NeoCyberKG: Enhancing Cybersecurity Laboratories with a Machine Learning-enabled Knowledge Graph

Yuli Deng
Arizona State University
Tempe, Arizona
yuli.deng@asu.edu

Zhen Zeng
Arizona State University
Tempe, Arizona
zzeng22@asu.edu

Dijiang Huang
Arizona State University
Tempe, Arizona
dijiang@asu.edu

ABSTRACT

The hands-on lab is a critical component of cybersecurity education. There lacks of a coherent way to manage existing labs to provide a practical learning plan for learners in the cybersecurity area. Previous studies utilized the word embedding technologies to construct a knowledge graph and adopt it as a learning guide for students, but this approach has its limitations. In this paper, we present a new approach based on latent semantic analysis (LSA) method to replace word embedding in previous studies as it is more appropriate in a small-size corpus, and it is also able to create a mapping that connects both the topic of each lab and concepts contained in each lab. We use LSA to identify relevant semantic relations, extract relevant lab problems, and construct knowledge graphs from lab contents related to cybersecurity topics. We utilize the output of this study by establishing a web-based lab environment for students that: 1. providing lab index and searching, which contains concepts and knowledge extract from each lab. 2.building a recommendation/guidance system for cybersecurity labs and suggesting more relevant labs based on users learning preferences and past lab history to maximize learning outcomes. To measure the effectiveness of the proposed solution, we conducted a use case study and collected survey data from a graduate-level cybersecurity class at a public university. Our study shows that users tend to gain enhanced learning outcomes and express more interest in the cybersecurity area by leveraging the knowledge graph as a learning guide.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Applied computing** → **E-learning**; *Interactive learning environments*;

KEYWORDS

Laboratory, Knowledge Graph, Cybersecurity

ACM Reference Format:
Yuli Deng, Zhen Zeng, and Dijiang Huang. 2021. NeoCyberKG: Enhancing Cybersecurity Laboratories with a Machine Learning-enabled Knowledge Graph. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2021), June 26-July 1, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3430665.3456378

1 INTRODUCTION

Many studies have been conducted on developing a cybersecurity curriculum or guide for universities: [16], [22], [10], [12]. Furthermore, a multitude of frameworks and learning objectives for cybersecurity have been established (e.g., CAE-CO [17], NICE Cybersecurity Workforce Framework (NCWF) [14], ACM Joint Task Force on Cybersecurity Education [2]). Nevertheless, there is still a significant gap in maintaining and updating cybersecurity instruction guide at a practical level.

Frameworks such as NCWF and CAE-CO provide a detailed listing of knowledge and concepts required to succeed in a cybersecurity career. These sources of material are solid and are increasingly being recognized. However, adopts the baseline requirements or objectives of these frameworks makes learning mainly focus on science and literature topics instead of hands-on practical learning skills. Many institutes that offer cybersecurity programs still require a comprehensive guide to improve established learning guidelines.

To meet these challenges, researchers adopted the knowledge graph as an AI tool to generate learning guides in an automotive fashion for students [7], [18]. Knowledge graph technology has drawn a lot of research attention in recent years [21]. Furthermore, information extraction and recommendation system are among the most popular real-world applications of the knowledge graph. However, these approaches have their limitations. [18] requires significant human input during the knowledge graph construction stage to reduce errors, limiting feasibility in real-world applications for complex education areas. [7] uses embedding-based relation extraction approach to generate knowledge graph automatically from text data, but suffers in accuracy and reliability due to its limited data source size as word embedding requires large-size text corpora to perform well.

The issues described above inspired us to design a new learning guide solution based on a knowledge graph. We focus on the hands-on lab, as it is a critical component for cybersecurity education and provides good integration of cybersecurity topics and relating them to real-life practice. But there are a few challenges: First, it is more challenging to organize lab materials than textbooks, let alone manage complicated concept indexes in labs. Second, due to inherent diversities in problems and tasks in cybersecurity lab contents, it is not easy to guide the learning process and keep tracking of students' learning progress. Third, for instructors, the knowledge-sets and problems must be kept up-to-date to cope with emerging vulnerabilities, attacks, and defense solutions. Fourth, due to the

ITiCSE 2021, June 26-July 1, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8214-4/21/06…$15.00
https://doi.org/10.1145/3430665.3456378

inherent diversity in knowledge and skill sets in cybersecurity education, it requires significant effort to gather text corpora for NLP study. To address the above-described challenges, we proposed NeoCyberKG, a cybersecurity knowledge graph for college-level education, including learning-related and domain-specific knowledge. Our contribution in this paper is given as follows:

1) We built a knowledge graph for hands-on labs to present embedded cybersecurity concepts and terminologies in the lab to illustrate these labs' detailed knowledge. Instead of the embedding-based approach [3] used in previous works, nodes of the knowledge graph and their dependency relationship are obtained by the latent semantic analysis (LSA) technique [9], which improves accuracy.

