PNNL- 36847

Efficient Hybrid Attack Graph Generation for Cyber-Physical System Resilience Experimentation

Final Project Report

September 2024

Sumit Purohit
Rounak Mayur
Oceane M Bel
Armando Sanchez Mendoza
Braden K Webb
Sam Donald

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL- 36847

0BEfficient Hybrid Attack Graph Generation for Cyber-Physical System Resilience Experimentation

Final Project Report

September 2024

Sumit Purohit
Rounak Mayur
Oceane M Bel
Armando Sanchez Mendoza
Braden K Webb
Sam Donald

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

HAGEN project has developed theory, algorithms, and capabilities to assist cyber physical system modelers and operators to perform system and device-level vulnerability assessment, risk assessment, impact assessment, and mitigation planning. The project generates hybrid attack graphs for Cyber-Physical System (CPS) resilience experimentation at desired scale and speed. The project will produce composite attack datasets, algorithms, and demonstrable prototypical tools, and a library of high-impact attack sequences for a given CPS of interest. This report provided overall summary of research and development performed between FY22-24.

Summary

Efficient Hybrid Attack Graph Generation for Cyber-Physical System Resilience Experimentation (HAGEN) LDRD performed research to identify credible test cases at the edge for CPS resilience experiments at desired scale and speed. To achieve project objectives, HAGEN focused on following S&T research questions:

1. How to characterize hybrid CPS dynamics across scale with sparse data?
2. How can the hybrid dynamics be leveraged to generate credible course-of-action HAGs?
3. How to perform dimension reduction of HAGs for edge test case experimentation?

HAGEN project produced capabilities in the research areas shown in the Figure 1.

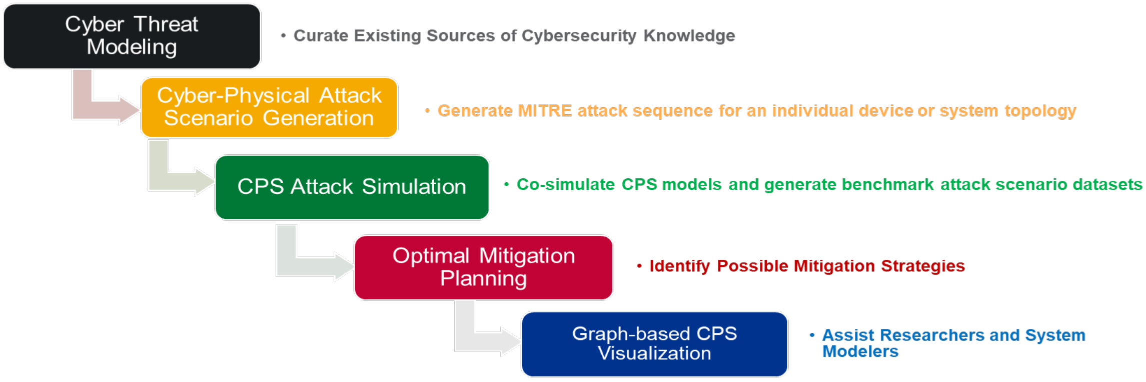

Cyber Threat Modeling
• Curate Existing Sources of Cybersecurity Knowledge
Cyber-Physical Attack Scenario Generation
• Generate MITRE attack sequence for an individual device or system topology
CPS Attack Simulation
• Co-simulate CPS models and generate benchmark attack scenario datasets
Optimal Mitigation Planning
• Identify Possible Mitigation Strategies
Graph-based CPS Visualization
• Assist Researchers and System Modelers

Figure 1 HAGEN Research and Development Areas

Acknowledgments

This research was supported by the Resilience through Data-Driven Intelligently Designed Control (RD2C), under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Acronyms and Abbreviations

CPS: Cyber Physical System

DER: Distributed Energy Resources

CVE: Common Vulnerabilities and Exposures

CWE: Common Weakness Enumeration

CAPEC: Common Attack Pattern Enumeration and Classification

TTP: Tactics, Techniques, and Procedures

ICS: Industrial Control System

Contents

Figures

Tables

1.0 Introduction

Cyber-Physical Systems (CPSs) feature a vast input space, encompassing both discrete and continuous elements across multiple layers. Critical infrastructures have become increasingly complex, interconnected, and susceptible to adverse conditions, including cyber and physical attacks as well as operational faults. The integration of distributed energy resources (DER), coupled with high-temporal-resolution control and hybrid ownership patterns, further complicates the identification of potential adversarial scenarios and their impacts on CPS. Over the years, cyber-attacks on these systems have grown in number, sophistication, and impact.

Experimentation with "what-if" scenarios, particularly those involving low-probability but high-impact cases, is essential for identifying resilient CPS operations. However, the explosion of CPS state-space demands substantial computational resources for the timely generation of critical decision-support inputs. Hybrid Attack Graphs (HAG) offer a flexible and efficient approach to generating attack sequences within a CPS. In HAGs, CPS states and dynamics are represented as nodes, while adversary tactics and physical actions are represented as edges, illustrating transitions from one state to another.

1.1 Background and Research Gap

CPS resilience assessment process has been a manual, collaborative, and iterative process by a team of experts with knowledge of CPS system design, cyber-security, and vulnerability assessment. This time-consuming and costly process leads to incomplete and erroneous assessment. Various model-based approaches [Xinming et al. 2005, Bruhadeshwar et al. 2019, Ghazo et al. 2019] can automate the generation of small-scale attack graphs but requires complete observability of the architecture, connectivity, component states, vulnerabilities, and attack types supported by the model. Li et. al [Li Ming et al.] use NVIDIA's CUDA-based GPU to generate small attack graphs (AG) in parallel. A2G2V [Ghazo et al. 2019] is a model-based approach to generate attack graphs for SCADA network. Ibrahim et. al [Ibrahim et al., 2019] extend the model-based approach to generate hybrid attack graphs applied only to the communication network. The model-based approaches are limited in required scalability and heterogeneity to produce large-scale attack graphs critical for resilience experimentation. Our proposed data-driven and physics-informed approach addresses gaps in state-of-the-art by producing credible test cases to

Figure 2 HAGEN For CPS Resilience

facilitate CPS experimentation and co-design efforts.

A prerequisite of generating such HAGs is to understand relationship between different cyber concepts such as vulnerability, weakness, adversary, techniques, software, mitigations, etc. There have been many rule-based and AI/ML-based approaches to align such concepts. Most of these capabilities focus on enterprise domains. Additionally, Large Language Models (LLMs) provides exciting opportunity to leverage summarization and explainability of LLMs to extend existing ground-truth alignment using few shots approach [Das S et al. 2021]. HAGEN project organized existing cyber knowledge siloed in different repositories and provided a common interface to connect cyber concepts. It also explored LLM-based alignment capabilities. HAGEN also addressed gaps in predicting adversary's technique given a set of preconditions. HAGEN built-upon existing reinforcement and Q-learning based approaches to develop a simple reward policy to generate an adversarial scenario encoded as a MITRE technique graph walk. Resilience assessment is crucial for maintaining high availability, security, and quality of service in power grids. However, most current grid research lacks hardware testbed capabilities. Consequently, simulation testbeds have emerged to model real-world power grid topologies and evaluate the impact of various disruptions. HAGEN developed a co-simulation capability to measure the impact of cyber physical attack scenarios. Mitigation Recommendation is a key research topic and HAGE present a novel framework capable of addressing the aspect of allocating budget to different organizational sectors serving a common goal of reducing vulnerability of adversarial cyber-attacks. Finally, HAGEN also developed novel capability to model and explore distribution power grid via a graph-based, cross platform application. All the research deliverables and capabilities are discussed in detail.

