

AISeckG: Knowledge Graph Dataset for Cybersecurity Education

Garima Agrawal^{1,*}, Kuntal Pal¹, Yuli Deng¹, Huan Liu¹ and Chitta Baral¹

¹*School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA*

Abstract

Cybersecurity education is exceptionally challenging as it involves learning the complex attacks; tools and developing critical problem-solving skills to defend the systems. For a student or novice researcher in the cybersecurity domain, there is a need to design an adaptive learning strategy that can break complex tasks and concepts into simple representations. An AI-enabled automated cybersecurity education system can improve cognitive engagement and active learning. Knowledge graphs (KG) provide a visual representation in a graph that can reason and interpret from the underlying data, making them suitable for use in education and interactive learning. However, there are no publicly available datasets for the cybersecurity education domain to build such systems. The data is present as unstructured educational course material, Wiki pages, capture the flag (CTF) writeups, etc. Creating knowledge graphs from unstructured text is challenging without an ontology or annotated dataset. However, data annotation for cybersecurity needs domain experts. To address these gaps, we made three contributions in this paper. First, we propose an ontology for the cybersecurity education domain for students and novice learners. Second, we develop AISeckG, a triple dataset with cybersecurity-related entities and relations as defined by the ontology. This dataset can be used to construct knowledge graphs to teach cybersecurity and promote cognitive learning. It can also be used to build downstream applications like recommendation systems or self-learning question-answering systems for students. The dataset would also help identify malicious named entities and their probable impact. Third, using this dataset, we show a downstream application to extract custom-named entities from texts and educational material on cybersecurity.

Keywords

Knowledge Graph, Cybersecurity Education, Ontology, Knowledge Base, KG Dataset, Language Model

1. Introduction

Learning cybersecurity requires mastering the academic content and developing critical thinking and problem-solving skills based on cyber attacks and defense scenarios. We can achieve this interactive and active learning by creating an AI-powered education system where students can control their learning process [1, 2, 3]. Knowledge graphs have been effectively used in education and improving the learning experience [4].

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.


*Corresponding author.

✉ garima.agrawal@asu.edu (G. Agrawal); kkp@asu.edu (K. Pal); ydeng19@asu.edu (Y. Deng); huanliu@asu.edu (H. Liu); chitta@asu.edu (C. Baral)

ORCID 0000-0002-4383-7850 (G. Agrawal); 0000-0003-1278-3252 (K. Pal); 0000-0001-7715-9966 (Y. Deng)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

A knowledge graph combines two things, a graph with domain-specific data and an explicit representation of knowledge. The graph can capture the domain-related key concepts and their interactions with each other. It allows the user to analyze and understand the connections or relationships between different entities. Using explicit knowledge or metadata provides the relevant background and important information about the domain. This metadata allows the system to establish a common vocabulary and use shared references. The knowledge graphs are thus an integrated tool that can use the underlying data and knowledge for concept visualization and contextual reasoning [5]. Their ability to translate data into usable knowledge makes them suitable for education. They can promote cognitive engagement in the problem-based learning environment.

However, to build such systems, there is a need for annotated datasets. There are no public datasets in the cybersecurity education domain. The education material includes unstructured texts in lecture notes, lab manuals, Wiki pages, capture the flag (CTF) writeups, and others. Scraping unstructured text and creating domain-specific knowledge graphs is challenging, especially without standard ontology and annotated datasets. The task of annotating data is expensive and time-consuming as it can be done only by cybersecurity domain experts accurately. The increase in demand for cybersecurity professionals requires preparing an effective cybersecurity specialist workforce and equipping them with intelligent learning tools. A comprehensive dataset with cybersecurity-related named entities is a significant bottleneck in this area.

In this paper, we address this issue by making three main contributions. First, using domain knowledge, we propose an ontology for self-paced cybersecurity learning for novice users. Second, we create an annotated named entity dataset. Using this dataset, we show one downstream application to extract named entities from texts and educational material on cybersecurity. Third, we present a triple dataset AISecKG, for cybersecurity education as defined by our ontology. Using the triples data, we show one downstream task to construct a concept flow graph for a cybersecurity tool. It is possible to write graph queries to generate sub-graphs focusing on the specific learning needs of a user. Also, by combining the AISecKG ontology schema and the cybersecurity named entities, knowledge graphs can be created from any unstructured texts on cybersecurity. Our ontology and labeled dataset can also be used to build applications like question-answering and recommendation systems for students.

The paper is organized as follows. In the next section, we discuss the related work. **Section 3** describes the method and ontology. **Section 4** presents our work's results and two applications. Finally, we conclude the paper in **Section 5**.

2. Related Work

Security plays an integral role in software development and has become more critical with the Internet of Things (IoT). Various studies to formalize security and develop security ontologies and knowledge models address different security aspects. Souag et al. [6] gave a security ontology to elicit *security requirements*. The Unified Cybersecurity Ontology (UCO) [7] focused on identifying the *vulnerabilities and threat levels* to assess the system security. Doynikova [8] proposed an ontology on *security metrics for cybersecurity assessment* to determine the attack

goal. An extensive study on *formalizing information security* focuses on security concepts, and threat mitigation and control process [9]. A cybersecurity ontology was proposed to build and monitor the *security in the cloud* for IoT environment [10]. Iannacone et al. [11] developed an ontology for managing cybersecurity knowledge database from different data sources to propose a *search mechanism for blacklisted systems*. MALOnt [12] gives the ontology and knowledge graphs for *malware threat intelligence*. Martins et al. [13] presented a conceptual characterization of available cybersecurity ontologies based on their application. In this work, we introduce an ontology AISeckG which covers a broader spectrum of the fundamental concepts, tools, techniques, and applications used in the cybersecurity ecosystem. Essentially this ontology is helpful for any first-time user or new learner in the cybersecurity domain.