2) We constructed a new knowledge/concepts index module, which contains topics, concepts, and knowledge for each lab. It can be used to establish a relationship between any two labs and a searching tool for both instructors and students to explore labs and concepts available in the system.

3) By combining 1 and 2, we created a lab guidance tool in our hands-on lab environment [5], [6] to guide both course builders and students. This system can suggest lab contents, knowledge, and concepts to users by exploiting the knowledge from the knowledge graph and labs.

NeoCyberKG was then applied in an e-learning virtual lab environment [5], [6] used by college students for a case study. By using the system as a recommendation/guidance tool for students, the study proves that NeoCyberKG can meet students' expectations when making a recommendation. Users also tend to gain enhanced learning outcomes and express more interest in the cybersecurity area.

The rest of this paper is organized as follows. Section 2 describes the background of the paper and related work. Section III discusses the system architecture and the LSA model for NeoCyberKG, and explains how we utilize it as a learning guide. Section 4 reports a case study that using the concept map in a graduate-level cybersecurity course, and discusses and discusses students survey results. Finally, there is a discussion and conclusion of the paper in Sections 5.

2 BACKGROUND

2.1 Knowledge Graph Construction

Researchers construct various knowledge graphs (also known as knowledge bases) to organize knowledge. A typical knowledge graph is usually a multiple relational directed graph, recorded as a set of relational triples (h, r, t), which indicate relation r between two entities h and t. Knowledge graphs play an important role in many applications such as question answering and recommendation because of their rich structural information. In order to construct knowledge graphs, text mining and relation extraction are used as common approaches. The goal of these approaches is to extract relational facts from plain text. Kernel-based models [23] and embedding-based models [19] [7] have been introduced, but each has its disadvantage. For the kernel-based approach, the extracted features and human-designed kernels result in errors of the various modules accumulating downstream. Also, the manually constructed features may not capture all the relevant information that is required. Single-word embedding models have been successful at learning lexical information. However, they cannot capture the static meaning of longer phrases of text, preventing them from a deeper understanding of human language. Embedding-based models also perform poorly for semantic representations of small-size text corpora [1].

2.2 Latent Semantic Analysis

The Latent Semantic Analysis (LSA) [9] is one of the most important bag-of-words methods. It describes each word in a vector space, where each word is represented based on its contextual-usage to a document. LSA takes as input a training corpora formed by a collection of documents. A word by document co-occurrence matrix is constructed, which contains the distribution of occurrence of the different words and the documents. A mathematical transformation is usually applied to reduce the weight of uninformative high-frequency words in the words-documents matrix. Finally, a linear dimensionality reduction is implemented by a truncated Singular Value Decomposition (SVD) [15], which projects every word in a subspace with lower dimensions. The success of LSA in capturing the latent meaning of words comes from this low-dimensional mapping. LSA is widely used in the Natural Language Processing (NLP) domain.

3 SYSTEM DESIGN

In this chapter, we described how to construct NeoCyberKG. Due to limitations of embedding-based knowledge graph models discussed in Section 2.1, we adopt the LSA technique [9] to extract knowledge graph nodes and use the topic modeling to explore their dependency relationship from a data source, instead of using Word2Vec [3] in previous works. The process includes data pre-processing, topic modeling and graph construction (Section 3.1, Section 3.2 and Section 3.3). Then, we utilized the constructed knowledge graph as a tool to make hands-on lab recommendation (in Section 3.4).

3.1 Latent Semantic Analysis

The NLP tool of latent semantic analysis (LSA) is used to perform the data preprocessing on text data. Latent features are extracted from text data, usually in three steps:

(1) The lab descriptions are transformed into a corpora. Data pre-processing is to transfer lab description into a corpora. The techniques for text preprocessing include lower-casing all text data; stemming and lemmatization, which transfers words into their root forms (e.g., using 'connect' to replace the words 'connected', 'connects', using 'good' to replace the words 'better', 'best'); removing stop-words (e.g., is, a the, etc.); normalization (e.g., using 'iptables' to replace 'iptable', 'ip-tables', 'ip tables', etc.); removing noise (e.g., digits characters, special symbols, etc.)
(2) An NLP's technique of Term Frequency-Inverse Document Frequency (TF-IDF) [9] is used to assign each term a weight from 0 to 1 to indicate the importance of that term to the description as a whole. TF-IDF weights a term by calculating the product of TF and its IDF. The score of TF-IDF shows how relevant a term is throughout all documents in a corpora.