2.0 HAGEN Methodology

HAGEN project used graph-based approaches, coupled with CPS system modeling and optimization techniques to understand different aspects of cyber-physical system resilience assessment and risk mitigation. HAGEN focused on different research sub-problems and produced impactful deliverables as described in the subsequent sections.

2.1 Cyber Threat Modeling

The MITRE Corporation is a non-profit organization that works to address issues in diverse fields such as cybersecurity, national defense, and healthcare. Among MITRE's key contributions to cybersecurity research are several comprehensive databases that catalog and organize information related to cyber threats. These resources include:

- **Common Vulnerabilities and Exposures (CVE)**: Publicly disclosed cybersecurity flaws.
- **Common Weakness Enumeration (CWE)**: Common types of software and hardware weaknesses.
- **Common Attack Pattern Enumeration and Classification (CAPEC)**: A classification of known cyber-attack patterns.
- **Techniques**: Specific methodologies used by adversaries, categorized by operational environments.
- **Mitigations**: Strategic actions designed to reduce or eliminate the impact of specific techniques.
- **Software and Groups**: Collections of techniques implemented by specific tools, software, or groups.

These resources are related to one another in the following manner:

Common Vulnerabilities and Exposures (CVE) is a standardized cataloging system for publicly known information security vulnerabilities and exposures. Each entry includes a brief description of the vulnerability, metrics related to their potential impact severity, and references associated CWEs.

Common Weakness Enumeration (CWE) is a community-driven list of common weakness types that affect software and hardware. It serves as a tool for identifying, addressing, and mitigating security flaws in the design and architecture of technology products. CWEs can be related to one another through child of, preceding, and member of relationships, along with being associated with CAPECs.

Common Attack Pattern Enumeration and Classification (CAPEC) is resource that provides details of known attack patterns. It categorizes approaches, such as CAPEC-112 describing “Brute Force” attacks, that adversaries may use to exploit specific vulnerabilities and weaknesses.

MITRE categorizes **Techniques** used by adversaries into different operational environments, including enterprise systems, mobile devices, and Industrial Control Systems (ICS). Each matrix is structured by tactics, detailing the high-level objectives such as gaining initial access to a system, along with the specific techniques used to achieve the objective tactic. This organization aids in understanding the progression of a cyber-attack. **Mitigations** are strategic actions to reduce or eliminate the impact of specific techniques.

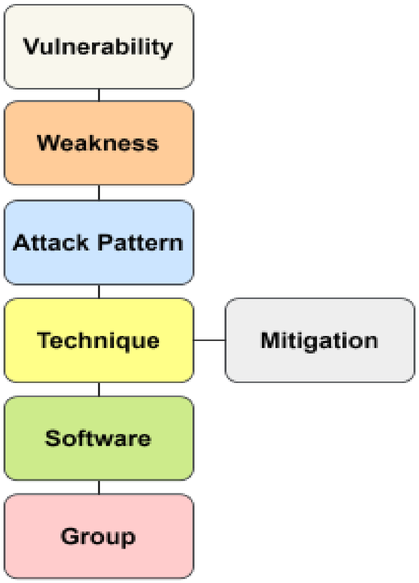


Figure 3 MITRE KG node layers.

MITRE provides documentation regarding **Software** instances and the associated techniques applied by them, along with threat actor **Groups** associated with specific tactics, techniques, and procedures (TTPs) or software instances.

The MITRE Knowledge Graph (MITRE KG) is a tool designed to facilitate the generation of threat profiles using data gathered from the above MITRE sources. It is based on a Neo4j graph database with multiple endpoints, allowing for relational-based querying, enabling users to easily explore the data structure and extract specific information needed. Currently MITRE KG is provided in the form of multiple Docker containers, with the intent of future AWS deployment. MITRE KG is built by aggregating MITRE data and forming relationships based on the provided connections between data types. The resultant structure is visualized below.

To efficiently extract information from the graph, several endpoints are provided:

- **`/find_node`** allows users to search for nodes based on their type and optionally their ID, returning structured JSON information about the node of interest.

- **`/find_subgraph`** enables the creation of subgraphs by specifying a target node and defining a range of connected node types and is visualized below. This functionality helps users find relationships between different elements within the cybersecurity ecosystem, such as linking a specific threat actor group to mitigations that would be effective at countering their applied techniques. The **`/find_related_nodes`** endpoint is similar in function but returns the set of nodes within the associated subgraph that are of a specific type as opposed to the entire subgraph.

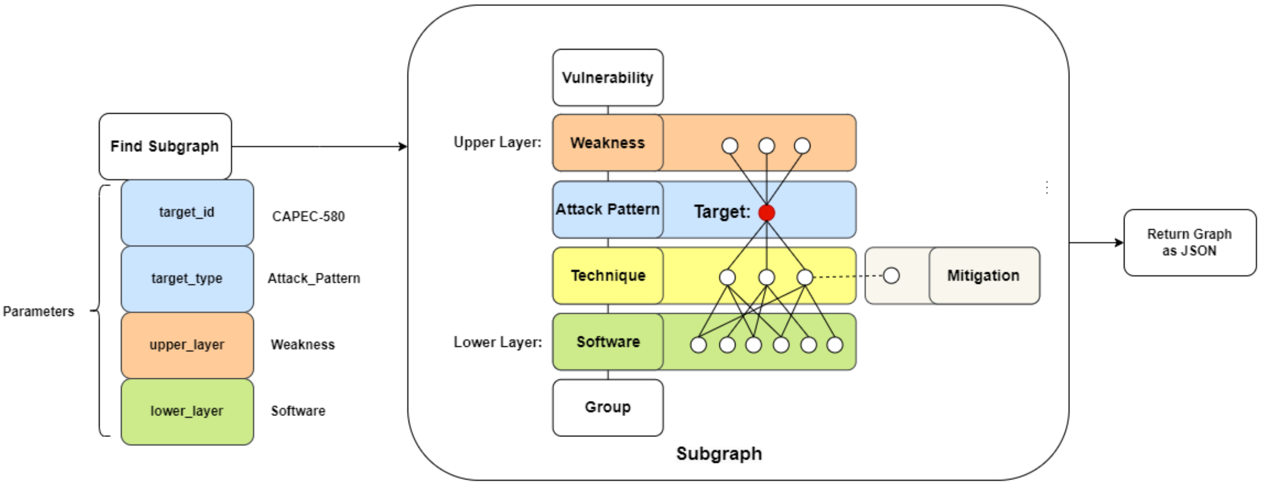


Figure 4: Find Subgraph endpoint visualization

- The **`/search`** endpoint of the MITRE Knowledge Graph is notable for its ability to both find CVEs based on keyword searches of their descriptions and add these CVEs to the knowledge graph. Given the vast number of CVEs (240k as of September 2024) and the frequency of updates, this endpoint is essential for maintaining an accurate and manageable representation of CVE relationships. A cached option of the **`/search`** endpoint is also available and is designed to be used with a precompiled subset of approximately 25k CVEs, which can be loaded into the knowledge graph via the **`/load`** endpoint.

