN models compared to state-of-the-art GNNs across datasets.

VI. Limitations

While our findings suggest fingerprints and classical machine learning models as the superior choice for molecule classification it is important to note that our comparison of different models, specifically GNNs, is limited. Given the scope of this project, only GCNs were considered, and only two types of convolutional layers tested. However, there exists a vast variety of GNN models (GAT, GIN, GAN and others) and convolutional layers that potentially increase classification performance on these datasets. Computational resources limited our ability to train GNNs by restricting the model size and number of epochs that could be used. Furthermore, our dataset selection considered computational costs, which impacted our ability to incorporate more and larger datasets. While our results are consistent across the selected datasets, this study was conducted on a limited number of datasets, with the highest class imbalance ratio being 1:4. However, many real-world molecular datasets exhibit even greater imbalance, often exceeding ratios of 1:9. Although this constraint was appropriate given the scope of our work, more severe imbalance may pose significant challenges for classical machine learning models.

VII. Future Work

Naturally, future work can aim to address the limitations of this study. One promising direction is the inclusion of more advanced GNN architectures and the fine-tuning of their performance. Moreover, the marginal utility observed from combining features for classical ML models suggests further investigation, such as applying dimensionality reduction techniques to optimize the combined feature space. Building directly on this study, future research could also examine the outlier performances observed on the PTC and AIDS datasets. This includes fine-tuning models to achieve state-of-the-art performance on the PTC dataset, or investigating the underlying factors contributing to the unexpectedly high performance on the AIDS dataset.

VIII. Conclusion

This study presents a novel comparison of three different paradigms; (1) graph-theoretic descriptors, (2) molecular fingerprints, and (3) graph neural networks, for graph classification on biomedical datasets that could act as a guide for initial model selection and screening for active compounds. Our results indicate that ML approaches on molecular fingerprints often outperforms or delivers comparable results to simple Graph Neural Networks and can even provide competitive accuracy to state of the art models on most datasets, while only requiring a fraction of the compute. Whereas GT metrics offer statistically significant insights into structural distribution, they appear insufficient for high performing classification when used in isolation. Overall, the methodological progression from targeted significance testing to comprehensive, cross-dataset structural analyses highlights the rigor and depth of this evaluation.

Our general recommendations drawn from this study is to de-prioritise GT features and instead focus on molecular fingerprints, and use node attributes if available. In the light of computational constraints, we suggest to start with ML models, specifically SVM, Logistic Regression and Random Forest, and only consider GNNs if time allows for extensive finetuning.

References

[1] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. *arXiv preprint arXiv:1111.4570*, 2012.

[2] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

[3] Claudio N. Cavasotto and Valeria Scardino. Machine learning toxicity prediction: Latest advances by toxicity end point. *ACS Omega*, 7(51):47536–47546, 2022. doi: 10.1021/acsomega.2c05693.

[4] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2019. doi: 10.1109/TKDE.2018.2849727.

[5] Vivek Gadiya, Poonam Nagrath, Avinash K Dubey, and Piyush Jain. Screening of antibacterial compounds with novel structure from the fda approved drugs using machine learning methods. *Computers in Biology and Medicine*, 146:105641, 2022. doi: 10.1016/j.compbiomed.2022.105641.

[6] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[7] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[8] Christoph Helma, Tobias Cramer, Stefan Kramer, and Luc De Raedt. Predictive toxicology: benchmark data sets and prediction models of mutagens and nonmutagens. *Journal of chemical information and computer sciences*, 41(5):1158–1165, 2001.

[9] Christoph Helma, Verena Schöning, Juergen Drewe, and Peter Boss. A comparison of nine machine learning mutagenicity models and their application for predicting pyrrolizidine alkaloids. *Frontiers in Pharmacology*, 12: 708050, 2021. doi: 10.3389/fphar.2021.708050. URL https://doi.org/10.3389/fphar.2021.708050.

[10] Asela Hevapathige and Qing Wang. Permutation-invariant graph partitioning:how graph neural networks capture structural interactions?, 2025. URL https://arxiv.org/abs/2312.08671.

[11] Saiful Islam, Md. Nahid Hasan, and Pitambar Khanra. A structural feature-based approach for comprehensive graph classification, 2024. URL https://arxiv.org/abs/2408.05474.

[12] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13:1–23, 2021.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

[14] Nils Kriege and Petra Mutzel. Subgraph matching kernels for attributed graphs, 2012. URL https://arxiv.org/abs/1206.6483.

[15] Greg Landrum. RDKit: Open-source cheminformatics. http://www.rdkit.org, 2013. Accessed: 2025-05-14.

[16] Hang Li, Xiangnan He, Jiayi Huang, and Zheng-Jun Zha. Graph classification via topological and label attributes. *Advances in Neural Information Processing Systems*, 24, 2011.

[17] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 33: 4602–4609, 2019.

[18] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs, 2020. URL https://arxiv.org/abs/2007.08663.

[19] Dai Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. Universal graph transformer self-attention networks. In *Companion Proceedings of the Web Conference 2022*, pages 193–196, 2022.

[20] Álmos Orosz, Károly Héberger, and Anita Rácz. Comparison of descriptor-and fingerprint sets in machine learning models for adme-tox targets. *Frontiers in Chemistry*, 10:852893, 2022.

[21] Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In Niels da Vitoria Lobo, Takis Kasparis, Fabio Roli, James T. Kwok, Michael Georgiopoulos, Georgios C. Anagnostopoulos, and Marco Loog, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-89689-0.

[22] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[23] Shubhajit Roy, Hrriday Ruparel, Kishan Ved, and Anirban Dasgupta. Fit-gnn: Faster inference time for gnns using coarsening, 2025. URL https://arxiv.org/abs/2410.15001.

[24] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. In *Advances in neural information processing systems*, volume 24, 2011.

[25] Nikil Wale, David Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

[26] Ningning Wang, Xinliang Li, Jing Xiao, Shao Liu, and Dongsheng Cao. Data-driven toxicity prediction in drug discovery: Current status and future directions. *Drug Discovery Today*, 29(11):104195, 2024. doi: 10.1016/j.drudis.2024.104195.

[27] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684): 440–442, 1998.

[28] Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, et al. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature communications*, 14(1):2585, 2023.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[30] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370–3388, 2019.

[31] Xin Yang, Yifei Wang, Ryan Byrne, Gisbert Schneider, and Shengyong Yang. Concepts of artificial intelligence for computer-assisted drug discovery. *Chemical Reviews*, 119(18):10520–10594, 2019. doi: 10.1021/acs.chemrev.8b00728.

[32] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.

[33] Lei Zhang, Jinfeng Tan, Dongmei Han, Qili Zhu, and Yudong Li. Using molecular fingerprints as descriptors for supervised machine learning of chemical properties. *Scientific reports*, 9(1):1–10, 2019.

[34] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. Hierarchical graph pooling with structure learning, 2019. URL https://arxiv.org/abs/1911.05954.

[35] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81, 2020. doi: 10.1016/j.aiopen.2021.01.001. URL https://doi.org/10.1016/j.aiopen.2021.01.001.

APPENDIX

	Model	Fingerprint Accuracy	Graph Metric Accuracy	Delta (Fingerprint - Graph Metric)
0	Logistic Regression (LR)	0.7976	0.7051	0.0925
1	Linear Discriminant Analysis (LDA)	0.7091	0.6931	0.0160
2	K-Nearest Neighbor (KNN)	0.7805	0.7156	0.0649
3	Decision Tree (CART)	0.7554	0.6971	0.0583
4	Naïve Bayes (NB)	0.7090	0.5621	0.1469
5	Support Vector Machine (SVM)	0.7884	0.6640	0.1244
6	AdaBoost (AB)	0.7778	0.6733	0.1045
7	Gradient Boosting (GBM)	0.7871	0.7182	0.0689
8	Random Forest (RF)	0.7924	0.7368	0.0556
9	Extra Trees (ET)	0.7752	0.7130	0.0622
10	Multi-Layer Perceptron (MLP)	0.7924	0.6773	0.1151

TABLE VI: Accuracy comparison between Fingerprint-based and Graph Metric-based models on DHFR dataset.

A. Model Runtime Comparison

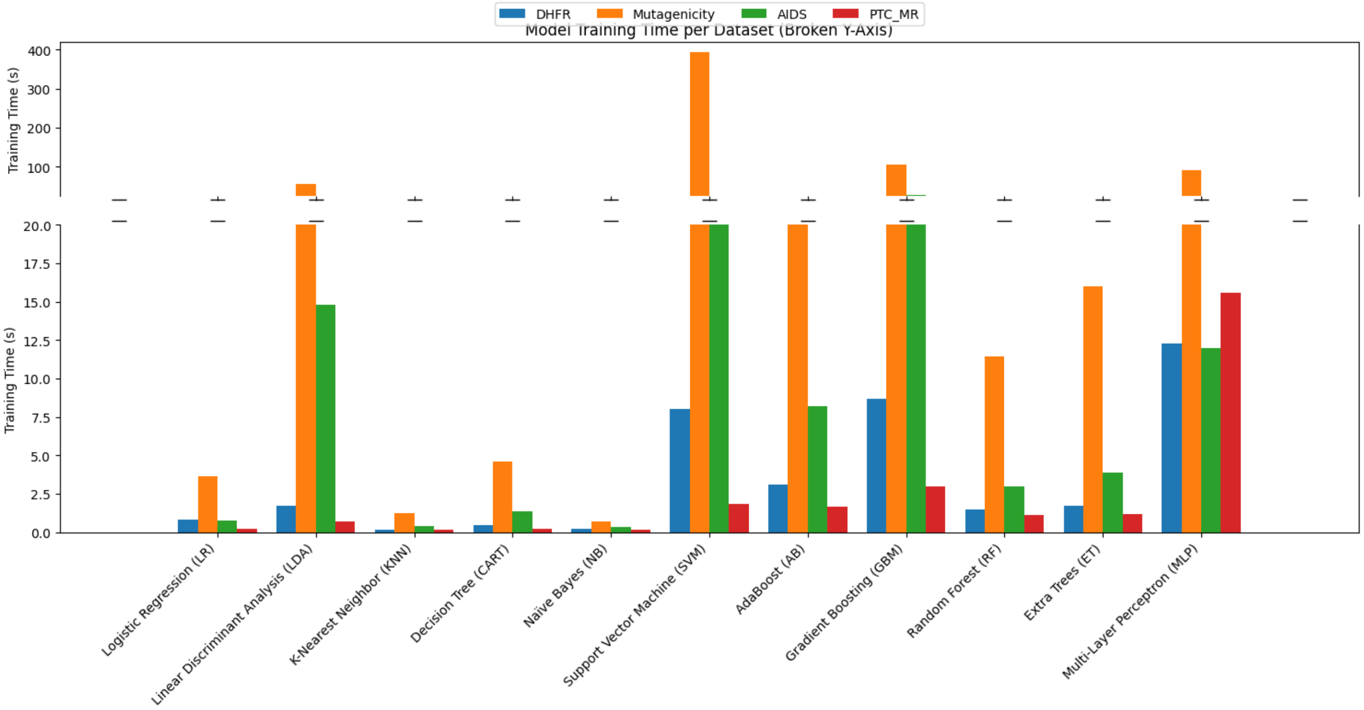


Fig. 6: Training time (in seconds) for each ML model across the three feature sets: Graph-theoretic, Fingerprint, and Combined.

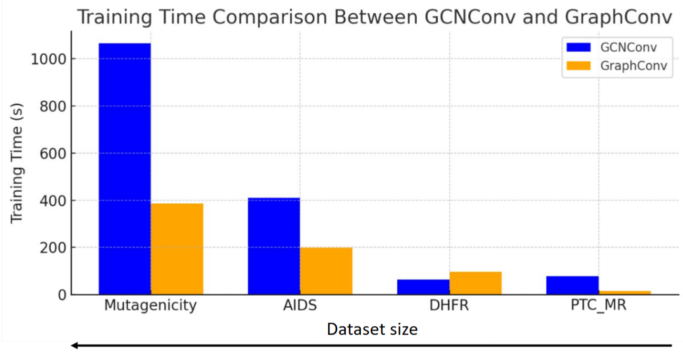


Fig. 7: Training time (in seconds) for each GNN model

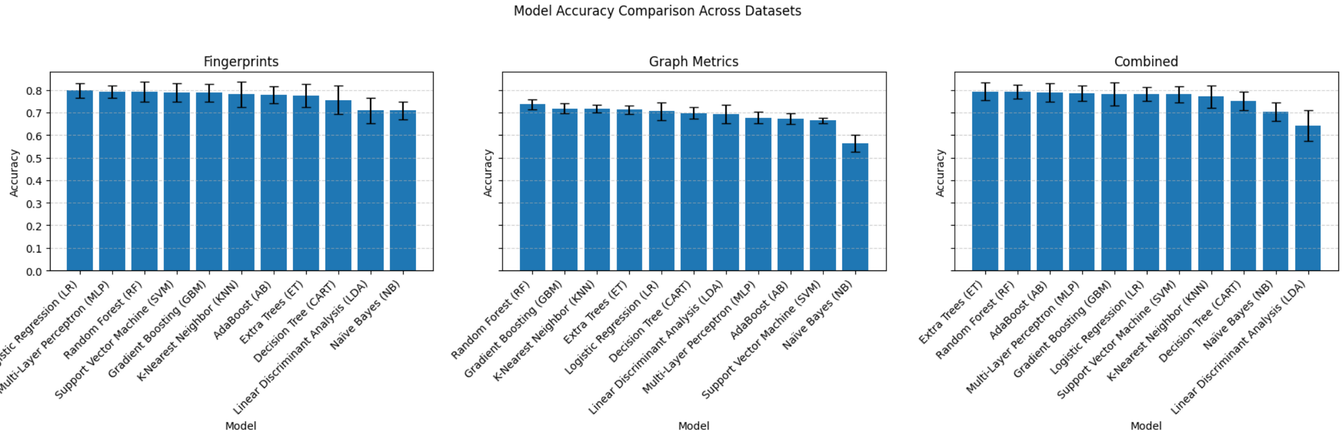


Fig. 8: Accuracy across all feature inputs for each model on the DHFR dataset.

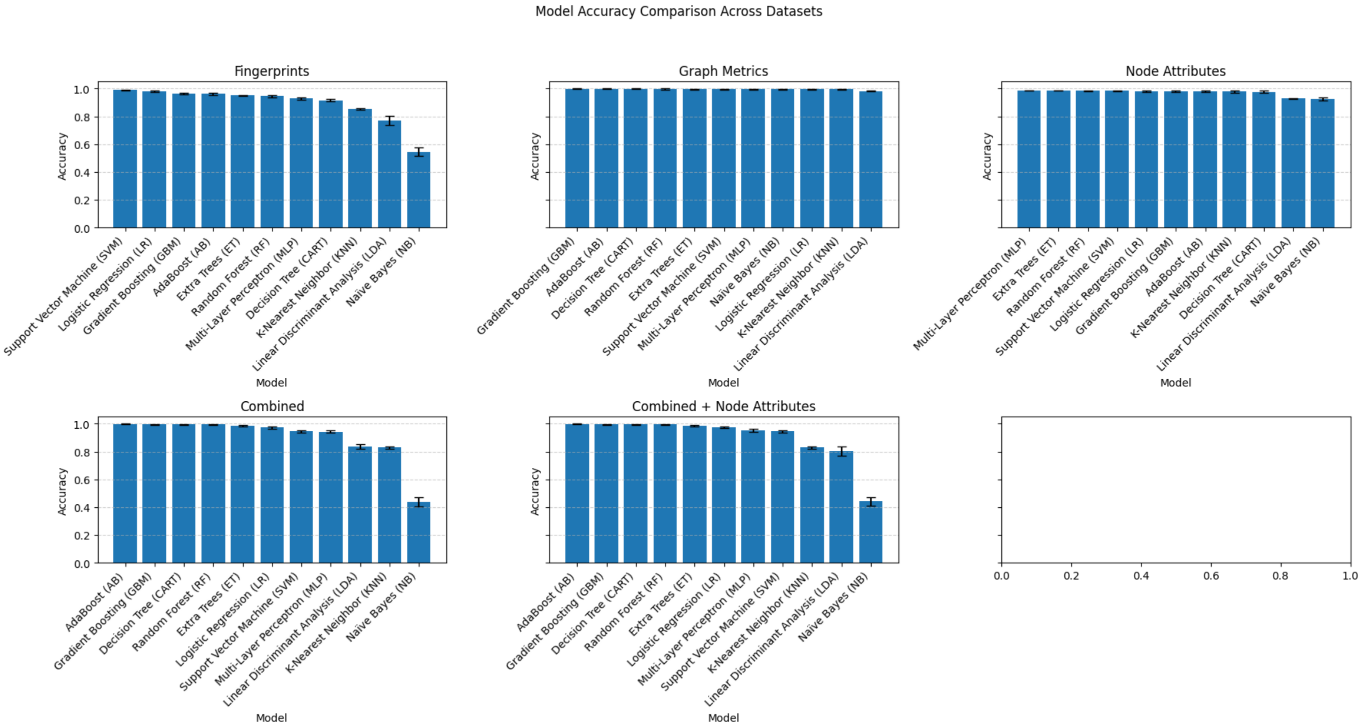


Fig. 9: Accuracy across all feature inputs for each model on the AIDS dataset.

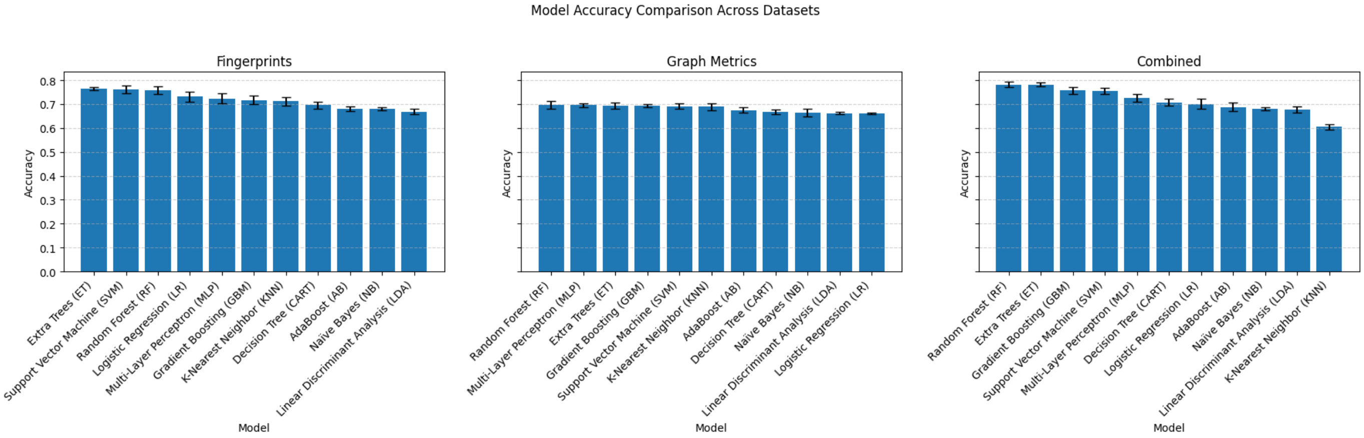


Fig. 10: Accuracy across all feature inputs for each model on the Mutagenicity dataset.

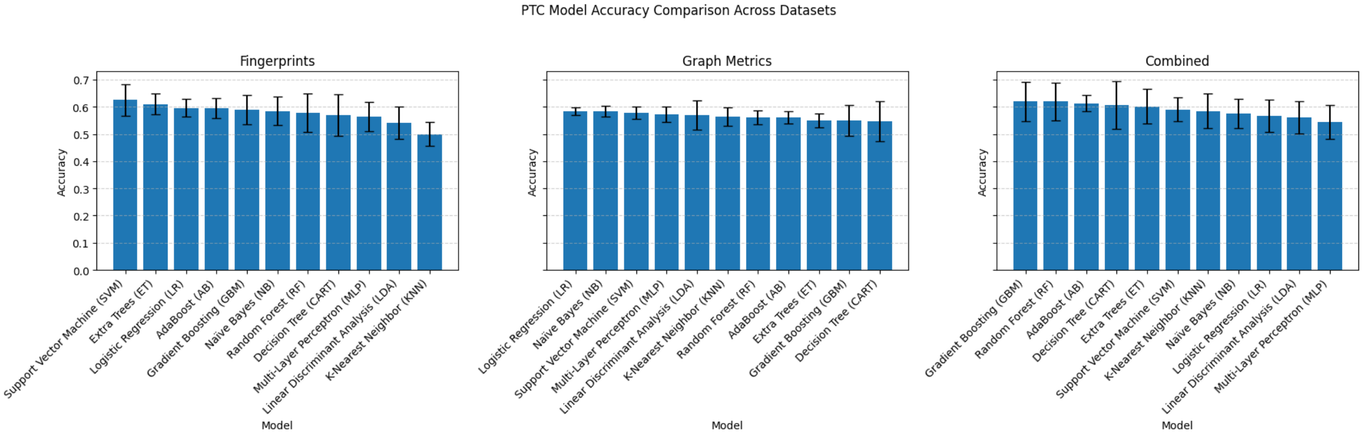


Fig. 11: Accuracy across all feature inputs for each model on the PTC_MR dataset.