For example, terms that frequently show up in most documents are weighted with a low score. In contrast, terms that frequently show up in few documents are weighted a high score since they frequently appear in only a document that carries more relevant information representing this specific document.

(3) Latent features are identified using a truncated SVD algorithm [9]. The truncated SVD algorithm finds the most valuable information of the data matrix. It can reduce the TF-IDF matrix dimension by finding similar patterns between terms and documents and combining them into a latent feature vector with a value between -1 and 1.

Each latent feature is a topic represented by specific terms in a document. By following these steps, we obtain the latent features of lab materials and use such latent features as input to automatically identify which labs are highly correlated than others through similarity clustering.

3.2 Topic modeling

The topic model gives an insight into latent semantic topics in a collection of documents and has better predictive accuracy. The inferred topics are more meaningful than using statistics by providing a hierarchical generative probabilistic model. LSA uses vector representation to represent the text's semantic content. An LSA model replaces raw statistic counts in the document-term matrix with a term TF-IDF score. Then, map these high-dimensional count vectors to a lower-dimensional representation in a latent semantic space. Using LSA, the semantic relations between words and/or documents are represented in the semantic space.

In this study, we use the TF-IDF matrix generated by the LSA tool to calculate the value of each lab's input as vector representations. Our goal of using topic modeling is to represent features of each lab into a vector space, which help us to connect highly related labs in our knowledge graph. The input of LSA model is lab materials used in our university, most of which are from class lab repository created by instructors in our school, and also labs from SEED lab [8]. We used these lab materials to build an input dataset for our LSA model, and 130 latent features are extracted. Table 1 shows an example of the first 10 topics identified by latent features in this study. The table shows the difference among topics represented by concepts identified. For example, Topic 1 represents labs on attacking through ftp protocol with the concepts of 'ftp', 'file', 'firewall', etc, and Topic 6 represents labs that using Mininet to construct network topology with the concepts of 'mininet', 'switch', 'controller', 'topology', etc. In this way, different topics represent different labs. By computing the 0-1 values that a lab on each specific topic, we obtain a vectorized representation for a lab to show its value on each topic. Such vector representation captures the latent features of a lab for each topic. We then apply K-means clustering algorithm to group similar labs together. The clustering result is generated based on 130 latent features under comprehensive correlations among these 130 topics for vector representations. Each vector represents a lab's text data from 180 text inputs for 36 labs, which is identified as a dot in Figure 1. It is hard to show the clustering result under all latent features visually; Figure 1 shows an example of clusters identified in this study with latent feature value in Topic 1.

Table 1: The topics of the first ten latent features

Topics	Terms in topics
Topic 1	ftp, file, linux, directory, packet, attack, firewall
Topic 2	packet, attack, lab, ip, server, dns, report
Topic 3	attack, dataset, python, datum, csml, training, dns
Topic 4	dns, server, attack, attacker, domain, corn, web
Topic 5	vpn, packet, trn, tunnel, interface, datum, program
Topic 6	attack, secret, mininet, switch, controller, cache, topology
Topic 7	web, http, apache, elgg, site, request, ftp
Topic 8	student, lab, vpn, section, firwall, security, vm
Topic 9	vpn, secret, firewall, execution, array, cpu, cache
Topic 10	xterminal, ftp, connection, mitnick, attack, tcp, server

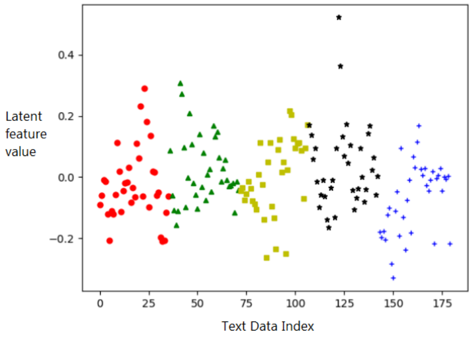

Figure 1: The 5 clusters of topic 1 identified by latent features

3.3 Knowledge Graph Generation

We define NeoCyberKG as $G = \{V, E\}$, where $V = \{v_i\}$, $E = \{e_{ij} : (s_{ij}, d_{ij})\}$, v_i represents a lab, an edge e_{ij} includes two measurements: similarity measurement s_{ij} and dependency measurement d_{ij}. For example, Figure 2 shows the graph for a single lab. In the graph, a statement node represents a single lab task, e.g., "*setup basic networking in Linux*", "*setup network application*", etc. A statement node can be mapped to one hands-on lab, and each lab is described by a procedure of tasks. And each statement node is connect to a set of concept nodes. Each concept node represents a concept that is required to solving the corresponding task, and its explanation contains knowledge.