2.2 Cyber Knowledge Completion

To fully assess the risks to which Cyber-Physical Systems (CPSs) are exposed and to better model the behavior of potential adversaries, we wanted to ensure that all available information from both the Common Attack Pattern Enumerations and Classifications (CAPEC) and the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework is fully incorporated into our tools. While CAPEC serves as a publicly available catalog of cyber-attack patterns, ATT&CK provides a catalog of adversarial motivations or goals (*tactics)* and a list of actions to achieve the goals (*techniques*) in enterprise, mobile, and ICS networks. To this end, we expand upon the Cyber Threat Modeling work by creating a method of mapping to-and-from CAPEC attack patterns and ATT&CK ICS techniques.

This integration is a large task—as of August 2024, there were 559 CAPEC attack patterns and 83 ATT&CK ICS techniques, for a combination of over 46k possible mappings between the two information repositories. Since these records are updated frequently and significant domain expertise is often necessary to verify whether a given connection is valid, we developed an automated method for generating these mappings.

Although traditional machine learning classifiers might seem useful for identifying similar descriptions of attack patterns and techniques, most of these approaches require structured input to learn representations. Almost all the information describing a CAPEC attack pattern or ATT&CK ICS technique, however, is expressed in fields of unstructured text consisting of entry identifier, name and description.

We therefore turn to methodologies of natural language processing, where recent advances in large language models provide an opportunity to use artificial intelligence as a tool in automating the mapping task. In particular, many embedding models can algorithmically encode text strings as arrays of floating-point numbers in a high-dimensional normed vector space. We can interpret these vectors, known as *document embeddings* or simply *embeddings*, as representing the semantics of the input text. Embeddings of documents with similar meanings end up close together, and unrelated documents generate embeddings that are farther apart. We use this process to treat difficult-to-handle unstructured text as mathematical vectors, to which we can then apply more standard machine learning tools. Specifically, we compare mapping approaches that identify nearest neighbors in embedding space, and a retrieval-augmented generation (RAG) approach. We evaluate our results on a hand-labeled data set.

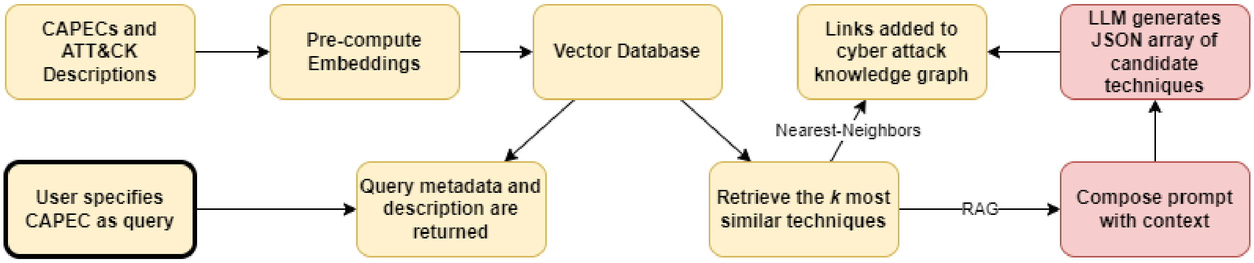

CAPECs and ATT&CK Descriptions
Pre-compute Embeddings
Vector Database
Links added to cyber attack knowledge graph
LLM generates JSON array of candidate techniques
Nearest-Neighbors
User specifies CAPEC as query
Query metadata and description are returned
Retrieve the *k* most similar techniques
RAG
Compose prompt with context

Figure 5: Cyber Knowledge Completion using LLMs

Our approach to learn the mapping function between the CAPEC and ATT&CK taxonomies utilizes document embedding models. Specifically, we use transformer-based neural networks that produce fixed-length, dense representations of variable length documents—allowing us to compare the taxonomies quantitatively. We prepare a *description string* for each CAPEC attack pattern and ATT&CK ICS technique by concatenating their name, ID, and description, as displayed in Fig. 1. An embedding model then tokenizes each input description string to a list of tokens and transforms those tokens to a vector, which is stored in a vector database.

For an embedding of any given CAPEC attack pattern, we can then retrieve the *k* closest embeddings of ATT&CK ICS techniques. Associating these to the CAPEC attack pattern provides us with a first mapping of CAPECs to ICS techniques, parameterized by *k*. We can likewise generate this nearest-neighbor mappings in the reverse direction, associating *k* CAPEC embeddings to any ICS technique.

The nearest-neighbor embedding approach provides a baseline method of retrieving potential candidates for CAPEC-ATT&CK mappings but suffers from filtering those candidates with precision. It also requires every CAPEC attack pattern (or ATT&CK ICS technique) to be linked to the same fixed number *k* of ATT&CK ICS techniques (or CAPEC attack patterns), when the number of links might vary widely. We therefore implemented a RAG-based mapping to address these problems.

The RAG-based approach begins with a nearest-neighbor mapping and then systematically wraps each retrieved technique in a set of prompts and passes them to a large language model (in our latest version, we utilized Meta's 8-billion parameter, instruction fine-tuned variant of Llama 3). Since we want the language model's output to be structured in a predictable, machine-accessible format, we leverage a decoding technique to sample generated tokens according to any context-free grammar of our choosing. This allows us to specify a JSON schema, which we can then convert to a Backus-Naur form of a formal grammar to constrain the LLM's output. As a result, the language model filters down the set of retrieved techniques to only those which it deems to be relevant to the queried attack pattern, providing a justification for each of its "decisions".

We compare four embedding models for each mapping: *E5-large-v2, instructor-large, all-MiniLM-L6-v2*, and *text-embedding-ada-002*. For each model, we evaluated the pipeline's performance using standard metrics like Recall, Precision, and F-Score, as well as some custom task-specific metrics. We found that RAG-based mappings generally outperform nearest neighbor mappings in terms of precision and F-score across most embedding models, and that *instructor-large* and *text-embedding-ada-002* exhibit the best performance.

2.3 CPS Attack Scenario Generation

In cyber-physical energy systems (CPES), defenders face challenges due to the complexity and evolving nature of cyber threats. Cybersecurity professionals typically rely heavily on databases of prior documented attacks, like those described by the MITRE ATT&CK framework. However, the limited number of documented attack sequences restricts defenders' abilities to anticipate and mitigate new and increasingly sophisticated attacks.

Hybrid Attack Graphs (HAGs) are a representation that can be used to model these possible attack sequences within CPES. The nodes within a HAG represent specific MITRE ATT&CK techniques, while edges denote their order within an attack sequence, effectively mapping potential pathways an adversary might follow. A single attack is represented as a linear sequence of techniques, while a collection of attacks forms a graph describing potential attack paths. HAGs provide a structured way to understand the interplay between different attack methods and the overall strategy an attacker might use to compromise a system.

Given the limited set of existing HAGs (less than 30 fully defined HAGs), there is a need to generate realistic synthetic HAGs to feed data-hungry downstream models and provide cyber analysts with robust inference tools. By training a deep-learning generator model on the limited data available, it is possible to produce synthetic HAGs that expand the effective size of the dataset, making it more diverse and comprehensive. Furthermore, through reinforcement learning, we can direct generation towards HAGs with specific user-defined attributes, such as difficulty in detection or energy expenditure.