Many datasets have also been proposed in the cybersecurity domain, but most are based on network flow data and are used to train machine learning algorithms to build intrusion detection systems. Alshaibi et al. [14] gave a comparative study of these datasets. Recently more efforts have been made to create cybersecurity named-entity datasets. A new dataset for event detection in cybersecurity texts [15] annotated 30 types of critical events in cybersecurity to train the machine learning models. A named-entity recognition (NER) python library called CyNer [16] based on MALOnt ontology [12] was developed to extract the malware and threat indicators. Language models were also built for cybersecurity [17, 18, 19] using the open source CVE [20], and NVD Mitre [21] datasets on vulnerability and attacks. Dasgupta et al. [22] gave a comparative study on NER algorithms based on these datasets for cybersecurity. In our current work, we develop a labeled named-entity dataset for cybersecurity based on AISeckG ontology which is used to build knowledge graphs from any unstructured text on cybersecurity.

Most of the available cybersecurity ontology and datasets are used to build intrusion detection systems or perform vulnerability analysis and threat detection. Table 1 compares our ontology and cybersecurity dataset with existing works. There is limited research to educate novice learners on cybersecurity concepts, tools, and techniques. Deng et al. [23] proposed using knowledge graphs as lab project guidance to teach cybersecurity. They focused on finding similar concepts on the web using similarity measures [24] and word embeddings [25]. In our paper [26], we proposed a semi-automated approach to build knowledge graphs from the unstructured cybersecurity course material and conducted a survey and interview with students to assess the perception of students on using knowledge graphs as a problem-solving education tool aid. The students found the knowledge graphs very useful, which motivated us to propose AISeckG, a comprehensive ontology and a labeled dataset that can be used to build AI systems to learn about cybersecurity. In this work, we give a detailed ontology to understand the cybersecurity ecosystem from different views and present an annotated named-entity recognition dataset to extract cybersecurity-related entities.

3. Method for Development of AISeckG

Cybersecurity is the application of state-of-the-art technologies, control processes, policies, tools, and procedures for protecting or recovering systems and information from malicious attacks[27]. As a novice learner, one must know the cybersecurity ecosystem comprising fundamental concepts, tools, and techniques and how to use and deploy them to assess and

Cybersecurity Model	Purpose	Ontology	Dataset
Souag et al. [6]	Security Requirement Elicitation	✓	X
UCO [7]	Vulnerability Assessment	✓	X
Doynikova et al. [8]	Security Metrics	✓	X
Fenz et al. [9]	Threat Mitigation and control	✓	X
Mozzaquatro et al. [10]	IoT Security Monitoring	✓	X
Iannacone et al. [11]	Search cybersecurity knowledge base	✓	X
Alshaibi et al. [14]	Intrusion Detection Models	X	Network Flow datasets
Tikhomirov et al. [18]	Vulnerability/Attack detection	X	Open Source (CVE/NVD)
Ma et al. [19]	Vulnerability/Attack detection	X	Open Source (CVE/NVD)
Gao et al. [17]	Vulnerability/Attack detection	X	Open Source (CVE/NVD)
Trong et al. [15]	Event Detection	X	✓ (Annotated dataset)
MALOnt [12]	Malware Threat Intelligence KG	✓	✓ (Annotated dataset)
CyNer [16]	Malware Threat Entity Extraction	X (using MALOnt)	X
AISeckG	Cybersecurity education KG	✓	✓ (Annotated Triples KG Dataset)

Table 1

Comparative analysis of existing ontologies and datasets on cybersecurity.

detect vulnerabilities and attacks. This section presents our ontology design and dataset for cybersecurity education called AISeckG.

We propose a comprehensive view of concepts, applications, and roles involved in the cybersecurity ecosystem. Since the objective is to build a self-paced learning tool for cybersecurity students and novice learners, we use the graduate-level course material and hands-on lab instruction manuals to teach graduate students majoring in cybersecurity as the data source. Our ontology, AISeckG, is built using domain knowledge and motivated by the lab guides. The dataset is created by annotating the lab documents using AISeckG ontology. We then use this annotated dataset to develop two applications. The first application is to *train a language model on our dataset to extract named entities related to cybersecurity*. The second application is to create a *triple dataset with entity-relation-entity pairs to construct knowledge graphs*. Both these applications are described in the next section.

3.1. Data Source

We collected data from the laboratory instruction manuals. The manuals are for projects of advanced cybersecurity courses for graduate students. These courses cover topics such as using tools like NMap and Snort [28] to build intrusion detection systems, employing honeypot techniques in Metasploit framework to deceive attackers, setting up Kali Linux systems, and monitoring system activities and attack events using Syslog. The manuals are in standard English and explain the concepts and instructions for implementing laboratory tasks. Each manual is 15-20 pages long. For annotation, we used six such lab manuals with a total of 100 pages with approximately 26886 words and 110953 characters.

3.2. Ontology

Ontology is a formal and explicit schematic representation of a system using a well-defined taxonomy. It allows semantic modeling of the domain knowledge and thus can be used as the skeleton to build any AI application for that system [29]. Ontology also defines the rules and

constraints of the system and facilitates the validation of semantic relationships and conclusions or inferences from known facts. For a knowledge-based system, ontology serves as a backbone of the system and should be meticulously developed.