We use the document-topic vector representation matrix generated in Section 3.2 to compute the similarity between the embedded vectors ($\vec{T^i}$ and $\vec{T^j}$) for labs (v_i and v_j). The semantic similarity ($s_{ij} = f(v_i, v_j)$) between labs is computed using the *cosine* of the angle between two vectors projected in an n-dimensional that corresponds to a topic $t_k^i, k = 1..n$ in the lab v_i:

$$s_{ij} = \frac{\vec{T^i} \cdot \vec{T^j}}{\left\|\vec{T^i}\right\| \left\|\vec{T^j}\right\|} = \frac{\sum_{k=1}^{n} t_k^i t_k^j}{\sqrt{\sum_{k=1}^{n} t_k^i}\sqrt{\sum_{k=1}^{n} t_k^j}}, \qquad (1)$$

where, $\vec{T^i} \cdot \vec{T^j} = \sum_{k=1}^{n} t_k^i t_k^j = t_1^i t_1^j + t_2^i t_2^j + ... + t_n^i t_n^j$ is the dot product of the two vectors that represent lab v_i and lab v_j with n topics, smaller the angle between labs, higher the similarity. We then construct NeoCyberKG by measuring this similarity among labs. If the cosine similarity between two laps is over a threshold, we connect their lab node in the knowledge graph.

A knowledge graph was ultimately built, which contained 372 concept nodes, 130 statement nodes and 36 lab objects. To compare the quality of our LSA model with the embedding-based approach used [7], we tested both methods on the same 180 text data from 36 labs used in Section 3.2. We then ran both models for 10 times with the same number of targeted topic number/embedding dimension value of 180, and calculate the Cohen's kappa coefficient [20] (range from 0 to 1, high is better) between the machine learning output and expert knowledge. The embedding-based approach achieves a kappa coefficient of 0.56 while the LSA's result is 0.71. Thus, LSA is able to achieve a more substantial agreement with expert knowledge and provide a solid improvement over the embedding-based approach.

3.4 Recommendation of Hands-on Labs

Our system utilizes the NeoCyberKG and lab materials to recommend labs for instructors and students based on their learning goals and expected learning outcomes.

To achieve that, We first created an entry-survey to check students' background in the cybersecurity domain. Then, each student selects either a set of concepts/knowledge they want to cover or a lab that they want to finish independently as their personal learning goal in a lab repository.

The NeoCyberKG system estimates the concept node coverage of a student based on his/her entry-survey results and updated these concepts as understood in their personal knowledge graph. We define the initial knowledge as initial concepts C_M and the student's learning goal as a targeting concept C_G. After that, NeoCyberKG can generate a set of paths P_{MG} between C_M and C_G using the knowledge graph. Each path P in P_{MG} contains a set of concepts C_p that the student needs to learn. By combining concepts from all the paths, we can list all path concepts the student needs to achieve his/her learning goal. The last step is to find a set of labs L that covers all concepts in C_P. Currently, our system will recommend labs that cover more identified concepts and are directly connected (in the knowledge graph) to each other. Thus, the output is a set of labs that share a lot of concepts between them. When students do such labs, they've got the chance to consolidate their current mastered concepts while learning some new concepts. The collection of these labs becomes our recommendation to a user. Each time the student finishes a new lab, we update initial concepts C_M and regenerate the recommendation to check if there is any update needed.