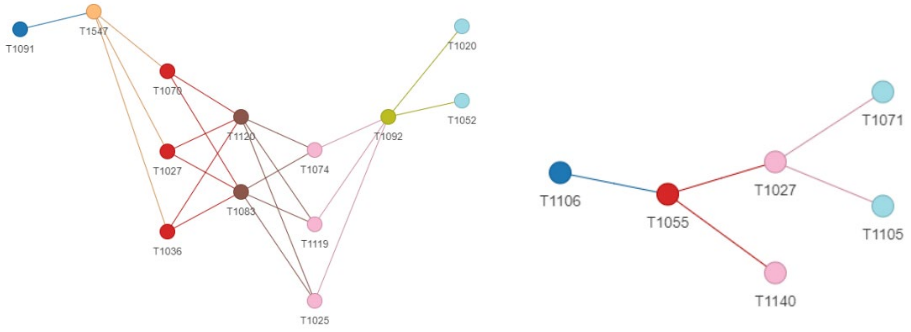


Figure 6: Dense (left) and sparse (right) training HAGs

The Graph Convolutional Deep-Q Learning (GCDQ) model accomplishes this goal of generating HAGs with targeted properties. The model was trained on 620 documented software instances from the MITRE KG database, along with the associated techniques they incorporate. An ordering scheme collates techniques into their corresponding tactics and adds edges between techniques in adjacent tactics, creating dense HAGs approximating the software's operation. These dense graphs are then sampled into 4096 sparse HAGs using a random walk algorithm with backtracking. A sample dense (left) and sparse (right) training HAG is displayed above.

Training is done under a GAN framework, with the Graph Convolutional Generator synthesizing HAGs and a Graph Convolutional Discriminator simultaneously learning to distinguish them from the training dataset. The generator model is incrementally trained to favor outputs producing a high reward through deep-Q learning, with the reward comprising a user-defined trait along with its ability to fool the discriminator. This process results in realistic HAGs that align with the user-defined traits. Attack detectability was used as an example during training, calculated by the product of each individual HAG nodes likelihood of being detected. Following training, the GCDQ model was able to produce HAGs with a high normalized detection reward component (0.81) and a modest discriminator reward (0.44), while maintaining high similarities in graph structure as determined by the Frobenius norm and Laplacian spectral distance. The convergence plot for reward components during GCDQ training is provided below.

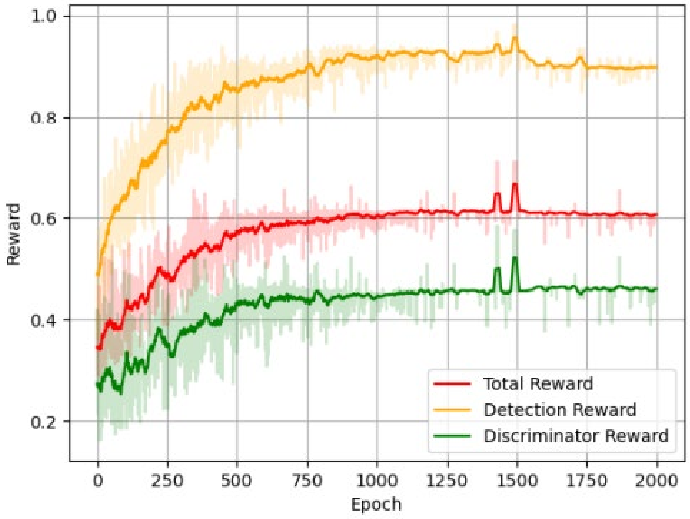


Figure 7: Reward convergence during GCDQ training

Importantly, the GCDQ model can be applied to different areas of cybersecurity strategies, beyond simply generating larger datasets. Utilizing the trained GCDQ model, we can predict the distribution of likely next techniques based on an existing partially complete HAG or create a HAG with specific techniques based on prior knowledge of a systems vulnerability. This versatility provides cyber analysts with a broad tool to enhance their predictive capabilities and strengthen overall system defenses.

2.4 CPS Attack Simulation:

NATIG
NETWORK ATTACK TESTBED IN [POWER] GRID

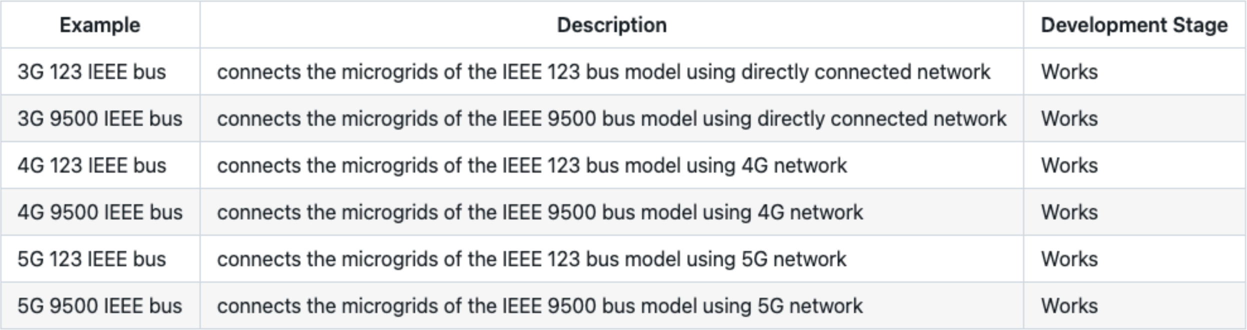
The integration of GridLAB-D and NS3 allows users to simulate and analyze the interaction between physical grid components and communication networks, providing a comprehensive view of grid operations. This multi-layer architecture permits the detailed study of control strategies and their impact on grid stability and performance under various scenarios, including cyberattacks. Additionally, users can experiment with different network topologies, communication protocols, and security measures, making the platform a versatile tool for both academic research and practical utility management.

Network Attack Testbed in Power Grid (NATIG) evaluates cyber-physical attack impacts by simulating attacks. It enables realistic simulations and helps maintain reliable operations by integrating real and simulated environments. By enabling NATIG to work with both the IEEE 123 feeder model and the IEEE 9500 bus model with minimal to no changes needed by the user in the code, NATIG allows for a more extensive and scalable NS3 setup, showcasing the framework's flexibility to handle larger, more complex networks compared to the IEEE 123 bus model. Users benefit from the ability to tailor the network division to their specific needs, enabling detailed and customizable simulations. This flexibility enhances the platform’s applicability for diverse research and operational scenarios, providing a comprehensive tool for grid management and analysis.

NATIG can simulate complex cybersecurity attacks, such as Man-in-the-Middle (MIM) and Distributed Denial-of-Service (DDoS) attacks, within a DNP3 application on NS3 with needing much setup from the user. Users can simulate sophisticated cyberattacks on grid networks without needing extensive cybersecurity or network programming knowledge, thus lowering the barrier to entry. NATIG enables users to evaluate the resilience and security measures of their network under various attack scenarios, providing valuable insights into potential vulnerabilities.

Here is a list of the available examples in NATIG. Overall, we have 2 IEEE models and 2 different attack types now functional and useful as part of NATIG. The 3G examples represent any communication network type that is not cellular. We called it 3G for consistency in the code. 3G topologies, that have been tested, include ring, mesh and star. The connection types used in 3G include point to point, csma and wifi. We also added for 3G the ability for the user to setup their own topology using topology.json configuration file.

