Predicting Consequences of Cyber-Attacks

Prerit Datta[1], Natalie Lodinger[2], Akbar Siami Namin[1], and Keith S. Jones[2]
[1]Department of Computer Science, [2]Department of Psychological Sciences
[1,2]Texas Tech University
{*prerit.datta, natalie.lodinger, akbar.namin, keith.s.jones*}*@ttu.edu*

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377825

***Abstract*—Cyber-physical systems posit a complex number of security challenges due to interconnection of heterogeneous devices having limited processing, communication, and power capabilities. Additionally, the conglomeration of both physical and cyber-space further makes it difficult to devise a single security plan spanning both these spaces. Cyber-security researchers are often overloaded with a variety of cyber-alerts on a daily basis many of which turn out to be false positives. In this paper, we use machine learning and natural language processing techniques to predict the consequences of cyberattacks. The idea is to enable security researchers to have tools at their disposal that makes it easier to communicate the attack consequences with various stakeholders who may have little to no cybersecurity expertise. Additionally, with the proposed approach researchers' cognitive load can be reduced by automatically predicting the consequences of attacks in case new attacks are discovered. We compare the performance through various machine learning models employing word vectors obtained using both tf-idf and Doc2Vec models. In our experiments, an accuracy of 60% was obtained using tf-idf features and 57% using Doc2Vec method for models based on LinearSVC model.**

I. Introduction

The National Institute of Standards and Technology (NIST) defines Cyber-Physical systems (CPS) as systems "comprising interactions between digital, analog, physical and human components" through physics and logic for enabling smart services and improving the quality of life [1]. The concept of Cyber-physical systems can be applied to a variety of areas such as manufacturing, healthcare, agriculture, aviation, business etc. Some of the popular CPS technologies include smart grids, smart cities, Internet-of-Things (IoT) and industrial control systems. Due to their combination of cyber as well as the physical domains, CPS systems brings about unique challenges from both domains. In addition to physical attacks, cyberattacks remain one of the critical challenges of cyber-physical systems security. With the recent increase in cyber incidents involving CPS [2], [3], reliability and availability of CPS systems remains a top security goal [4], [5].

Given the ever-evolving threat landscape, security researchers managing the security operation center (SOC) are often overloaded with numerous security incidents and, at the same time, trying to keep abreast with the latest threats in the wild. Situation awareness (SA) [6] is the concept of perceiving the elements in the environment, comprehending their meaning and making decisions or taking action. Cyber-situation awareness (cyber-SA) is the concept of situation awareness applied to the cyber-security domain. Cyber-SA can help security researchers to reduce their cognitive load and help them to focus on what is important for cyber threat analysis and mitigation.

Threat mitigation and analysis involves communication with different stakeholders who may not be well versed with the concepts of cybersecurity. One of the ways that cybersecurity personnel can communicate with stakeholders is through the realization of the (non)-technical consequences of attacks to end users and thus informing them about the impact of such attacks to them.

This paper is a first step towards reducing cognitive workload of security experts and even average Internet users. The research presented in this paper investigates whether it is possible to train a model that predicts the technical and non-technical layman consequences of novel cyber attacks. The trained model should be able to digest textual descriptions of new cyber attacks through vectorization and then map them to known cyber attacks with clear consequences to end users. Machine learning-based algorithms can perform document embedding and then vectorization. More specifically, the distance of embedded vectors and a threshold value can decide whether two cyber attacks have similar semantics and thus consequences.

We have created our own dataset of cyber attacks along with their technical and non-technical consequences. The repository consists of 93 diverse attacks and their descriptions, which are annotated with their consequences, all in textual and non-structured formats. We then apply natural language processing and machine-learning algorithms to cyber attacks descriptions to predict the consequences of the attack. This paper makes the following key contributions:

- Introduce a dataset of cyber attacks and their (non)-technical consequences to end users.
- Build machine learning models to predict the (non)-technical consequences of attacks.
- Report the performance of prediction models through experimental studies.

This paper is organized as follows. Section II describes situation awareness and its relationship to cyber-security for CPS. Section III discusses the related work. The technical background of the employed machine learning classifiers is presented in Section IV. We describe our methodology in Section V and results in Section VI. The conclusion and future work are described in Section VII.