Some deep learning-based methods rely on automatically building the AI application from the data without using an ontology, but they fail to capture the comprehensive view of the domain, and the quality of applications becomes questionable [30]. On the other hand, if a well-defined structured dataset is unavailable and most of the knowledge is present in unstructured texts, it is overwhelming for a domain expert to scrape long texts and create a domain-specific ontology. Also, it is costly to find domain experts in cybersecurity.

The domain experts should have practiced or significantly demonstrated sufficient knowledge and experience. In this work, the first and second authors are graduate researchers in cybersecurity, and the third author is a cybersecurity expert and instructor. He teaches graduate-level cybersecurity courses at his university.

To develop AISecKG ontology, we used the bottom-up approach given in the paper [26]. We used the lab documents as a reference and then used domain knowledge to design the ontology. First, we extracted the generic entities and relations from the lab documents using the parts of speech tagging and the dependency parsing given by spacy-based named-entity recognition (NER) [31] natural language processing (NLP) methods. The entities extracted using NER are the subject-object pairs, and relations are the predicates in the sentences. These entity-relation-entity triples are not specific to cybersecurity, but they help break down the long texts into simple graph-like structures and create a preliminary visual representation of information. They serve as a good reference point for domain experts. This step semi-automated the ontology construction process and significantly reduced time and effort. It helped in discovering the schematic and semantic relationships of core entities.

3.2.1. Key Entities

The cybersecurity ecosystem essentially has three foundational pillars, namely, **concept, application, and role**. We can classify *concepts* into features, functions, data, attacks, vulnerabilities, and techniques. In addition to defensive and attack methods, the techniques here include security policies and management processes. The *application* denotes the tools, systems, and apps. The user, attacker, and securityTeam are the three *roles*. Thus in our ontology, we defined these three categories with 12 types of entities.

Figure 1 depicts the cybersecurity education ecosystem with each category and entity type. The attributes or metadata considered for the entities are *entityID*, *entityName*, *entityType*, and *entityCategory*. The examples from each entity type within the respective category are shown in **Table 3.2.1**.

3.2.2. Relations

We used the nine most common and appropriate relations to represent the real-world interactions between cybersecurity entities. **Table 3** shows the relations along with examples from our dataset as entity-relation-entity triples.

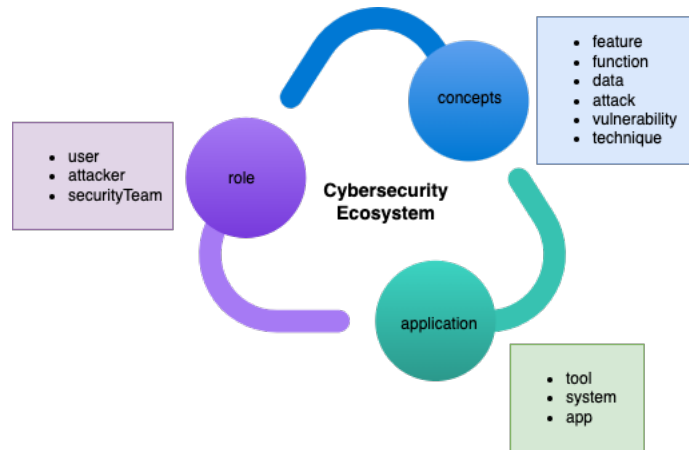


Figure 1: Cybersecurity Education Ecosystem: Concepts, Roles, Applications

Table 2

The table shows the category and types of key entities in the ontology with examples for each.

Category	Type	Examples of Entity Names
Concept	feature function attack vulnerability technique data	session ID, cookies, protocol tcpdump, snort rules, hash, XOR smurf attack, sql injection, spyware bad config, weak password honeypot, security policy, risk assessment files, logs, message, packet
Application	tool system app	burp, wireshark, snort, sniffer linux, server, client, host browser, webapp, service
Role	attacker securityTeam user	black hat, attack host security engineer, white hat employee, user

3.2.3. Cybersecurity Schema

We now present the schema design for learning cybersecurity. We illustrate the interactions between different components from the perspective of the roles. **Figure 2** shows the user’s view. It depicts how users use the data, applications, and systems routinely. The system and apps, in turn, use different tools for their usual operations and defensive techniques to monitor and analyze the environment. The icons in the diagram represent the respective entities, and the labeled edges show the relationship between different entities.

Figure 3 gives the attacker view. When the applications expose vulnerabilities and the attacker can exploit them using various tools and attack techniques, the attacker and attacks can harm the data and applications.

The third view is the security view. **Figure 4** shows how the security team uses tools and

Table 3

The table shows the sample triples from the dataset in entity-relation-entity form as per the schema.