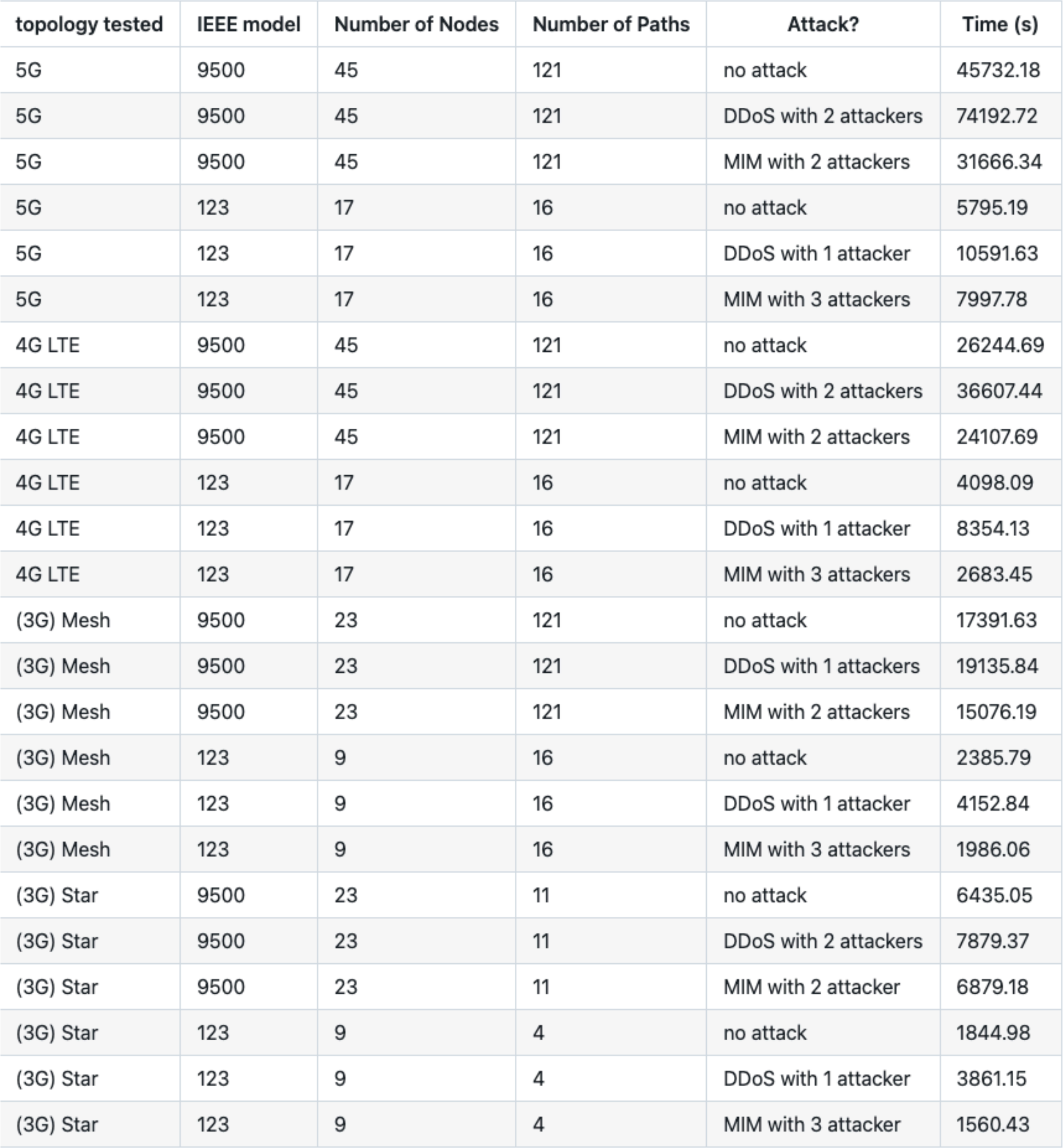
Table 8: NATIG status of Out-of-box scenario execution



Example	Description	Development Stage
3G 123 IEEE bus	connects the microgrids of the IEEE 123 bus model using directly connected network	Works
3G 9500 IEEE bus	connects the microgrids of the IEEE 9500 bus model using directly connected network	Works
4G 123 IEEE bus	connects the microgrids of the IEEE 123 bus model using 4G network	Works
4G 9500 IEEE bus	connects the microgrids of the IEEE 9500 bus model using 4G network	Works
5G 123 IEEE bus	connects the microgrids of the IEEE 123 bus model using 5G network	Works
5G 9500 IEEE bus	connects the microgrids of the IEEE 9500 bus model using 5G network	Works

Here is a list of the runtime of each example. This table showcases the impact of the size of the topology and the type of communication topology on the performance of the simulation in a docker container.

Table 9: NATIG runtime on SLURM-based system



topology tested	IEEE model	Number of Nodes	Number of Paths	Attack?	Time (s)
5G	9500	45	121	no attack	45732.18
5G	9500	45	121	DDoS with 2 attackers	74192.72
5G	9500	45	121	MIM with 2 attackers	31666.34
5G	123	17	16	no attack	5795.19
5G	123	17	16	DDoS with 1 attacker	10591.63
5G	123	17	16	MIM with 3 attackers	7997.78
4G LTE	9500	45	121	no attack	26244.69
4G LTE	9500	45	121	DDoS with 2 attackers	36607.44
4G LTE	9500	45	121	MIM with 2 attackers	24107.69
4G LTE	123	17	16	no attack	4098.09
4G LTE	123	17	16	DDoS with 1 attacker	8354.13
4G LTE	123	17	16	MIM with 3 attackers	2683.45
(3G) Mesh	9500	23	121	no attack	17391.63
(3G) Mesh	9500	23	121	DDoS with 1 attackers	19135.84
(3G) Mesh	9500	23	121	MIM with 2 attackers	15076.19
(3G) Mesh	123	9	16	no attack	2385.79
(3G) Mesh	123	9	16	DDoS with 1 attacker	4152.84
(3G) Mesh	123	9	16	MIM with 3 attackers	1986.06
(3G) Star	9500	23	11	no attack	6435.05
(3G) Star	9500	23	11	DDoS with 2 attackers	7879.37
(3G) Star	9500	23	11	MIM with 2 attacker	6879.18
(3G) Star	123	9	4	no attack	1844.98
(3G) Star	123	9	4	DDoS with 1 attacker	3861.15
(3G) Star	123	9	4	MIM with 3 attacker	1560.43

Finally, we also have a version of NATIG that can be run on a slurm system. That version has a drastic impact on the performance of the simulation. It manages to cut down the run time overhead of worst performing simulation by 90%.

NATIG capability is also used for a red-teaming scenario. We worked with SCOREDEC project to define a set of objective function and adversarial scenarios to identify optimal design configuration.

2.5 Optimal Mitigation Planning

A system planner needs to identify optimal set of mitigation measures to thwart probable adversary actions and at the same time improve the existing mitigation efficacy given an available planning budget. Our optimal mitigation framework (depicted in Figure 2) achieves two main objectives: (i) it minimizes adversarial risk by creating hybrid attack graphs to identify and mitigate critical threats, and (ii) it prioritizes organizational goals by categorizing mitigation measures into budget sectors and allocating funds in alignment with the organization's priorities.

We harness the MITRE ATT&CK framework to plan mitigation measures for a component. To this end, we need to define a component's vulnerability in the context of the MITRE ATT&CK framework and the HAG generated for the component. The generated HAG provides us with possible sequences of techniques that adversaries could utilize to perform a successful cyber-attack. A planner might seek to choose an optimal set of mitigation measures to minimize the probability of compromising the component through any of the attack sequences identified through the HAG. This goal objective requires us to minimize the success probability of each and every attack sequence in the HAG. However, for all practical purposes, reducing these success probabilities to a sufficiently small value is acceptable. We term this objective as minimizing the number of "highly likely" attack sequences.

