Graph-Based Feature Engineering for DeFi Security Incident Severity Prediction

Daria Pavlova*
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
daria.pavlova@mps.si

Inna Novalija
Jožef Stefan Institute
Ljubljana, Slovenia
inna.koval@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Abstract

Decentralized Finance (DeFi) has emerged as a rapidly growing sector, but it has been plagued by numerous security incidents resulting in billions of USD in losses [1]. An important challenge is predicting which security incidents will lead to *severe* financial losses, as this can inform risk management and mitigation strategies. In this paper, we present a novel approach that integrates a semantic knowledge graph of the DeFi ecosystem into the machine learning pipeline for incident severity prediction. We construct a knowledge graph capturing rich relationships between DeFi protocols (including protocol fork lineage, multi-chain deployments, and historical incidents), and we engineer graph-based features from this graph to augment traditional incident features. Using these features in a classification model, we predict whether an incident will cause above-threshold (severe) losses. Our results show that incorporating graph-based features yields a substantial improvement in predictive performance: the model with semantic graph features achieves an Area Under ROC Curve (AUC) of 0.787, a 31.6% relative increase over the baseline model using only non-graph features. We observe particularly large gains in precision (from 0.341 to 0.490), indicating a significantly reduced false alarm rate. The findings demonstrate the practical value of graph-enriched feature engineering for security analytics in DeFi. This work provides new insights into how protocol interconnections and characteristics contribute to incident severity, opening avenues for more robust DeFi risk assessment tools.

Keywords

Decentralized Finance; DeFi; Security; Knowledge Graph; Feature Engineering; Incident Severity Prediction

1 Introduction

Decentralized Finance (DeFi) platforms have experienced rapid growth, alongside a surge in security breaches such as hacks and exploits. In 2022 alone, crypto attacks led to over $3.8 billion in stolen assets, with the majority coming from DeFi protocol exploits [1]. These incidents vary widely in impact: while many attacks result in limited losses, a significant fraction escalate into catastrophic failures causing losses in the tens or hundreds of millions of dollars. Predicting which security incidents will become *severe* (high-loss) events is crucial for proactive risk management, insurance underwriting, and developing early warning systems for the DeFi ecosystem. Prior research has analyzed DeFi vulnerabilities and attack taxonomy [2], and industry reports highlight the growing scale of DeFi hacks. However, there is a gap in predictive approaches: existing studies focus on identifying vulnerabilities or classifying attack types, rather than forecasting the *severity level* of an incident before it fully unfolds. In traditional cybersecurity contexts, incorporating relational context via knowledge graphs and network models has been shown to improve threat detection [4]. For example, graph-based severity triage using attack graphs has been studied in traditional cybersecurity [7]. In this work, we propose a novel graph-based feature engineering approach to address this challenge. We construct a semantic **knowledge graph** of the DeFi ecosystem that encodes domain knowledge: nodes represent entities such as protocols and incidents, and edges capture relationships like *"forked-from"* (denoting protocol lineage) and *"deployed-on"* (connecting protocols to blockchain platforms), among others. From this knowledge graph, we derive a set of **graph-based features** for each security incident. These features quantify properties such as a protocol's structural position in the ecosystem (e.g., number of fork "children," cross-chain deployments, past incident count), which we posit are predictive of how severe an incident could be. We integrate these semantic graph features with conventional features (e.g., time of incident, incident type categories) in a machine learning classifier to predict whether an incident's loss will exceed a severity threshold. The contributions of our work are as follows:

- We introduce a methodology to incorporate a DeFi-specific knowledge graph into security incident severity prediction.
- We demonstrate significant performance gains over a baseline model lacking graph features (improving AUC by 31.6% and F1-score by 25%).
- We provide a comprehensive analysis including case studies. For example, Figure 2 shows a knowledge-graph subnetwork for Convex Finance, illustrating how related protocol dependencies can influence risk.
- We discuss practical implications of our findings for improving DeFi risk assessment.

All code and data for this work are available in an open-source repository [5].

2 Methodology

2.1 Knowledge Graph Construction

We built a knowledge graph representing the DeFi ecosystem to serve as a basis for feature engineering. The graph's schema defines several entity types and relations relevant to DeFi security:

*First author and presenter.