Relation	Sample Triples
has_a	Nmap has_a network mapper
can_analyze	Packet Decoder can_analyze header anomaly
can_expose	Intel CPU can_expose CVE-2017-5754
can_exploit	Attack host can_exploit TCP syn packet
implements	Network administrators implements map
uses	Team defense uses firewall
can_harm	Attack can_harm target host
can_detect	Full scan can_detect Trojan horses
is_part_of	Metasploit Framework is_part_of Kali Linux

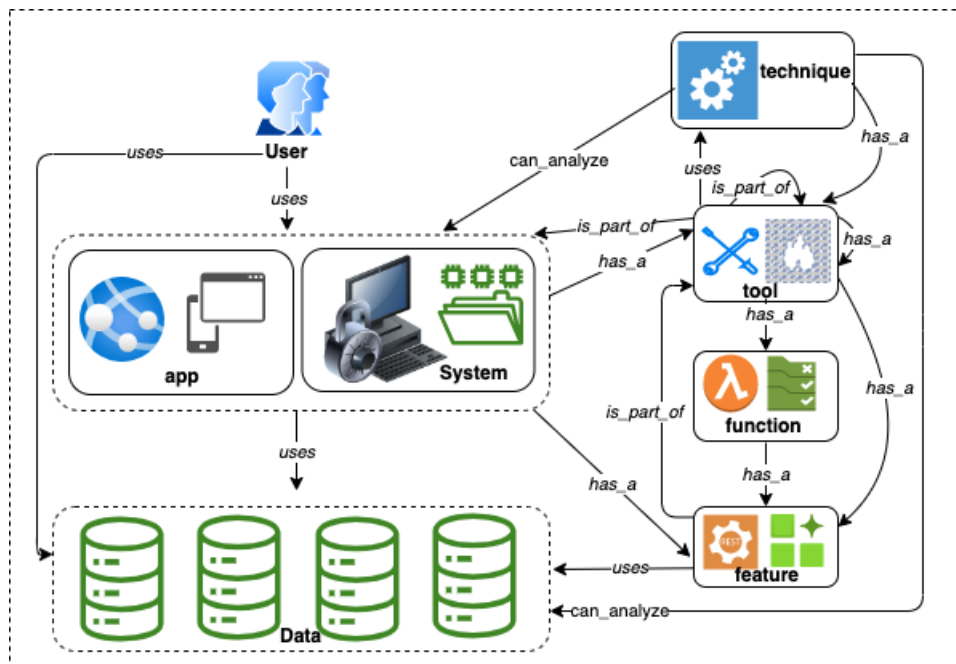


Figure 2: User View shows the interaction of users with apps, system and data which in turn use different tools and techniques.

defensive techniques to analyze and detect vulnerabilities and attacks.

The three views shown in **Figure 2, 3 and 4** give the landscape of cybersecurity concepts, tools, techniques, systems, and policies that are required for a novice learner to gain an understanding of the domain. AISeckG ontology identifies 68 schema edges or interactions among the 12 types of entities shown in the respective figures.

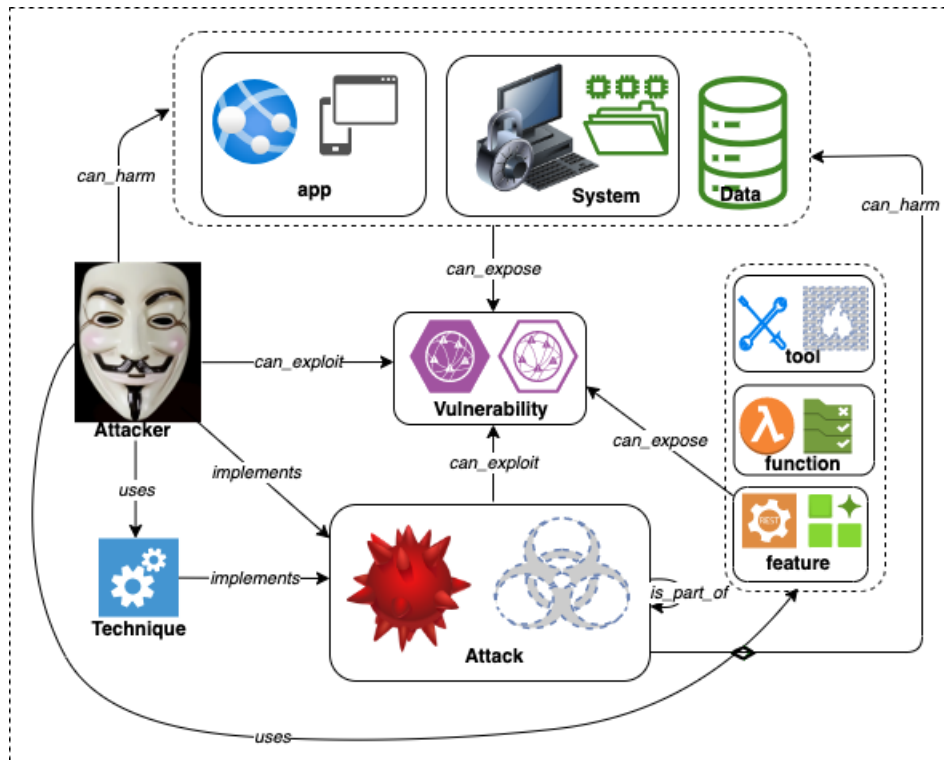


Figure 3: Attacker View shows the interactions between different entities when a system is exposed to attacks.

3.3. AISeckG Dataset Annotation

Using the AISeckG Ontology, we identified 964 cybersecurity-related unique entities from the course materials. There are 12 entity types in three categories in the ontology. We labeled the attributes, entity Id, entity type, and entity category against each entity and created an entity info list. To train the model to predict custom cybersecurity-related entities, we created the annotated dataset using BIO (Beginning-Inside-Outside) sequence tagging scheme [32]. The entity boundary is defined by tags 'B' and 'I' called Beginning and Inside the label. All the words other than entity are labeled as 'O'. The lab documents were first split into sentences using a simple python script. There were 593 sentences, and 2354 entities were annotated in these sentences. The code and commands were discarded from the text. The annotation was done by the first and second authors and was validated by the third author.