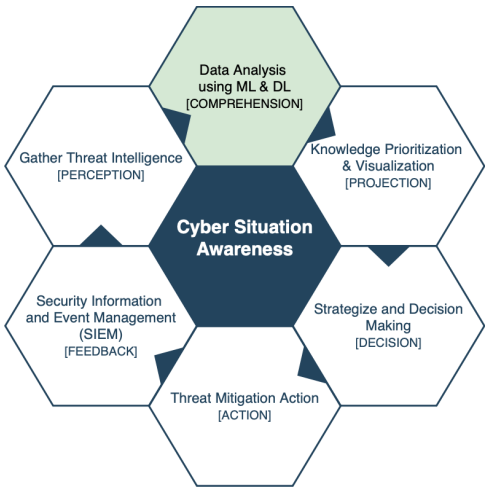

Fig. 1. Cyber-Situation awareness model based on Endsley's SA model.

II. Situation Awareness and Cyber-Security

In their seminal work, Endsley [6] described the concept of situation awareness as "the *perception* of the elements in the environment within a volume of time and space, the *comprehension* of their meaning and the *projection* of their status in the near future". The definition can be broken down into three components namely: perception, comprehension and projection. For Cyber-SA, *perception* involves gathering data from various threat sources, *comprehending* the disparate data by deriving patterns or knowledge and then taking actions based on the implications of the decisions made (*projection*). Given the voluminous amount of threat data and alerts, security practitioners often need automated tools to reduce their cognitive loads, prioritize alerts, do automated analysis, and generate reports for decision making. Additionally, cyber-situation tools can be used by defenders and security professionals to assess decision making and decide a recourse of action [7].

Figure 1 depicts the dimensions and a model for cyber-SA. The process begins by gathering the threat intelligence from various sources including the current state of sensors in the CPS system in addition to incident reports. After gathering the data, it is important that the SOC is powered with automated tools at their disposal that can use state-of-the art machine and deep learning algorithms to analyse the data. This is highlighted in green in the figure, as this is the main emphasis of the paper. After analysing the data, the SOC can prioritize the gathered knowledge and may visualize the threat reports to further strategize and make decisions after discussion with various stakeholders. After taking appropriate actions, the event logs and detailed action reports are stored in the security information and event management (SIEM) for future reference and decision-making.

III. Related Work

Cyber-physical systems present different security challenges compared to traditional IT systems for a variety of reasons. The microcosm of heterogeneous devices, which have limited processing and communication abilities, makes it difficult to offer encryption during communication [2]. Additionally, establishing trust to enable communication is another important challenge for securing CPS. Attacks on CPS systems, such as manufacturing and power grids, can incur significant damages and even loss of life [3]. An analysis of recent cyberattacks on CPS indicates that most of the attacks are driven to cause disruption, and are often politically motivated [3], [5].

There have been several efforts to incorporate cyber-situation awareness into CPS. Cyber-SA can be useful in simulating real cyber-incidents for decision making and training cyber defenders. Debatty and Mees [8] propose a tool assessing the cyber-situation awareness for cyber-defenders in the military. The tool consists of several scenarios that can be used to train and evaluate cyber defenders by simulating cyberattack scenarios to improve skills and knowledge of the cybersecurity personnel.

Preden [9] proposes a middle-ware architecture to enable situation awareness in cyber-physical systems. The key idea is to have a middle-ware that proactively handles tasks like service discovery, delivery and validation of data contracts to enable communication and resource constraint satisfaction. The authors argues that this aspect of decoupling data processing from communication will allow CPS to enable Cyber-SA.

Yang et al. [10] propose a framework to enable Cyber-SA in a smart metering system. The authors recommend coupling of both cyber and physical attacks data and then using a knowledge model for data analysis and decision-making. The knowledge model uses fuzzy logic to prioritize and assess attacks based on their severity, occurrence and detection using the underlying knowledge base and data from the previous step. This improves attack comprehension and helps security researchers automate the threat analysis process and make informed decisions.

Orojloo and Azgomi [11] propose a methodology to predict attackers' behavior and consequences of attacks on CPS. The authors use fuzzy logic to evaluate the possibilities of various attack scenarios with the help of an attack tree. The attacker's skills, knowledge and access are used to determine the likely attack paths with the help of expert knowledge. Attack impact is measured in terms of time-to-shutdown (TTSD) for each element in the CPS system and the attack's probability. The defenders can devise and assess the possible mitigation of various attack scenarios using this information.

Cardenas et al. [12] argue that, in addition to the challenges posed by the CPS, the research community has overlooked the consequences of the attacks on the CPS infrastructure. The authors contend that while security incident reports show some of the effects of cyberattacks, the actual consequence of a successful attack is the missing link in the CPS security literature. Our work presented in this paper is an attempt to bridge this gap and open up new research directions for the cybersecurity community to focus on the consequences of cyberattacks when deciding on the security of a system.

IV. Technical Background

A. Machine Learning Models

This section briefly discusses the technical background of the machine learning classifiers studied.