Following (Georgiadou, Mouzakitis, & Askounis, 2021), we categorize the mitigation measures into the following overlapping sectors (or categories): asset management, business continuity, access and trust, operations, defense, security governance and employee training. The underlying assumption is that allocating budget improves mitigation measure efficacy, i.e., the probability that a mitigation measure successfully prevents a technique. We compute analytic expression for the mitigation efficacy and its dependence on the budget partitions and present an optimization framework which helps us identify the optimal set of mitigation measures along with the optimal allocation of budget in the various sectors.

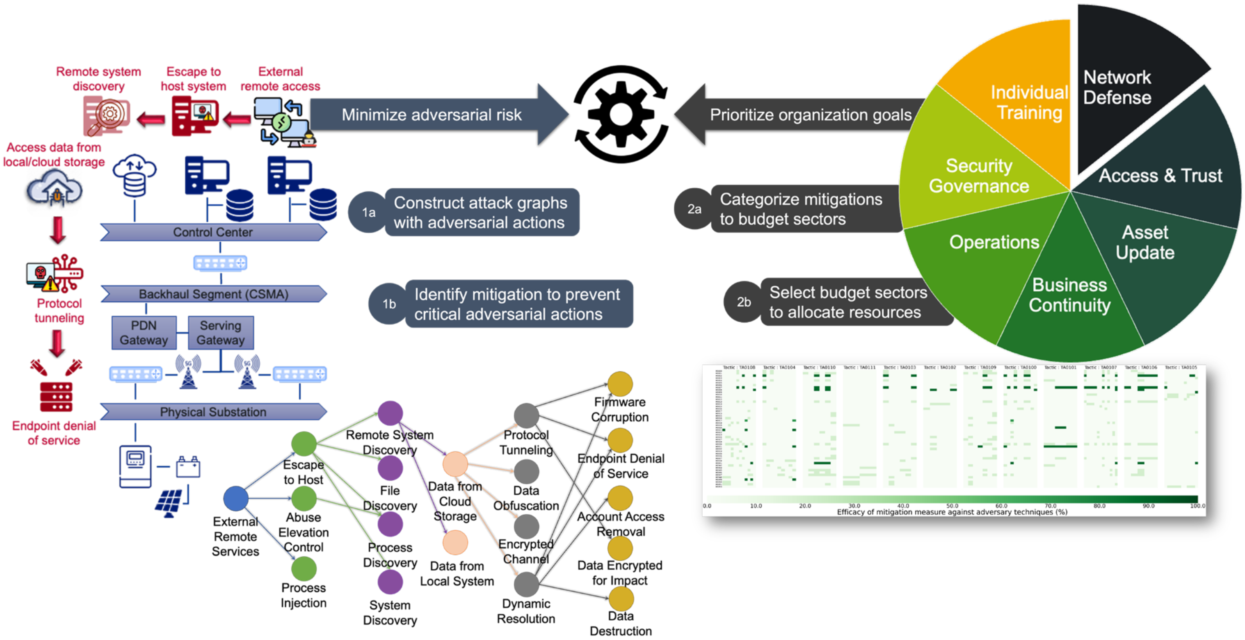


Figure 10 Overview of the optimal mitigation framework to identify mitigation measures and allocate budget to minimize cyber vulnerability

We compare three approaches to solve the proposed mixed integer optimization problem: (a) an off-the-shelf SCIP solver, (b) a strong branching agent-based solver (Khalil, Le Bodic, Song, Nemhauser, & Dilkina, 2016), and (c) a GNN-based solver (Prouvost, et al., 2020). In all instances, the off-the-shelf solver consistently outperforms the alternative methods, exhibiting the shortest solution times and exploring the fewest nodes. This superior performance can be attributed to the relatively small size of the problem instances, where traditional solvers are well-optimized to handle such cases efficiently. While existing literature acknowledges the potential advantages of strong branching agent-based and graph neural network-based solvers for larger, more complex problems, the scale of our problem remains within a range where the off-the-shelf solver is particularly effective. This observation underscores the importance of considering problem size when selecting an appropriate solver.

2.6 Graph-based CPS Visualization

GLIMPSE
GRID LAYOUT INTERFACE FOR MODEL
PREVIEW AND SYSTEM EXPLORATION

GridLAB-D is a simulation tool that enables researchers in the power systems domain to simulate and analyze power distribution systems. Power distribution systems are networks that down-convert high-voltage power from the transmission system and provide it to industrial, commercial, and residential users. GridLAB-D uses *glm* (GridLAB-d Model) files that are used to synthesize populations of objects and encode power distribution object behavior. These *glm* files can be large and complex making them hard to understand, update, and maintain.

Visualizing *glm* files can be difficult as well and often require the use of a programming languages like python and some specialized libraries. There is also a lack of light-weight visualization tools that offer a dynamic and intuitive user experience. Most visualization tools display static representations of these distribution models with little to no functionality to update and maintain such models.

We present GLIMPSE (**G**rid **L**ayout **I**nterface for **M**odel **P**review and **S**ystem **E**xploration) tool designed to provide an intuitive UI experience and offer a dynamic visualization of distribution systems as an attributed LPG (Label Property Graph). GLIMPSE provides users many models update, model exploration, and model analysis features that can aid in the understanding of complex *glm* files. The tool is developed using React.js, Node.js, Vis.js Electron.js, and Python.

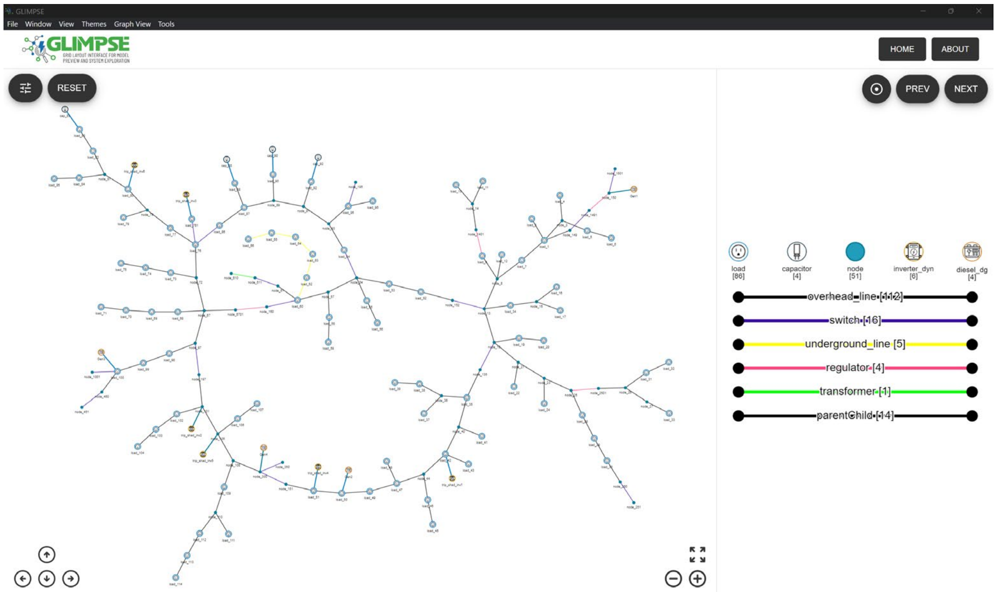
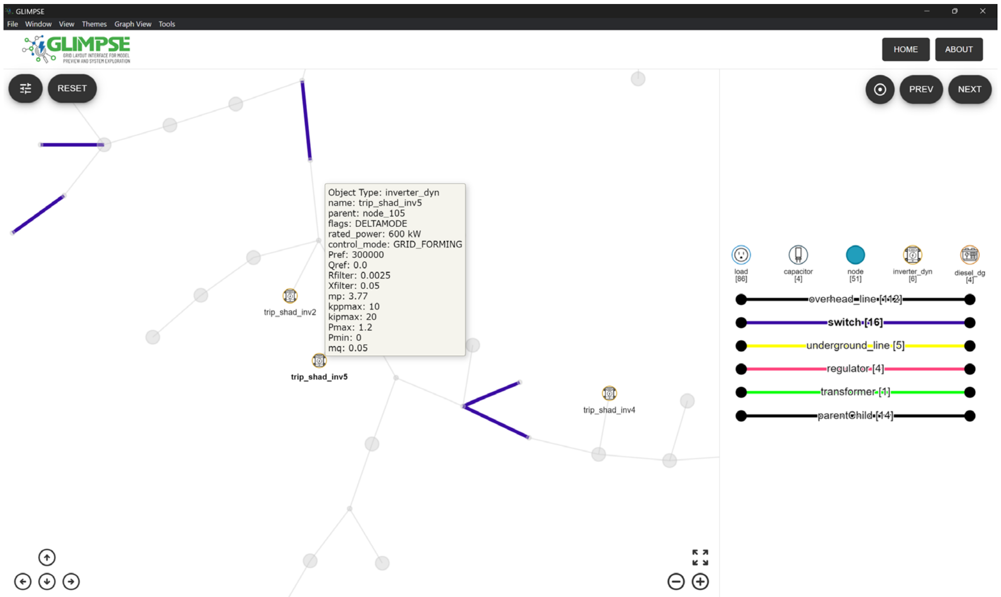