4. Applications of AISeckG

4.1. NLP Language Model to Extract custom Named-Entities

Here we present the first application of AISeckG. Using our AISeckG dataset and its Named Entity annotations, automated systems can be developed to help identify the named entities

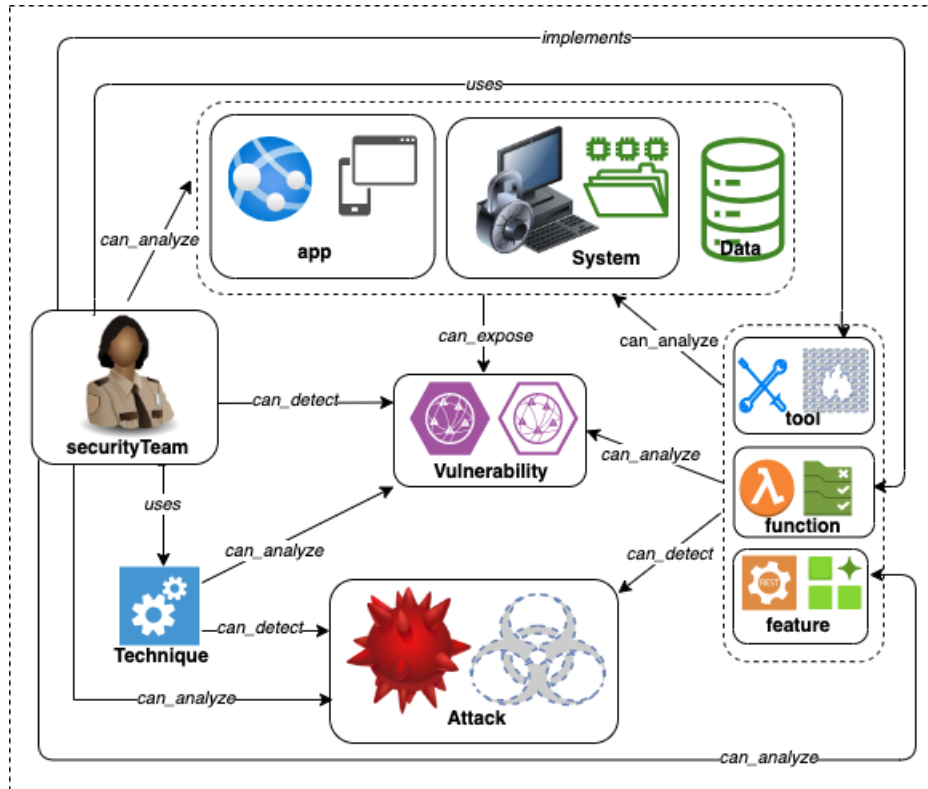


Figure 4: Security View shows the vulnerability and attack analysis by security team using different tools and techniques.

from public texts related to the education cybersecurity ecosystem.

4.1.1. Dataset Preparation:

We split the AISeckG annotated dataset into train, dev, and test keeping 3, 1, and 2 documents, respectively. The train, dev, and test splits contain 5772, 3591, and 195 entities in 372, 214, and 13 sentences, respectively. We keep an empty line as a separator for each cybersecurity sentence. This dataset is provided as input to each model.

4.1.2. Models

We experiment with six variations of two transformer-based language models: BERT [33], and RoBERTa [34]. For BERT, we use cased and uncased versions of the base (110M parameters) and large (340M parameters) variations, and for RoBERTa, we use both the base (125M parameters) and large (355M parameters) models. The BERT-base and RoBERT-base architectures have 12 layers, 12 attention heads, and 768 hidden dimensions, whereas both the BERT-large and RoBERTa-large have 24 layers, 16 attention heads, and 1024 hidden dimensions.

First, the model tokenizes the input sentence and generates embeddings of the tokens. Then we consider the *sequence labeling approach* of the language models, that is, classifying each

Metric	BERT-base-uncased	BERT-large-uncased	BERT-base-cased	BERT-large-cased	RoBERTa-base	RoBERTa-large
Accuracy (%) ↑	81.91	82.17	81.43	83.30	80.63	82.71
Precision ↑	45.69	45.49	47.32	48.73	44.20	47.97
Recall ↑	51.58	53.81	51.70	56.04	48.65	51.23
F1-score ↑	48.46	49.30	49.41	52.13	46.32	49.55

Table 4

Performance of BERT and RoBERTa on the AISeckG dataset: Bold represents best performance, higher value is better for each metric

token of a given sentence into any one of the 25 classes (12 entity types with B and I tags along with O representing other. We aggregate the classified continuous beginning and intermediate tokens into entities. In this approach, we not only extract the entities but also identify the type of these entities.

4.1.3. NER Results

We train each model for 30 epochs with a maximum sequence length of 128 per GPU batch size of 32. Table 4.1.3 shows the performance of our sequence classification on the test set. It can be seen from the table that case-sensitive BERT performs best in all the metrics. This shows that case sensitivity positively impacts the cybersecurity NER model. As expected, all smaller versions of the models perform comparatively poorly compared to their larger counterparts because of less number of parameters. Our accuracy in predicting the entities is over 80% across all the models. Our precision, recall, and F1 scores are pretty good, given the fewer training samples and many diverse class categories. This shows the effectiveness of our model in identifying entities involved in our ontology from cybersecurity texts.

4.2. Triples for Knowledge Graph

The second application uses the annotated dataset to create triples for the knowledge graphs. The triples are a way to store the graph data in the form of 'entity-relation-entity,' where the entity represents the nodes, and the relation represents the labeled edge. The triples data can be used to construct knowledge graphs to provide a visual representation. For this work, since the focus is to provide learning aids to students, we build visual concept graphs from the lab documents using these triples. The conceptual graphs help break down complex information and allow the students to visually analyze the underlying concepts and the interconnections between different concepts.

The constraints and rules of creating edges were defined based on the schema definition of AISeckG Ontology. There are 68 schema edges between the 12 entity types as per the schema given in **Section 3**. We use the annotated sentences from lab documents. The relations between labeled entities in each sentence were extracted automatically by matching with the tuples in the ontology. We manually validated the triples and removed the redundant and ambiguous triples. Around 812 triples were auto-generated, which were reduced to 730 triples after validation in the final dataset. **Table 5** gives a list of sample triples from the dataset for each tuple in our ontology.

Figure 5 shows one of the sub-graphs generated using the subset of triples. This graph shows the knowledge graph on NMap tool. The visual graphs related to a specific entity can be created by writing the graph queries. Any graph database, such as RDF or Neo4j, can be used, and graph query languages like GraphQL, SPARQL, or CyPher can query and generate the graphs [35]. We have used the Networkx library in Python to generate the graph. We store the triple in a csv file to make it publicly available. The triple dataset, annotated data and implementation code for both applications are available in our github repository¹. Thus the ontology and labeled entities in AISecKG can be used to create knowledge graphs from any unstructured texts on cybersecurity by extracting the cybersecurity-related named entities using the model and relations per the ontology.

5. Conclusion and Future Works

In this work, we present a novel ontology on Cybersecurity Education, **AISecKG**, and show that this ontology is vital to building self-paced AI-based learning tools for cybersecurity learners. More research must be done in this direction as these tools can be crucial to prepare the cybersecurity specialist workforce. Additionally, we introduce a manually annotated named entity dataset based on ontology. We also show how our AISecKG can be used in downstream tasks. First, we present how the language models can be trained with our annotated dataset to extract cybersecurity-related named entities from the cybersecurity documents. There are minimal works [36] on extracting such information from public forum cybersecurity learning materials written by professionals for novice vulnerability researchers. We want to extend this work beyond lab manuals to cybersecurity educational texts in public forums. Secondly, we show the process of creating triples by automatically extracting the relations based on the schema definition given by the ontology. We present one application as the construction of knowledge graphs from triple data for concept visualization. Other downstream applications like question-answering systems and learning recommendation systems can be built using the triple dataset.

Acknowledgments

We are thankful to National Science Foundation under Grant No. 2114789 for supporting this research work. We would also like to acknowledge Dijiang Huang for his vision and guidance.

References

- [1] G. Thomas, D. Anderson, S. Nashon, Development of an instrument designed to investigate elements of science students' metacognition, self-efficacy and learning processes: The semli-s, *International Journal of Science Education* 30 (2008) 1701–1724.
- [2] R. R. BRIEF, J. LY, B. S. E. A. ION, A framework for k-12 science education: Practices, crosscutting concepts, and core ideas (2012).