An example of the recommendation process for one user is shown in Figure 4. Based on the entry survey result, the user's initial knowledge coverage contains *Linux command line, set up Linux network*. It picks the learning goal of *setup SDN Firewall* only. Then AISecKG generated five recommended labs for him in sequence, as shown in the Figure 4 and listed in Table **??**. The five labs, in sequence, are: (1) Lab 1, *Linux network Lab*, which covers three statements (green boxes in figure) and demand basic computer network knowledge. (2) Lab 2, *MiniNet SDN sandbox lab*, which covers two problem statements (blue boxes in figure), this lab require the user to set up a MiniNet SDN environment, in which the user will set up firewall later. (3) Lab 3, *POX Controller Lab*, covers three problem statements (red boxes in figure) and covers how to set up POX as an SDN controller to forward traffic. (4) Lab 4, *Linux firewall lab*, which covers problem statements (yellow boxes in figure), this lab tests user's knowledge about network firewall and its usage. (5) Lab 5, *OpenFlow Based Stateless Firewall Lab*, which covers three problem statements (yellow squares in the figure), including the user's learning goal of setting up an SDN firewall. Notice that, only Lab 2, the Mininet lab, is optional, as other labs do not directly require it. But, since the Mininet lab gives users a better understanding of the SDN environment, both are still recommended.

4 CASE STUDY

An experiment using NeoCyberKG was conducted in a graduate-level network security class during Summer 2020 at Arizona State University. This class involves five hands-on labs for computer network security. Forty-three graduate students took the course, and thirty-four of them finished the post-course survey at the end of the semester.

During the semester, all forty-three student were required to first finish three labs in the virtual lab platform as part of their course evaluation. They were also asked whether they wanted to volunteer in this research practice, and thirty-eight students from the class participated. These thirty-eight students set their own learning goals on our knowledge graph and then got the labs' recommendation as an outcome of the NeoCyberKG system. They continued to work on these labs, and thirty-four of them finished all recommended labs. At the end of the semester, All these thirty-four students finished this post-course survey. Twenty-three of the students strongly agreed that this lab-based learning approach motivates them to learn computer science security. Further, thirty one students enjoyed this lab-based learning experience.

To construct this post-course survey, we follow the Instructional Materials Motivation Survey (IMMS) [11] to identify student motivation when doing this problem-based learning lab. IMMS is widely used in previous studies on education to evaluate students' motivation to work with technology [13] or a web-based course [4]. These survey questionnaires evaluate students' motivation from eight areas, including course overview, student's attention, the relevance of learning materials, the relevance of projects, student's confidence, student's satisfaction, and lab-based learning through role-playing and lab-based learning in general.

The following questions were asked in the post-course survey: (Answer on a scale of 1 to 5, 1 means totally disagree, while 5 is fully agree.)

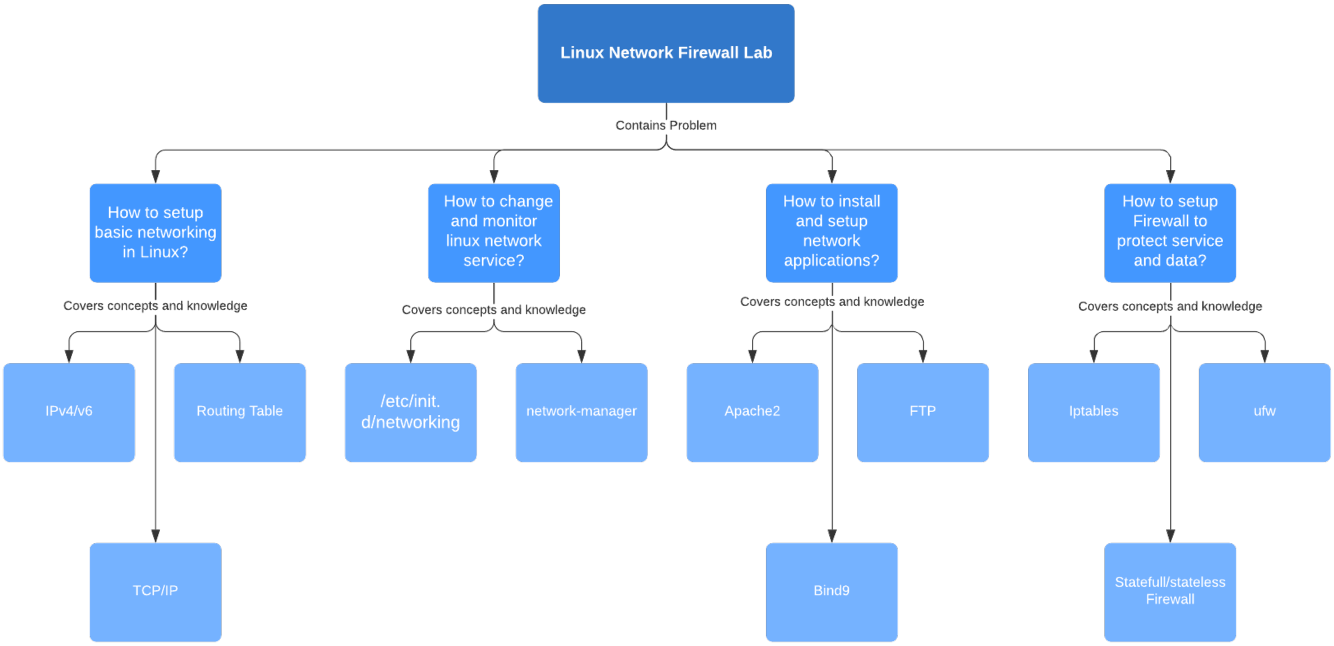
Linux Network Firewall Lab
Contains Problem
How to setup basic networking in Linux?
How to change and monitor linux network service?
How to install and setup network applications?
How to setup Firewall to protect service and data?
Covers concepts and knowledge
Covers concepts and knowledge
Covers concepts and knowledge
Covers concepts and knowledge
IPv4/v6
Routing Table
/etc/init.d/networking
network-manager
Apache2
FTP
Iptables
ufw
TCP/IP
Bind9
Statefull/stateless Firewall