IS2025, Ljubljana, Slovenia

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

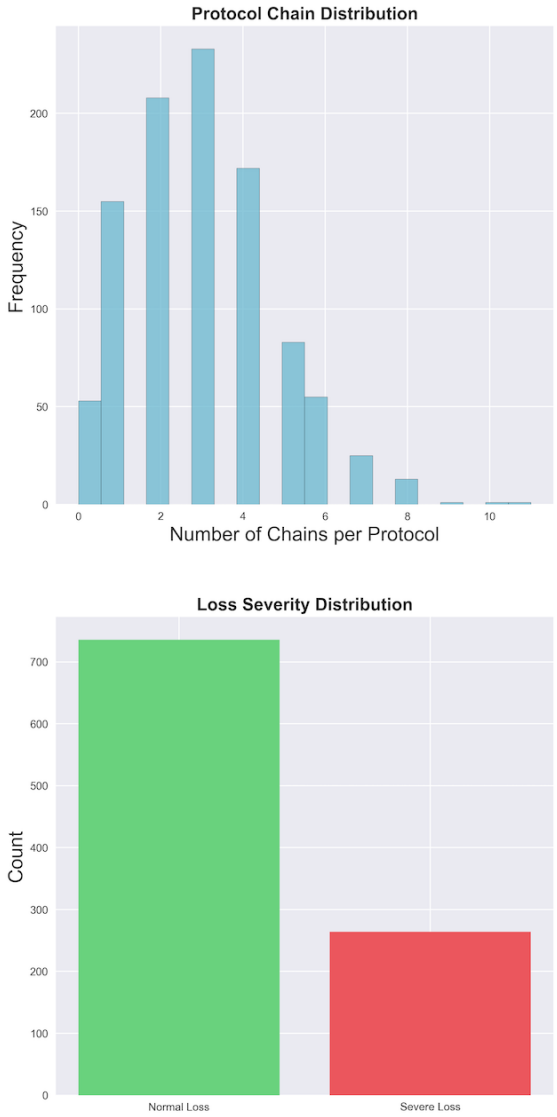

Figure 1: DeFi knowledge graph overview: protocols, blockchains, and incidents with relations (forked-from, deployed-on, involves).

- **Protocol nodes:** Each DeFi protocol (e.g., lending platform, DEX, yield aggregator) is a node. Attributes include protocol name and launch date.
- **Incident nodes:** Major recorded security incidents (hacks, exploits) are represented as nodes with attributes such as date, loss amount, and qualitative classification (e.g., flash loan, smart contract bug).
- **Blockchain nodes:** Blockchain platforms (Ethereum, Binance Smart Chain, etc.) are included to capture deployment contexts.

Key relationships are encoded as directed edges:

- *Fork-of:* Connects a protocol to the protocol it was forked from (if applicable), capturing lineage (e.g., SushiSwap → Uniswap).
- *Deployed-on:* Links a protocol to a blockchain platform on which it is deployed.
- *Incident-involves:* Links an incident node to the protocol(s) affected by that incident.

Figure 2: Convex-centric subgraph. Dependency on Curve highlights potential severity propagation via upstream vulnerabilities.

The knowledge graph was constructed by integrating multiple data sources on DeFi projects and security events, ensuring semantic consistency in how entities and relations are defined. The resulting graph captures a rich hierarchical structure of protocol relationships (including parent–child fork trees and cross-chain deployment links), as well as the association of past incidents with protocols. We use Neo4j to store and query this graph.

2.2 Feature Engineering with Graph-Based Features

From the knowledge graph, we derived several quantitative features that characterize the structural and historical context of the protocol involved in a given incident:

- **Protocol multi-chain count**: the number of distinct blockchains on which the protocol is deployed (degree of *deployed-on* edges). A higher count indicates a widely deployed protocol, potentially implying larger user bases or attack surfaces.
- **Fork lineage indicators**: whether the protocol is a fork of another (*has parent*) and the number of forks derived from it. These capture if a protocol inherits code (and possibly vulnerabilities) from a parent and how prevalent its code is in offspring projects.
- **Past incident count**: the total number of past security incidents involving the protocol (count of *incident-involves* edges to prior incidents). A history of frequent past incidents might signal underlying security weaknesses or attractive target value.