¹<https://github.com/garima0106/AISecKG-cybersecurity-dataset.git>

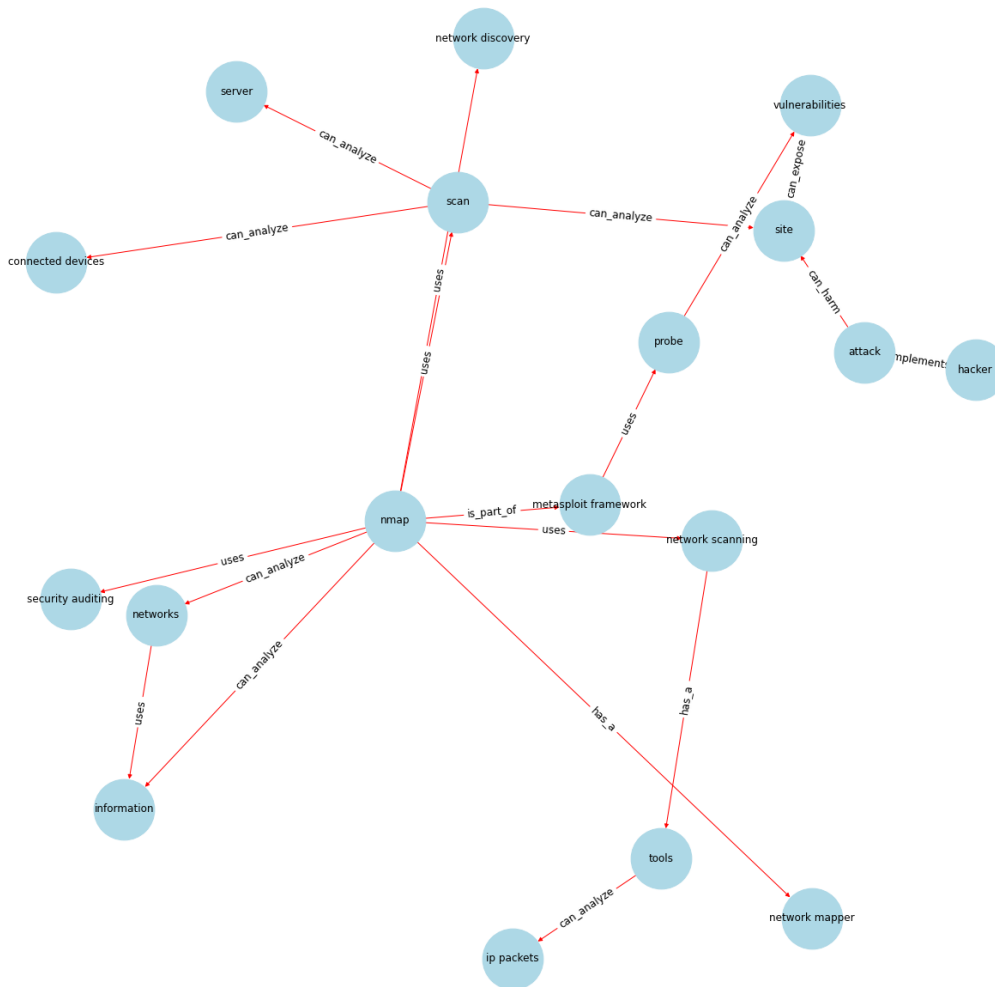


Figure 5: A sub-graph on Nmap to show Knowledge graph generated from the triples in dataset

- [3] N. Shah, P. Verma, T. Angle, S. Srivastava, Jedai: A system for skill-aligned explainable robot planning, in: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS '22, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2022, p. 1917–1919.
- [4] Y. Fettach, M. Ghogho, B. Benatallah, Knowledge graphs in education and employability: A survey on applications and techniques, IEEE Access (2022).
- [5] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (CSUR) 54 (2021) 1–37.

- [6] A. Souag, C. Salinesi, R. Mazo, I. Comyn-Wattiau, A security ontology for security requirements elicitation., in: ESSoS, Springer, 2015, pp. 157–177.
- [7] Z. Syed, A. Padia, T. Finin, L. Mathews, A. Joshi, Uco: A unified cybersecurity ontology, UMBC Student Collection (2016).
- [8] E. Doynikova, A. Fedorchenko, I. Kotenko, Ontology of metrics for cyber security assessment, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019, pp. 1–8.
- [9] S. Fenz, A. Ekelhart, Formalizing information security knowledge, in: Proceedings of the 4th international Symposium on information, Computer, and Communications Security, 2009, pp. 183–194.
- [10] B. A. Mozzaquatro, C. Agostinho, D. Goncalves, J. Martins, R. Jardim-Goncalves, An ontology-based cybersecurity framework for the internet of things, *Sensors* 18 (2018) 3053.
- [11] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, J. Goodall, Developing an ontology for cyber security knowledge graphs, in: Proceedings of the 10th Annual Cyber and Information Security Research Conference, 2015, pp. 1–4.
- [12] N. Rastogi, S. Dutta, M. J. Zaki, A. Gittens, C. Aggarwal, Malont: An ontology for malware threat intelligence, in: Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020, Proceedings 1, Springer, 2020, pp. 28–44.
- [13] B. F. Martins, L. Serrano, J. F. Reyes, J. I. Panach, O. Pastor, B. Rochwerger, Conceptual characterization of cybersecurity ontologies, in: The Practice of Enterprise Modeling: 13th IFIP Working Conference, PoEM 2020, Riga, Latvia, November 25–27, 2020, Proceedings 13, Springer, 2020, pp. 323–338.
- [14] A. Alshaihi, M. Al-Ani, A. Al-Azzawi, A. Konev, A. Shelupanov, The comparison of cybersecurity datasets, *Data* 7 (2022) 22.
- [15] H. M. D. Trong, D.-T. Le, A. P. B. Veyseh, T. Nguyen, T. H. Nguyen, Introducing a new dataset for event detection in cybersecurity texts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 5381–5390.
- [16] M. T. Alam, D. Bhusal, Y. Park, N. Rastogi, Cyner: A python library for cybersecurity named entity recognition, arXiv preprint arXiv:2204.05754 (2022).
- [17] C. Gao, X. Zhang, H. Liu, Data and knowledge-driven named entity recognition for cyber security, *Cybersecurity* 4 (2021) 1–13.
- [18] M. Tikhomirov, N. Loukachevitch, A. Sirotina, B. Dobrov, Using bert and augmentation in named entity recognition for cybersecurity domain, in: Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings 25, Springer, 2020, pp. 16–24.
- [19] P. Ma, B. Jiang, Z. Lu, N. Li, Z. Jiang, Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields, *Tsinghua Science and Technology* 26 (2020) 259–265.
- [20] C. V. CVE, N. Exposures, Url <http://cve.mitre.org>, Accessed in January (2014).
- [21] C. MITRE, National vulnerability database (nvd), <https://nvd.nist.gov/> (2017).
- [22] S. Dasgupta, A. Piplai, A. Kotal, A. Joshi, A comparative study of deep learning based named entity recognition algorithms for cybersecurity, in: 2020 IEEE International Conference

- on Big Data (Big Data), IEEE, 2020, pp. 2596–2604.
- [23] Y. Deng, D. Lu, D. Huang, C.-J. Chung, F. Lin, Knowledge graph based learning guidance for cybersecurity hands-on labs, in: Proceedings of the ACM conference on global computing education, 2019, pp. 194–200.
 - [24] Y. Deng, Z. Zeng, K. Jha, D. Huang, Problem-based cybersecurity lab with knowledge graph as guidance, *Journal of Artificial Intelligence and Technology* 2 (2022) 55–61.
 - [25] Y. Deng, Z. Zeng, D. Huang, Neocyberkg: enhancing cybersecurity laboratories with a machine learning-enabled knowledge graph, in: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, 2021, pp. 310–316.
 - [26] G. Agrawal, Y. Deng, J. Park, H. Liu, Y.-C. Chen, Building knowledge graphs from unstructured texts: Applications and impact analyses in cybersecurity education, *Information* 13 (2022) 526.
 - [27] D. Craigen, N. Diakun-Thibault, R. Purse, Defining cybersecurity, *Technology Innovation Management Review* 4 (2014).
 - [28] M. Roesch, et al., Snort: Lightweight intrusion detection for networks., in: *Lisa*, volume 99, 1999, pp. 229–238.
 - [29] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs., *SEMANTiCS (Posters, Demos, SuCCESS)* 48 (2016) 2.
 - [30] M. Kejriwal, *Domain-specific knowledge graph construction*, Springer, 2019.
 - [31] Y. Vasiliev, *Natural language processing with Python and spaCy: A practical introduction*, No Starch Press, 2020.
 - [32] E. F. Sang, J. Veenstra, Representing text chunks, *arXiv preprint cs/9907006* (1999).
 - [33] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
 - [34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, *arXiv preprint arXiv:1907.11692* (2019).
 - [35] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, D. Vrgoč, Foundations of modern query languages for graph databases, *ACM Computing Surveys (CSUR)* 50 (2017) 1–40.
 - [36] K. K. Pal, K. Kashihara, P. Banerjee, S. Mishra, R. Wang, C. Baral, Constructing flow graphs from procedural cybersecurity texts, *arXiv preprint arXiv:2105.14357* (2021).

	Triples (e1, r, e2)	Example
User	(user uses app) (user uses system) (user uses data) (app uses data) (system uses data) (system has_a tool) (app has_a tool) (tool is_part_of system) (tool is_part_of app) (tool is_part_of tool) (tool has_a tool) (tool has_a function) (tool has_a feature) (tool uses technique) (technique has_a tool) (technique can_analyze system) (technique can_analyze app) (technique can_analyze data) (function has_a feature) (feature is_part_of tool) (system has_a feature) (app has_a feature) (feature uses data)	vendors uses application users uses network employee uses files applications uses packets machine uses network traffic Windows XP has_a Nping Webapp has_a SQLMAP Firewall is_part_of network system WAF s_part_of webapp Nmap is_part_of Metasploit frameworks Snort has_a Packet Decoder snort has_a rules Nmap has_a flag Nmap uses RPC scanning three-way handshake has_a port scanner Scan can_analyze connected devices Web Pen testing can_analyze webapp ACK scan can_analyze TCP packets Snort rules has_a payload options System configurations is_part_of firewall networks has_a IP RPC services has_a port iptables uses logs
attacker	(app can_expose vulnerability) (system can_expose vulnerability) (data can_expose vulnerability) (feature can_expose vulnerability) (attacker can_exploit vulnerability) (attacker uses feature) (attacker uses function) (attacker uses tool) (attacker implements attack) (attacker uses technique) (technique implements attack) (attacker can_harm app) (attacker can_harm data) (attacker can_harm system) (attack is_part_of attack) (attack can_harm system) (attack can_harm app) (attack can_harm data)	services can_expose vulnerabilities target host can_expose spoofed data packets can_expose forged sessionID can_expose websession vulnerability attackers can_exploit security holes attacker uses established connections hacker uses malicious scripts attackers uses Nmap cybercriminals implements hack attacker uses packet filtering sniffing traffic implements man in middle attack bad guys can_harm services attacker can_harm data packets hacker can_harm target DOS is_part_of floods denial-of-service attacks can_harm target fuzzing attacks can_harm apps Forging can_harm RST packets
securityTeam	(securityTeam can_analyze app) (securityTeam can_analyze data) (securityTeam can_analyze system) (securityTeam uses tool) (securityTeam implements function) (securityTeam can_analyze feature) (securityTeam uses technique) (securityTeam can_detect vulnerability) (securityTeam can_analyze attack) (tool can_analyze system) (tool can_analyze app) (tool can_analyze data) (tool can_analyze vulnerability) (tool can_detect attack) (tool can_analyze feature) (function can_analyze vulnerability) (function can_analyze system) (function can_analyze app) (function can_analyze data) (function can_detect attack) (feature can_analyze system) (feature can_analyze app) (feature can_analyze data) (feature can_analyze vulnerability) (technique can_analyze vulnerability) (technique can_detect attack)	White hats can_analyze webapps ethical hackers can_analyze logs pen tester can_analyze client VM network operators uses scanners system admins implements programs pen tester can_analyze iptables penetration tester uses brute force pen tester can_detect configuration vulnerabilities white hats can_analyze pentesting attacks Nmap can_analyze enterprise-scale networks Metasploit can_analyze security services Snort can_analyze traffic Preprocessor can_analyze HTTP anomalies Nmap can_detect malicious attacks Snort can_analyze network behavior post scan scripts can_analyze vulnerabilities remote commands can_analyze machine Scripts can_analyze 3rd-party services Snort rules can_analyze system variables dos script can_detect Denial of service attack flag can_analyze system ports can_analyze services established connections can_analyze packets ACK bit IP address can_analyze zombie host URL can_detect attack idle scan can_analyze spoofing Scanning can_detect Packet Forgery

Table 5

The table shows the triples generated from the labeled text. The relations were extracted using the ontology defined in Section 3