Figure 2: Problems and Concepts Mapping for a Single Lab

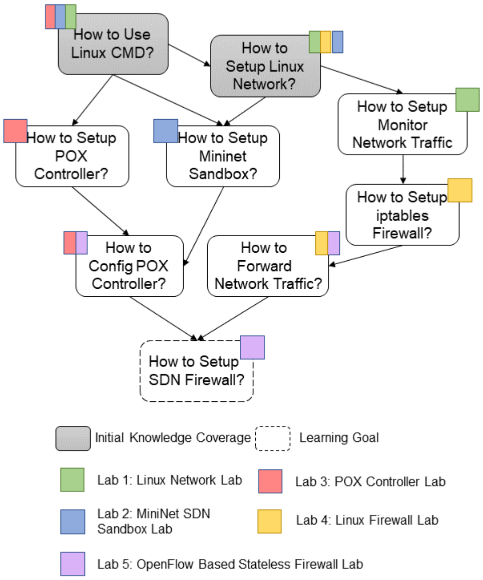
How to Use Linux CMD?
How to Setup Linux Network?
How to Setup Monitor Network Traffic
How to Setup POX Controller?
How to Setup Mininet Sandbox?
How to Setup iptables Firewall?
How to Config POX Controller?
How to Forward Network Traffic?
How to Setup SDN Firewall?
Initial Knowledge Coverage
Learning Goal
Lab 1: Linux Network Lab
Lab 3: POX Controller Lab
Lab 2: MiniNet SDN Sandbox Lab
Lab 4: Linux Firewall Lab
Lab 5: OpenFlow Based Stateless Firewall Lab

Figure 3: Lab Recommendation Process Example. Best viewed with color.

Course overview

Q1: Have you been (motivated to) learn computer science security with a lab-based learning approach?

Q2: Do you think that the lab-based learning approach has influenced your learning?

Q3: Do you consider the labs we did in this class close to the real world?

Q4: Do you consider these projects important for your own professional growth?

Attention

Q5: the lab material and lab platform helped to hold my attention.

Q6: The way the information is arranged in the lab instructions and lab platform helped keep my attention.

Q7: The variety of reading materials, exercises, illustrations, etc., helped keep my attention on the labs.

Relevance

Q8: It is clear to me how these lab materials' content is related to things I learn during class videos and slides.

Q9: The content in the labs is relevant to my interests and worth knowing.

Q10: The content of these labs will be useful to me in the future

Lab-relationships

Q11: Lab 1 is necessary for me, as it prepared me well for Lab 2,3, and 4.

Q12: it is clear that Lab 2 and Lab 3 are more related when compared to Lab 1 and 4.

Q13: The instructor should keep Lab 2 and 3 separate, proceeding in an orderly way and step by step, instead of merging Labs 2 and 3 together.

Q14. Lab 4 is closely related to other Projects.

Confidence

Q15. As I worked with these lab materials, I was confident that I could learn computer network security well.

Q16. After reading these lab instructions, I was confident that I would complete labs and the class well.

Q17. I could not really understand quite a bit of the material in the lab instructions. (Negative question)

Satisfaction

Q18. I enjoyed working with these labs so much that I was stimulated to learn more about network security and other related topics.

Q19. It felt good to accomplish lab tasks.