Figure 11: GLIMPSE Interface

The GLIMPSE User Interface offers the *NEXT and PREV* (previous) buttons along with Legend's highlighting features for model exploration. Double Clicking on any node or edge in the legend will highlight those object types in the network visualization on the left. Double clicking on more than one object type, either edge or node, will be highlighted, and if any highlighted type is double clicked again that type will not be highlighted anymore. If there are nodes that are highlighted users may click on the next and previous buttons to zoom in on each highlighted node and cycle between highlighted node types. Clicking the *RESET* button will revert all highlighting back to the visualization's original styles.

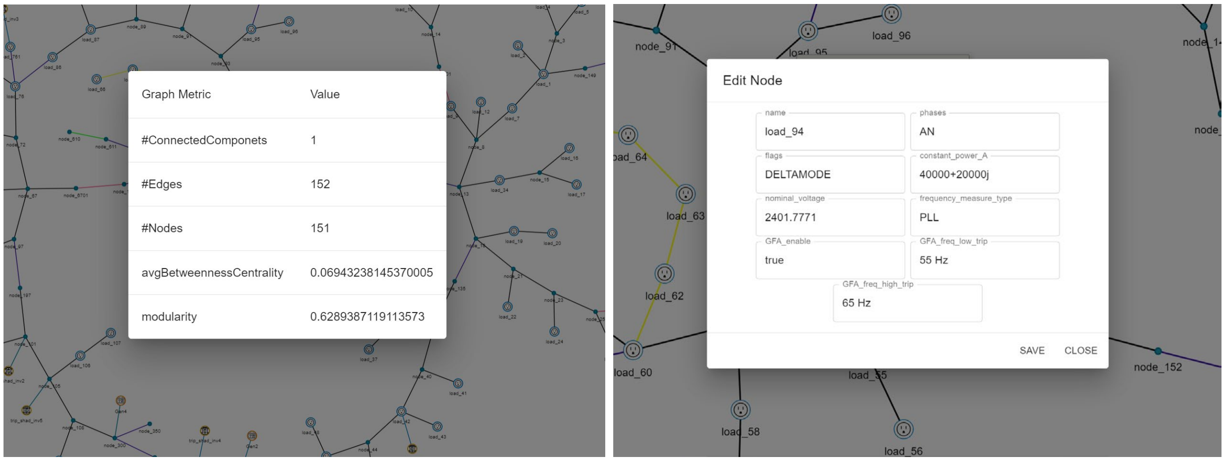


Object Type: inverter_dyn
name: trip_shad_inv5
parent: node_105
flags: DELTAMODE
rated_power: 600 kW
control_mode: GRID_FORMING
Pref: 300000
Qref: 0.0
Rfilter: 0.0025
Xfilter: 0.05
mp: 3.77
kppmax: 10
kipmax: 20
Pmax: 1.2
Pmin: 0
mq: 0.05

Figure 12: Node/Edge Highlight

When gaining further insight other than a clear view of the distribution system's topology users may select the menu item *Graph Metrics* to get some network summary statistics. These figures are quired by creating and maintaining a Networkx graph object in GLIMPSE's accompanied local

WebSocket server. This server is packaged and initialized when the application is started allowing for users with programming experience to send data to GLIMPSE's WebSocket server API for real time visual updates to an active visualization. Details for the WebSocket server API can be found in the tool's public GitHub recent releases.



Graph Metric	Value
#ConnectedComponets	1
#Edges	152
#Nodes	151
avgBetweennessCentrality	0.06943238145370005
modularity	0.6289387119113573

Edit Node

name load_94	phases AN
flags DELTAMODE	constant_power_A 40000+20000j
nominal_voltage 2401.7771	frequency_measure_type PLL
GFA_enable true	GFA_freq_low_trip 55 Hz
GFA_freq_high_trip 65 Hz	

Figure 13: GLIMPSE-based system update

Updating a distribution model's objects can be done by double clicking on a node object and a form will show allowing the user to edit that object's many attributes. These changes are saved to the GLIMPSE object datatype in the background. When a user is satisfied with their changes, they can export a copy of their upload *glm* files with the appropriate changes reflected in the new files. These new files can then be re-uploaded to GLIMPSE for further visualization.

Since GLIMPSE is developed using Electron.js there is installer of the tool made for Windows allowing users to easily install and begin visualizing. In the case of Mac and Linux OS users would need to build the tool to use which can be done by following the tools clear instructions in the public GitHub repository. Currently the tool offers many more features including:

- Right clicking on an empty space in the visualization will prompt the user to add a new node or save the current visualization as an image to their computer.
- Right clicking on a node will allow you to delete that node while right clicking on an edge will allow you to hide that edge or hide all edges of that type.
- Right clicking on the node or edge types in the legend will show a context menu and by selecting the edit theme button a movable form will show. This form contains options to change the nodes color, shape, size, and image if selecting the *circularImage* shape. Users are also able to change an edge type's color and width.
- Any style changes done can be saved and exported to a custom.theme.json file by selecting export theme in the themes file menu at the top-left of the tool. This file can then be re-uploaded with any GLIMPSE compatible file that those themes apply to.
- The tool will automatically visualize larger models with 2800 nodes or more in clusters with the use of community detection.
- Clustered nodes can be opened and re-clustered when right clicking any of the nodes that belong to an open cluster.
- Edges can also be animated to show the flow of a connection between nodes.
- Support for Networkx node-link data JSON dump file, CIM (Common Information Model) XML files, and GLIMPSE JSON data structure based on the *glm2json* parser's JSON output.

GLIMPSE offers a dynamic visualization experience that is often not found in other lightweight tools used to visualize power distribution models. Although this tool was developed for the use in the power systems domain, we have implemented many features that allow researchers of different domains to visualize many different networks and systems. GLIMPSE's WebSocket API can encourage researchers to visualize in real time their simulations whether that be a cyber-attack or the flow of traffic in an internet network all in GLIMPSE.