In addition to these graph-derived features, we include conventional features for each incident:

- **Temporal features**: the year and month of the incident, and day-of-week if relevant, to capture any time-related patterns or trends in attack occurrence.
- **Categorical features**: the general type of attack or vulnerability exploited (e.g., reentrancy, price oracle manipulation), and the asset or protocol category targeted, which provide contextual information on the incident.

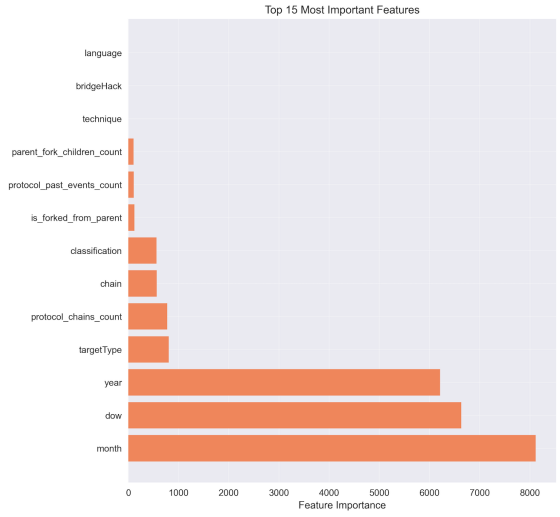

Figure 3: Workflow: derive graph-based features from the DeFi knowledge graph and combine with conventional incident features for classification.

All features are computed or retrieved at the time just before the incident (to avoid using any post-incident information). The combination of graph-based features with traditional features forms the feature vector used for prediction.

2.3 Classification Model and Training

We frame incident severity prediction as a binary classification task: *severe* vs. *non-severe* loss outcome. Following prior work, we define a severe incident as one with loss exceeding a high quantile threshold of the loss distribution. In our dataset, the threshold was the 75th percentile loss of approximately $2.21 million, which resulted in 402 incidents labeled as severe out of 1608 total incidents. Our model is a gradient boosting decision tree classifier (LightGBM [3]), chosen for its efficiency and ability to handle heterogeneous feature types. We enable LightGBM's built-in handling for class imbalance (`is_unbalance=True`) given that severe incidents are the minority class (25%). The model is trained on historical incidents with known outcomes. We use a stratified 5-fold cross-validation on the training set for model selection and to assess consistency, and finally evaluate on a held-out test set. We compare two feature sets: a **Baseline** model using only the non-graph features (temporal and categorical features, which include basic incident information), and a **Semantic Graph** model using the full feature set including the graph-based features described above. Model performance is evaluated primarily with Area Under the ROC Curve (AUC) to measure ranking performance, and we also report Precision, Recall, and F1-score to understand classification effectiveness.

3 Experiments and Results

3.1 Dataset and Experimental Setup

We compiled a dataset of 1,608 DeFi security incidents that occurred between 2020 and 2025, drawing on public reports and databases of crypto hacks (e.g., Rekt and other industry sources). Each incident record includes the loss amount (in USD) and details such as date and attack type. As described above, incidents with losses above $2.21 million were labeled as *severe*, which yields a severe class prevalence of roughly 25% (402 severe vs.

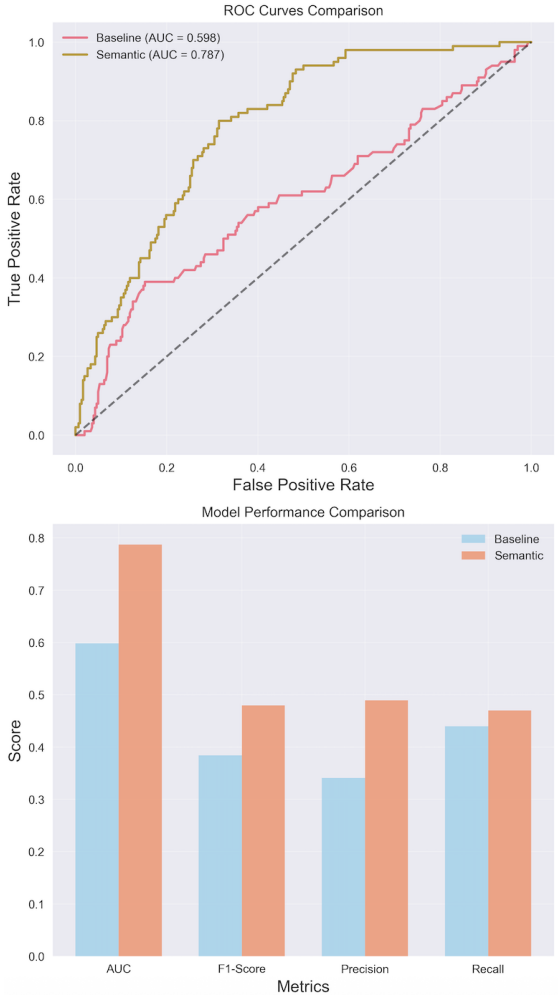

Figure 4: Performance comparison. Top: ROC curves (AUC baseline 0.598 vs. semantic 0.787). Bottom: bar chart for AUC, F1, precision, recall.

1206 non-severe cases). For training and evaluation, we split the data chronologically (to emulate predicting future incidents) into 70% for training (with internal cross-validation for tuning) and 30% for final testing. The classifier was trained with early stopping to prevent overfitting. We ensured that any given DeFi protocol's incidents are distributed across folds to avoid overestimating performance by learning protocol-specific quirks in both train and test.

3.2 Performance Comparison

Our results confirm that incorporating graph-based features markedly improves prediction performance. Table 1 summarizes the evaluation metrics for the baseline and semantic graph-enhanced models on the test set. The Semantic Graph model achieves an AUC of 0.787, substantially higher than the baseline's 0.598 (a relative improvement of 31.6%). This indicates that the model with graph features is much better at ranking incidents by risk. The F1-score also improves from 0.384 to 0.480, reflecting better overall classification accuracy. Notably, the Precision (positive predictive value) rises from 0.341 to 0.490—a 43.7% increase—while Recall increases slightly from 0.440 to 0.470. This suggests that the graph-enriched model is significantly more

Table 1: Performance comparison between the baseline model (numeric/categorical features only) and the semantic graph model (with knowledge graph features). The semantic model shows substantial gains in all metrics.

Metric	Baseline	Semantic Graph	Improvement
AUC	0.598	0.787	+31.6%
F1-score	0.384	0.480	+25.0%
Precision	0.341	0.490	+43.7%
Recall	0.440	0.470	+6.8%

effective at identifying truly severe incidents (fewer false positives) without sacrificing the ability to catch most severe cases. In practical terms, an analyst using the semantic model's predictions would receive far fewer false alarms for every true severe incident detected, which is a crucial improvement for real-world usability. While the absolute values of these metrics might appear moderate, it is crucial to note that they represent a substantial improvement over the baseline and are highly competitive for this specific and challenging prediction task. Figure 4 further illustrates the performance difference. The semantic model's ROC curve (solid line) is consistently above the baseline's (dashed line), indicating better true positive rates at every false positive rate. In addition to the hold-out test, we evaluated stability via cross-validation. The baseline model's mean AUC across 5 folds was 0.629 (std 0.036), whereas the semantic model averaged 0.809 (std 0.027). This not only reaffirms the performance boost but also indicates that the graph-augmented model is more consistent across different data subsets (lower variance), likely because the graph features provide a more robust signal that generalizes.

3.3 Feature Importance Analysis

To understand which features contributed most to the improved predictions, we inspected the feature importance rankings from the trained LightGBM model. The top features included several graph-based ones: notably, the *protocol past incident count* and *protocol multi-chain count* were among the highest contributors. This aligns with intuition—protocols that suffered many past issues or that operate on many chains (hence complex, with larger user pools) were more likely to have severe future incidents. The fork-related features also had significant importance.

4 Discussion

The substantial performance gains achieved by including knowledge graph features underscore the importance of relational context in DeFi security analysis. The knowledge graph captures ecosystem-level dependencies that are not apparent from incident-centric data alone. For instance, our model effectively learned that incidents on protocols deeply integrated into the ecosystem (either through many deployments or many forks) have a higher chance of cascading into severe losses. This reflects a practical reality: a vulnerability in a widely forked codebase or a multi-chain protocol can affect numerous users and liquidity pools, amplifying the impact. Practical applications: Our findings can directly inform DeFi risk assessment practices. Security teams and auditors could use similar graph-based analysis to identify "hot spots" in the ecosystem—protocols whose compromise would likely be particularly damaging. For example, if a protocol is identified as a central hub in the protocol relationship graph (due to being extensively forked or connected), extra scrutiny or precaution for that protocol is justified. Likewise, insurers offering DeFi hack coverage can incorporate these graph-derived risk factors into premium calculations, resulting in pricing that more accurately reflects protocol risk profiles beyond just historical loss statistics. The graph-based approach we used is related to existing work on security graphs and financial graphs. For instance, graph neural networks (GNNs) have been applied for fraud detection in cryptocurrency networks [8, 9] and for risk modeling in financial systems [10–12]. These studies suggest the broader promise of graph analytics in finance. Similarly, recent work has explored cybersecurity knowledge graphs [6].

5 Conclusion

We presented a graph-enriched machine learning framework for predicting the severity of security incidents in the DeFi domain. By engineering features from a semantic knowledge graph of DeFi protocols and their relationships, our model achieved notably better performance than traditional approaches using only flat features. The knowledge graph provided critical context that improved the identification of high-loss incidents, emphasizing that understanding *where* an incident occurs in the DeFi ecosystem is as important as understanding *what* the incident is. These results have immediate implications for improving DeFi security posture. Stakeholders can leverage graph-based analyses to anticipate and mitigate the most dangerous threats. For example, continuous monitoring of the DeFi knowledge graph could help detect vulnerable hubs and alert on incidents that involve those high-risk areas. Our approach can be extended in several directions: incorporating real-time graph updates to predict emerging threats, using more complex network metrics, or integrating our features into GNN models for end-to-end learning. In summary, graph-based feature engineering offers a powerful and practical avenue for enhancing security analytics in decentralized finance. We hope this work inspires further exploration into combining graph knowledge with machine learning to strengthen the resilience of DeFi platforms.

References

[1] Chainalysis Team. 2023. *2022 Biggest Year Ever For Crypto Hacking with $3.8 Billion Stolen, Primarily from DeFi Protocols and by North Korea-linked Attackers.* Chainalysis Blog (Feb. 1, 2023). Retrieved from https://www.chainalysis.com/blog/2022-biggest-year-ever-for-crypto-hacking/.

[2] S. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. Knottenbelt. 2021. SoK: Decentralized Finance (DeFi). *arXiv:2101.08778* (2021).

[3] G. Ke et al. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30.* 3146–3154.

[4] J. Michel and P. Parrend. 2023. Graph-Based Intelligent Cyber Threat Detection System. In *Cybersecurity in Intelligent Networking Systems.* CRC Press.

[5] D. Pavlova. 2025. DeFi Security Trends: Semantic Knowledge Graph Analysis (Code & Dataset). GitHub Repository. https://github.com/dariapavlova02/defi_trends_semantic.

[6] L. Sikos. 2023. Cybersecurity Knowledge Graphs. *Knowledge and Information Systems* 65, 3511–3531.

[7] L. Sadlek et al. 2025. Severity-Based Triage of Cybersecurity Incidents Using Kill Chain Attack Graphs. *Journal of Information Security and Applications* 89.

[8] A. Asiri and K. Somasundaram. 2025. Graph Convolution Network for Fraud Detection in Bitcoin Transactions. *Scientific Reports* 15, 11076.

[9] M. Li et al. 2025. Global-Local Graph Attention with Cyclic Pseudo-Labels for Bitcoin AML Detection. *Scientific Reports* 15, 22668.

[10] R. Canillas et al. 2020. GraphSIF: Analyzing Flow of Payments in a B2B Network to Detect Supplier Impersonation Fraud. *Applied Network Science* 5, 40.

[11] D. Cheng et al. 2020. Contagious Chain Risk Rating for Networked-Guarantee Loans. In *Proceedings of KDD '20.* ACM, 2715–2723.

[12] J. Wang et al. 2022. A Review on Graph Neural Network Methods in Financial Applications. *Journal of Data Science* 20(2): 111–134.