Q20. the feedback from the instructor helped me feel rewarded for my efforts in doing the labs.

Q21. Do you feel satisfied with the lab results delivered by you?

Role-based

Q22. Do you think including an "attacker" role in the lab-based learning approach would benefit you given a future real professional situation?

Q23. Do you consider the use of role-playing (attacker:defender:victim) important?

lab-based

Q24. Do you believe that using the Lab-based learning approach has helped you develop your learning skills?

Q25. Do you consider significant the time you have devoted to the project assignments?

Q26.Do you think that devoting the project's time to traditional lectures would be better? (Negative question)

Q27.Have you enjoyed the project experience?

Figure 4 shows each question's average score in this post-course survey on lab-based learning. Two questions (Q17 and Q26) are asked as negative questions, so we transfer the score into a positive score when counting the statistical results. This score shows that most students confirm that this lab-based learning positively impacts their learning attentions (average score = 4.0) and confidences (average score = 3.7). They are satisfied with this lab-based learning approach (average score = 4.2).

Specifically, we collect feedback from students to evaluate their perceived lab relationships in this case study. Figure 5 shows the average score of questions Q11 to Q14 in this area. Lab 1 is a background lab about *Linux networking and firewall setup*, Lab 3 is *SDN security labs*. Lab 2 is about SDN network, it is an recommendation generated by NeoCyberKG base on topics and concepts of and Lab 1 and 3. Q12 result shows that students strongly agrees that NeoCyberKG recommendation is highly related to Lab 3. Lab 4 is also picked by NeoCyberKG, not only base on topic from Lab 1 to 3, but also based on each student's personal learning preference this round. Q14 result shows students agree that Lab 4 topic is clearly distinguishable from other labs.

For lab-based learning, Figure 5 shows the average score of Q24-Q27. Students strongly believe that lab-based cybersecurity instruction enhances their learning skills and leads them to spend more time studying. They think this lab-based learning is better than traditional learning and have a good learning experience under this learning environment.

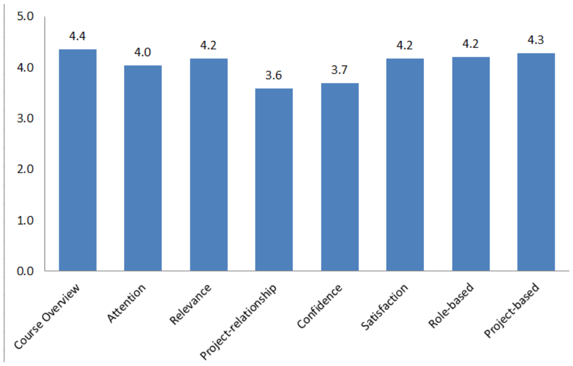

Figure 4: Average score of questions in each area in the post-course survey.

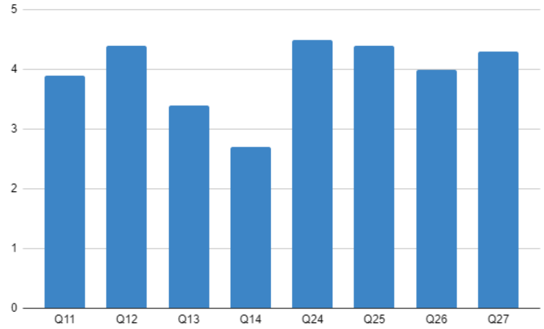

Figure 5: Average score of questions in the area of lab relationships and lab-based learning.

5 CONCLUSION AND FUTURE WORK

This paper describes our efforts towards creating a knowledge graph to represent concepts and their relationships in the cybersecurity domain. This work is intended to provide an organized knowledge graph that incorporates information from various data sources, including Wikipedia pages and instruction materials, including all relevant concepts within the domain for educational usage. We then applied such a knowledge graph into an e-learning virtual lab environment to test it. When using the knowledge graph as a recommendation/guidance tool for students, our case study proves that our prototype system can meet students' expectations when making the recommendation.

In future work, we want to incorporate more unstructured data into our system, including but not limited to textbooks, internet web pages, and online video transcripts. We plan to incorporate cybersecurity ontology, which is intended to support our knowledge graph generation. By adding ontology in NeoCyberKG, our knowledge graph will get the semantic definition, which is much more helpful than the similarity value we currently used. One limitation of current study is the limited size of students data, a lot of further experiments and in-class studies are necessary to verify and improve our research outcome.