3.0 Research Impact

3.1 Publications

HAGEN project has consistently published peer-reviewed papers in top conferences.

Co-Simulation Framework For Network Attack Generation and Monitoring PNNL-SA-190158 https://informationrelease.pnnl.gov/release/343152 Bel O., J. Kim, W.J. Hofer, M. Maharjan, B. Hyder, S. Purohit, and S. Niddodi. 2023. "Co-Simulation Framework for Network Attack Generation and Monitoring." IEEE Access. PNNL-SA-190158. Doi:10.1109/ACCESS.2024.3468272

GLIMPSE of Future Power Grid Models PNNL-SA-191994 https://informationrelease.pnnl.gov/release/345873 Mendoza Sanchez A., and S. Purohit. 2024. "GLIMPSE of Future Power Grid Models." In IEEE 18th International Conference on Semantic Computing (ICSC 2024), February 5-7, 2024, Laguna Hills, CA, 224-225. Piscataway, New Jersey: IEEE. PNNL-SA-191994. doi:10.1109/ICSC59802.2024.00042 03/22/2024

Enhancing Data Modeling, Exploration, and Monitoring through Knowledge Graph Visualization PNNL-SA-199164 https://informationrelease.pnnl.gov/release/355355 Purohit S., and A. Mendoza Sanchez. 2024. "Enhancing Data Modeling, Exploration, and Monitoring through Knowledge Graph Visualization." In 33rd ACM International Conference on Information and Knowledge Management (CIKM). PNNL-SA-199164.

Cyber Knowledge Completion using Large Language Models PNNL-SA-203400 https://informationrelease.pnnl.gov/release/360785 Webb B.K., R. Meyur, and S. Purohit. 2024. "Cyber Knowledge Completion using Large Language Models." In 2024 IEEE International Conference on Big Data. PNNL-SA-203400.

Impact-Driven Sampling Strategies for Hybrid Attack Graphs PNNL-SA-178629 https://informationrelease.pnnl.gov/release/328195 Subasi O., S. Purohit, A. Bhattacharya, and S. Chatterjee. 2023. "Impact-Driven Sampling Strategies for Hybrid Attack Graphs." In IEEE International Symposium on Technologies for Homeland Security (HST 2022) November 14-15, 2022, Virtual, Online, 1-7. Piscataway, New Jersey: IEEE. PNNL-SA-178629. doi:10.1109/HST56032.2022.10025439 01/30/2023

Hybrid Attack Graph Generation with Graph Convolutional Deep-Q Learning PNNL-SA-191137 https://informationrelease.pnnl.gov/release/344801 Donald S., R. Meyur, and S. Purohit. 2023. "Hybrid Attack Graph Generation with Graph Convolutional Deep-Q Learning." In IEEE International Con ference on Big Data (BigData 2023), December 15-18, 2023, Sorrento, Italy, 3127-3133. Piscataway, New Jersey: IEEE. PNNL-SA-191137. doi:10.1109/BigData59044.2023.10386675 12/31/2023

Fortify Your Defenses: Strategic Budget Allocation to Enhance Power Grid Cybersecurity PNNL-SA-192817 https://informationrelease.pnnl.gov/release/347009 Meyur R., S. Purohit, and B.K. Webb. 2023. "Fortify Your Defenses: Strategic Budget Allocation to Enhance Power Grid Cybersecurity." In Annual AAAI Conference on Artificial Intelligence. PNNL-SA-192817.

Hybrid Attack Graph Generation with Graph Convolutional Deep-Q Learning PNNL-SA-185588 https://informationrelease.pnnl.gov/release/337420 Purohit S., S. Donald, and R.

Meyur. 2023. "Hybrid Attack Graph Generation with Graph Convolutional Deep-Q Learning." In The 3rd Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics. PNNL-SA-185588.

Cyber attack sequences generation for electric power grid. Dutta, A., Purohit, S., Bhattacharya, A., & Bel, O. (2022, May). In 2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES) (pp. 1-6). IEEE.

3.2 Open-Source Software Release

https://github.com/pnnl/GLIMPSE A graph-based desktop application to visualize and update GridLAB-D power grid models.

https://github.com/pnnl/NATIG Network Attack Testbed In [Power] Grid (NATI[P]G), a co-simulation environment for distribution power grid network using state-of-the-art simulators.

3.3 Professional Development and Contributions

The HAGEN project frequently engaged in DOE-supported internship and workforce development programs such as SULI (Science Undergraduate Laboratory Internships), CCI (Community College Internships), and OMNI (Office of Minority and National Inclusion). The HAGEN team mentored summer students and provided them with hands-on research and development experience, working on challenging problems. Additionally, many early career researchers contributed to project deliverables over the course of the project's funding. The following current and past staff and interns also contributed to HAGEN: Arnab Bhattacharya, Omer Subasi, Ashutosh Dutta, Cimone L. Wright-Hamor, David Gaviria, Mohamed Abdelkader, and Lucy M. Tyrteos.

3.4 IP Generation and External Collaboration

HAGEN project generated multiple IP [32581-E, 32723-E, 33034-E, 33039-E, 33041-E, 33079-E] and exploring future technology transfer opportunities. HAGEN team has collaborated with Georgia Tech, Boise State University, North Carolina Agricultural and Technical State University, and Michigan State University.

4.0 Conclusion

Efficient Hybrid Attack Graph Generation for Cyber-Physical System Resilience Experimentation (HAGEN) project (FY22-24) performed research and development (R&D) to get insight into cyber-physical system modeling and its resilience assessment. It developed new algorithms and capabilities to assist system designer, modeling, and cybersecurity practitioners to perform vulnerability and risk assessment. It also provides mitigation recommendations constrained to given resource allocations. The project developed multiple capabilities and open-source software tools to further the research.

5.0 References

Georgiadou, A., Mouzakitis, S., & Askounis, D. (2021). Assessing MITRE ATT&CK risk using a cyber-security culture framework. *Sensors, 21*(9), 3267.

Khalil, E., Le Bodic, P., Song, L., Nemhauser, G., & Dilkina, B. (2016). Learning to Branch in Mixed Integer Programming. *Proceedings of the AAAI Conference on Artificial Intelligence.* ACM.

Prouvost, A., Dumouchelle, J., Scavuzzo, L., Gasse, M., Chetelat, D., & Lodi, A. (2020). Ecole: A Gym-like Library for Machine Learning in Combinatorial Optimization Solvers. *Learning Meets Combinatorial Algorithms at NeurIPS2020.* ACM.

Das, S. S., Dutta, A., Purohit, S., Serra, E., Halappanavar, M., & Pothen, A. (2022, November). Towards automatic mapping of vulnerabilities to attack patterns using large language models. In *2022 IEEE International Symposium on Technologies for Homeland Security (HST)* (pp. 1-7). IEEE.

Al Ghazo, Alaa T., et al. "A2G2V: automatic attack graph generation and visualization and its applications to computer and SCADA networks." IEEE Transactions on Systems, Man, and Cybernetics: Systems 50.10 (2019): 3488-3498.

Ou, Xinming, Sudhakar Govindavajhala, and Andrew W. Appel. "MulVAL: A Logic-based Network Security Analyzer." USENIX security symposium. Vol. 8. 2005.

Bezawada, Bruhadeshwar, Indrajit Ray, and Kushagra Tiwary. "AGBuilder: An AI Tool for Automated Attack Graph Building, Analysis, and Refinement." IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2019.

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov