Framework GNN-AID Graph Neural Network Analysis, Interpretation and Defense

Kirill Lukianov[1, 2, 3], **Mikhail Drobyshevskiy**[1, 2], **Georgii Sazonov**[2, 4], **Mikhail Soloviov**[2], **Ilya Makarov**[1, 5]

[1] ISP RAS Research Center for Trusted Artificial Intelligence Moscow, Russia
[2] Ivannikov Institute for System Programming of the Russian Academy of Sciences, 109004 Moscow, Russia
[3] Moscow Institute of Physics and Technology (National Research University), 141700 Moscow, Russia
[4] Lomonosov Moscow State University, Leninskie Gory, 1, Moscow, 119991, Russia
[5] AIRI, 121170 Moscow, Russia
{lukianov, drobyshevsky, sazonovg}@ispras.ru, m.solovev@phystech.edu, iamakarov@hse.ru

Abstract

The rising demand for Trusted AI (TAI) underscores the need for interpretable and robust models, yet existing tools rarely support graph-structured data or integrate interpretability with security. At the same time, Graph Neural Networks (GNNs) deliver state-of-the-art performance on numerous graph tasks.

We present GNN-AID (Graph Neural Network Analysis, Interpretation, and Defense), an open-source Python framework for analyzing, interpreting, and defending GNNs, addressing this critical gap. Built on PyTorch-Geometric, GNN-AID offers preloaded datasets, model libraries, flexible APIs, and a web interface for visualization and no-code model design. MLOps features further support reproducibility and experiment tracking.

GitHub repo: — https://github.com/ispras/GNN-AID
YouTube video: — https://youtu.be/uHxaxLSQ9JM

Introduction

Trustworthiness, encompassing interpretability and security, has become a central concern in modern AI. Extensive research has introduced interpretation methods across vision, NLP, and tabular domains (Burkart and Huber 2021), alongside a growing body of adversarial attacks and defenses (Tian et al. 2022). Developers increasingly rely on unified frameworks that consolidate these functionalities. However, most widely used interpretability libraries such as AI Explainability 360 (Arya et al. 2021) are not designed for Graph Neural Networks (GNNs), while attack and defense libraries like ART (Nicolae et al. 2018) require substantial adaptation to handle graph-structured data.

Meanwhile, GNNs have achieved state-of-the-art results across diverse graph-based applications, yet supporting tools remain fragmented. Graph-specific libraries provide only partial functionality: PyTorch Geometric (PyG) (Fey and Lenssen 2019) and DGL (Wang et al. 2019) offer limited interpretability; DIG (Liu et al. 2021) expands explainability; DeepRobust (Li et al. 2020) and GreatX (Li et al. 2022) focus on robustness; GraphFramEx (Amara et al. 2022) and CLARUS (Metsch et al. 2024) target explainability, though CLARUS appears inactive. As summarized in Table 1, no framework unifies interpretation, attack, and defense for GNNs, and almost none provide a graphical interface that lowers the entry barrier for practitioners.

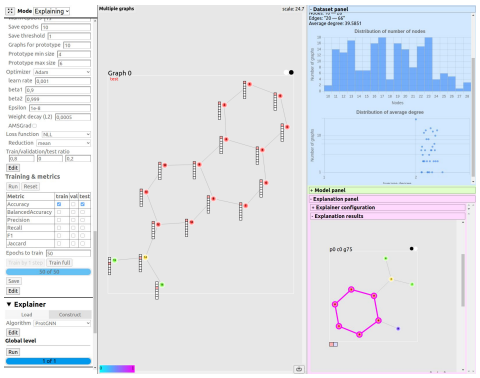

Figure 1: Web interface of GNN-AID in interpretation mode. The left panel handles the dataset and training setup, the center shows a MUTAG graph with node-level explanations, and the right presents statistics and a global ProtGNN explanation. The interface unifies interpretability, robustness, and defense, supporting analysis of trade-offs between explainability and security.

To address these gaps, we present GNN-AID (Analysis, Interpretation, and Defense for GNNs), an open-source framework that integrates robustness, interpretability, and adversarial analysis into a single environment. The system combines an extensible Python backend with a web-based interface (Figure 1), supporting both research workflows and practical applications.

The main contributions and novelty of this work are as follows:

- We introduce a unified framework that integrates the study of attacks, defense mechanisms, and interpretability techniques for Graph Neural Networks.
- We demonstrate how the framework can facilitate inter-

Name	Int.	Attacks	Defenses	GUI
PyG	5	-	-	-
DGL	1	-	-	-
DIG	8	-	-	-
DeepRobust	-	12	13	-
GreatX	-	18	17	-
GraphFramEx	15	-	-	-
CLARUS	4	-	-	+
GNN-AID	10	12	9	+

Table 1: Comparison of GNN analysis tools by number of algorithms. GNN-AID uniquely combines interpretability, attack, and defense features with a user-friendly GUI. Int. – Interpretation

disciplinary analysis at the intersection of interpretability and security, enabling systematic exploration of trade-offs and synergies between these dimensions.

- We provide an automated pipeline that supports both backend execution and user-friendly GUI-based interaction, making the framework accessible to both researchers and practitioners.

System Overview

Architecture and pipelines

Figure 2 illustrates the main general framework pipeline. Additional information about these steps is provided below.

Dataset Processing GNN-AID datasets consist of a graph (or set) and variable components: features, labels, and task. This design supports flexible task switching (e.g., node vs. edge prediction). Features are numerical encodings of raw attributes such as age or atom type.

Model and Robustness Modules Models are built from modular layers (convolutions, activations, normalization, pooling, dropout, skip connections). The model manager handles training, persistence, and robustness functions. Attacks and defenses (evasion, poisoning, privacy) share a unified API with base `Attacker`/`Defender` classes. Methods may alter data, gradients, or architecture; extension requires subclassing base classes.

Interpretation Module Interpretability is implemented through an `Explainer` interface that standardizes adding new methods, storing results, and computing metrics such as fidelity. Post-hoc and self-interpretable approaches are supported after training.

Frontend A browser-based interface mirrors backend functionality:

1. Graph analysis with visualized dynamics, outputs, predictions, and metrics;

2. Explanation via post-hoc or self-interpretable models;

3. Robustness eval with configurable attacks & defenses.

System Integration and Functional Summary The framework's modular PyTorch-based architecture separates core logic from interpretation, attack, and defense modules, enabling independent extension. An abstraction layer allows integration of external components.

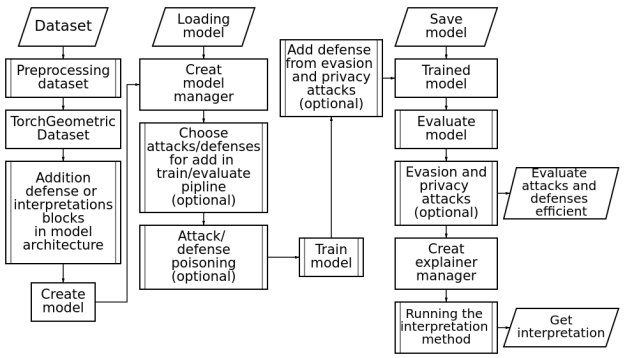

Figure 2: GNN-AID main pipeline. The framework supports all core aspects of interpretability, attacks, and defenses within a unified pipeline, enabling both isolated analysis and systematic investigation of their interactions, which constitutes the main contribution of this work.

MLOps features include experiment tracking, centralized model registry, automated CI/CD, and structured storage of datasets, checkpoints, and explanations. Combined with the web interface, these ensure scalability, reproducibility, and accessibility for research and practice in trustworthy GNNs.

Use Cases

GNN-AID serves a broad audience, from students to researchers, by supporting learning, development, and experimentation with Graph Neural Networks (GNNs). Core use cases include:

1. **Research**. Enables systematic studies of interpretability robustness interactions through diverse attacks, defenses, and explanation methods, highlighting trade-offs such as reduced fidelity under defense. Integrated MLOps tools ensure reproducibility and result tracking.
2. **Development**. Provides a modular API for custom models, datasets, and evaluation routines. Built-in interpretation, robustness testing, adversarial analysis, and interactive visualizations streamline the development and debugging process.
3. **Education**. Offers an intuitive environment for exploring GNNs with visualizations of training dynamics, parameter updates, features, and predictions, making it effective for classrooms, workshops, and tutorials.

Conclusion and Future Work

We developed GNN-AID, a framework that provides tools to analyze, interpret, and defend Graph Neural Networks with emphasis on Trusted AI principles, interpretability, and robustness. Combining user-friendly no-code options, API access, visualization, and MLOps support, it offers a comprehensive solution for researchers and developers.

GNN-AID promotes both scientific exploration and practical applications in graph analysis, thereby contributing to the principles of Trusted AI.

Acknowledgments

This work was supported by a grant provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025, No. 139-15-2025-011.

References

Amara, K.; Ying, Z.; Zhang, Z.; Han, Z.; Zhao, Y.; Shan, Y.; Brandes, U.; Schemm, S.; and Zhang, C. 2022. GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks. In *The First Learning on Graphs Conference*.

Arya, V.; Bellamy, R. K.; Chen, P.-Y.; Dhurandhar, A.; Hind, M.; Hoffman, S. C.; Houde, S.; Liao, Q. V.; Luss, R.; Mojsilović, A.; et al. 2021. Ai explainability 360 toolkit. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, 376–379.

Burkart, N.; and Huber, M. F. 2021. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70: 245–317.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Li, J.; Wu, B.; Hou, C.; Fu, G.; Bian, Y.; Chen, L.; Huang, J.; and Zheng, Z. 2022. Recent advances in reliable deep graph learning: Inherent noise, distribution shift, and adversarial attack. *arXiv preprint arXiv:2202.07114*.

Li, Y.; Jin, W.; Xu, H.; and Tang, J. 2020. Deeprobust: A pytorch library for adversarial attacks and defenses. *arXiv preprint arXiv:2005.06149*.

Liu, M.; Luo, Y.; Wang, L.; Xie, Y.; Yuan, H.; Gui, S.; Yu, H.; Xu, Z.; Zhang, J.; Liu, Y.; Yan, K.; Liu, H.; Fu, C.; Oztekin, B. M.; Zhang, X.; and Ji, S. 2021. DIG: A Turnkey Library for Diving into Graph Deep Learning Research. *Journal of Machine Learning Research*, 22(240): 1–9.

Metsch, J. M.; Saranti, A.; Angerschmid, A.; Pfeifer, B.; Klemt, V.; Holzinger, A.; and Hauschild, A.-C. 2024. CLARUS: An interactive explainable AI platform for manual counterfactuals in graph neural networks. *Journal of Biomedical Informatics*, 150: 104600.

Nicolae, M.-I.; Sinn, M.; Tran, M. N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. 2018. Adversarial Robustness Toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*.

Tian, Z.; Cui, L.; Liang, J.; and Yu, S. 2022. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8): 1–35.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.