Our ultimate goal is to build a knowledge graph that will serve as the backbone of the cybersecurity education domain, which would evolve and grow with additional cybersecurity lab sets as they become available and fully adaptive to different learners who want to utilize it.

ACKNOWLEDGMENTS

This research is supported by the National Science Foundation under grants DGE-1723440 and 2114789.

REFERENCES

[1] Edgar Altszyler, Mariano Sigman, Sidarta Ribeiro, and Diego Fernández Slezak. 2016. Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. *arXiv preprint arXiv:1610.01520* (2016).

[2] Diana Burley, Matt Bishop, Siddharth Kaza, David S Gibson, Elizabeth Hawthorne, and Scott Buck. 2017. ACM Joint Task Force on Cybersecurity Education. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 683–684.

[3] Kenneth Ward Church. 2017. Word2Vec. *Natural Language Engineering* 23, 1 (2017), 155–162.

[4] David A Cook, Thomas J Beckman, Kris G Thomas, and Warren G Thompson. 2009. Measuring motivational characteristics of courses: applying Keller's instructional materials motivation survey to a web-based course. *Academic Medicine* 84, 11 (2009), 1505–1509.

[5] Yuli Deng, Dijiang Huang, and Chun-Jen Chung. 2017. ThoTh Lab: A personalized learning framework for CS hands-on projects. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 706–706.

[6] Yuli Deng, Duo Lu, Chun-Jen Chung, Dijiang Huang, and Zhen Zeng. 2018. Personalized learning in a virtual hands-on lab platform for computer science education. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.

[7] Yuli Deng, Duo Lu, Dijiang Huang, Chun-Jen Chung, and Fanjie Lin. 2019. Knowledge graph based learning guidance for cybersecurity hands-on labs. In *Proceedings of the ACM Conference on Global Computing Education*. 194–200.

[8] Wenliang Du. 2011. SEED: hands-on lab exercises for computer security education. *IEEE Security & Privacy* 9, 5 (2011), 70–73.

[9] Susan T Dumais. 2004. Latent semantic analysis. *Annual review of information science and technology* 38, 1 (2004), 188–230.

[10] Barbara E Endicott-Popovsky and Viatcheslav M Popovsky. 2014. Application of pedagogical fundamentals for the holistic development of cybersecurity professionals. *ACM Inroads* 5, 1 (2014), 57–68.

[11] John M Keller. 1983. Motivational design of instruction. *Instructional design theories and models: An overview of their current status* 1, 1983 (1983), 383–434.

[12] Kenneth J Knapp, Christopher Maurer, and Miloslava Plachkinova. 2017. Maintaining a cybersecurity curriculum: Professional certifications as valuable guidance. *Journal of Information Systems Education* 28, 2 (2017), 101.

[13] Nicole Loorbach, Oscar Peters, Joyce Karreman, and Michaël Steehouder. 2015. Validation of the Instructional Materials Motivation Survey (IMMS) in a self-directed instructional setting aimed at working with technology. *British journal of educational technology* 46, 1 (2015), 204–218.

[14] William Newhouse, Stephanie Keith, Benjamin Scribner, and Greg Witte. 2017. National initiative for cybersecurity education (NICE) cybersecurity workforce framework. *NIST Special Publication* 800, 2017 (2017), 181.

[15] Christopher C Paige and Michael A Saunders. 1981. Towards a generalized singular value decomposition. *SIAM J. Numer. Anal.* 18, 3 (1981), 398–405.

[16] Rajendra K Raj and Allen Parrish. 2018. Toward standards in undergraduate cybersecurity education in 2018. *Computer* 51, 2 (2018), 72–75.

[17] Samuel Essa Said. 2018. *Pedagogical Best Practices in Higher Education National Centers of Academic Excellence/Cyber Defense Centers of Academic Excellence in Cyber Defense*. Ph.D. Dissertation. Union University.

[18] Daqian Shi, Ting Wang, Hao Xing, and Hao Xu. 2020. A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowledge-Based Systems* (2020), 105618.

[19] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 1201–1211.

[20] Susana M Vieira, Uzay Kaymak, and João MC Sousa. 2010. Cohen's kappa coefficient as a performance measure for feature selection. In *International Conference on Fuzzy Systems*. IEEE, 1–8.

[21] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[22] Belle Woodward, Thomas Imboden, and Nancy L Martin. 2013. An undergraduate information security program: More than a curriculum. *Journal of Information Systems Education* 24, 1 (2013), 63.

[23] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research* 3, Feb (2003), 1083–1106.