- **Linear Support Vector Classifier (SVC)** uses a linear kernel to perform classification. LinearSVC is preferred as it allows for additional parameters for tuning performance such as penalties and loss functions. Additionally, Linear SVC allows for a one-vs-rest classification strategy to be easily applied to multi-class problems.
- **Logistic Regression** is one of the most popular models in statistics and machine learning. Inherently, logistic regression is a predictive model that identifies the relationship between features (independent variables) and the target variable (dependent variable) in terms of probability.
- **Multinomial Naive Bayes (NB)** is a classifier that is best suited for discrete features. As a result, it is well suited for text-classification problems that use tf-idf features for text representation.
- **Random Forest** classifiers use an ensemble of decision trees wherein each individual tree predicts the class output. The class having the most votes is used as the final result. These trees are collectively called a forest. The word random in the decision represent that each tree picks only a subset of available features to determine the split and thus results in low correlation between different trees in the forest.
- **Multilayer Perceptron (MLP)** is the simplest form of artificial neural network. It consists of three layers: input layer, hidden layer and the output layer. The model tries to learn patterns in the data and then after being trained on the examples, it can classify the new instances. We used the MLP classifier from the scikit-learn library with maximum iterations of 1000 and default number of hidden layers.

B. tf-idf

One of the classical NLP-based algorithms called tf-idf is used to convert text into feature vectors that can be used by machine learning algorithms. tf-idf is an acronym formed by combining two words *term frequency (TF)* and *inverse-document frequency (IDF)*. The *term-frequency* gives a measure of how often a given word appears across documents/corpus whereas, the *inverse-term frequency* gives a measure of the importance of a word based on its rarity across documents. If a word appears in many documents frequently, it would have a high frequency and low rarity and thus, may not be really useful in the analysis. Term-frequency tf in a document d can be calculated by dividing the term's frequency by the total number of terms [13]:

$$tf = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Inverse-document frequency is calculated as $\log$ of total number of documents D divided by the total number of documents containing the term t [13]:

$$idf = \log \left[\frac{|D|}{\sum t \in d} \right]$$

The combined $tf - idf$ score is given by the product of tf and idf values for the term t:

$$tf\text{-}idf = tf_d * idf_{t \in D}$$

C. Doc2Vec

Document-to-vector (Doc2Vec), also known as *paragraph-to-vector*, was introduced in 2014 as an extension to a similar word embedding model - Word2Vec [14]. In addition to learning word vectors from the corpus, Doc2Vec introduces an additional vector called *paragraph vector* that represents unique features of each paragraph (such as the topic of the paragraph) in the corpus. Both the paragraph vectors and the context vector composed of word vectors are then used to predict the target word. This is similar to the Continuous-Bag-of-Words model (CBOW) in word2vec model [15]. The authors refer to this in their paper as *Paragraph Vector-Distributed Memory*. In this model, the paragraph vectors act as a memory by predicting the target word using the context words taken from the paragraph and the paragraph id. Another variant of Doc2Vec is distributed bag of words (PV-DBOW), which is similar to *skip-gram* of word2vec, wherein the target word is used to predict the surrounding context words.

V. Methodology

This section discusses the employed methodology along with data collection for building predictive models to capture consequences of novel cyber attacks.

A. Data Collection

As part of an ongoing research project, we collected 93 cyberattacks along with three descriptions explaining the attack compiled from publicly available security blogs and threat repositories such as: CWE [16], CAPEC [17] and ATT&CK [18]. We then annotated these attacks with 50 non-technical consequence descriptions, which are written in simple language (i.e., layman terms) so they can be understood by security practitioners as well as CPS end-users. This is important for communication because various stakeholders who make decisions about cyberattack mitigation and strategy plans have little to no security knowledge. The 50 consequences were then grouped into 7 clusters based on the similarity between the consequences. The process involved an open card sort activity through which we recruited a number of participants to sort and group the 50 consequences into groups. Table I shows some of the attacks, consequences and their cluster number. The cluster numbers and their labels are described in Table II. We only list the cluster labels and not the 50 individual consequences for brevity.

TABLE I
EXAMPLES OF THE ATTACKS, DESCRIPTION, CONSEQUENCES AND CLUSTER NUMBER.

Attack	Description	Consequence 1	Cluster#
Log Injection	An attacker can replicate log entries or inject malicious code by writing invalid user input. In most cases, this can be accomplished by entering certain characters. An attacker can also cause logs files to become unusable by corrupting the file format or inserting certain characters when rendered by systems that process these log files automatically. Additionally, attackers can also insert malicious code in the log files, which executes when the system parses the file. Corrupted log files can enable an attacker to hide their activities or can also make it look like some else performed the malicious activities [19].	The cyber-attacker modified your computer files in order to hide their activities.	4
Webpage/URL Spoofing	Web spoofing is a type of attack that enables an attacker to change and observe the web pages sent to a victim's browser. An attacker can also observe the information being entered by the victim into online-forms. This attack can happen even when there appears to be a secure connection to a website and can occur in any type of browser. The user is often oblivious to anything being out of place [20].	The cyber-attacker made you think an Internet site that the attacker created was a legitimate Internet site.	5
DNS Spoofing	Domain Name Server (DNS) Spoofing, also known as DNS poisoning, is an attack wherein the incoming network traffic to a legitimate website is directed to a malicious website by compromising the vulnerabilities in a DNS server fulfilling the request. DNS spoofing poses several challenges such as data theft and as a result, major banking and e-commerce website are often the target for attackers. Additionally, redirecting to the fake website may result in downloading malware and Trojans onto the victim's computers requesting the website. Simply cleaning the DNS cache alone is not a solution to this as the cache can get corrupted again. Flushing the DNS cache can be a potential solution to this problem [21].	The cyber-attacker rerouted your Internet requests to a device that they control .	6
DLL Tampering	Dynamic-linked libraries (DLL) are an important part of windows OS. It allows code reuse, modularization, and efficient memory utilization. Thus, DLL are crucial to help programs load and faster. An attacker can tamper with the DLL files and disrupt the normal functioning of the programs and system [22].	The cyber-attacker made your computer run software that your computer did not intend to run.	7
TCP SYN Flood	TCP protocol requires a three-way handshake to establish a connection. In TCP-SYN flood attacks, the attacker sends several new connection requests originating with different IP addresses. The server never receives an acknowledgment packet from the spoofed IP addresses and thousands of such connection requests cause the server to run of memory and it eventually crashes. This prevents the legitimate users to connect to the server as well [23].	The cyber-attacker caused your computer to crash.	7
UDP Flood	In UDP flood attack, a server is overloaded with UDP packets. UDP packets can be adjusted to a max size of 6500 bytes which can be used by attackers as a quick way to exhaust the server's memory and bandwidth using just a few compromised systems to flood the server [23]	The cyber-attacker caused your computer to run very slowly.	10

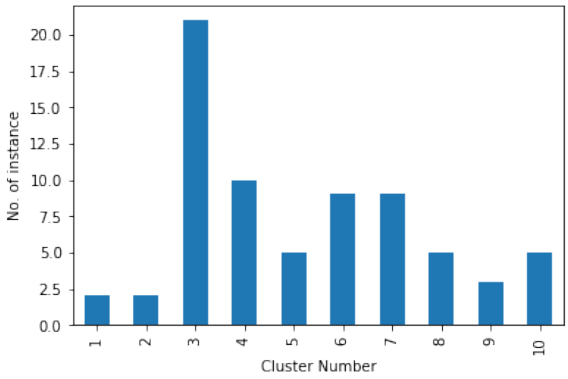

Fig. 2. Number of Instances in cluster labels.

TABLE II
LIST OF CLUSTER NUMBER AND THEIR CLUSTER LABELS

Cluster #	Cluster Label
Cluster 1	The attacker sent you emails that could lead to an attack if their request is granted
Cluster 2	The attacker disrupted your access to your computer or the Internet.
Cluster 3	The attacker gained access to your computer or one of your online accounts.
Cluster 4	The attacker altered your computer or its contents to allow them to use it for their purposes without you knowing.
Cluster 5	The attacker manipulated your use of or understanding about a website.
Cluster 6	The attacker changed or intercepted information that you have on the Internet.
Cluster 7	The attacker made your computer operate inefficiently or not at all.
Cluster 8	Cluster labels for 6, 5, and 3
Cluster 9	Cluster labels for 4, 7, and 2
Cluster 10	Cluster labels for 2, 7, and 7

B. Data Pre-processing

Some of the attacks had consequences that belong to more than one cluster. To accommodate this, we created

three additional cluster numbers and labels representing these combinations. For example, UDP flood attack in Table I had consequences belonging to clusters 2 and 7. We combined attacks having similar cluster numbers in any order (2,7,7 or 7,2,2) to be represented with cluster number 10. Thus, in addition to the 7 original cluster labels, we added 3 additional clusters labels representing a combination of consequences belonging to different clusters as shown in Table II. Thus, each attack ended up having a cluster number value between 1 to 10.

Fig. 2 shows the overall distribution of instances per each cluster number. It is clear that some of the clusters had fewer than 2 data instances. In order to avoid a class imbalance problem, we ended up ignoring clusters 1, 2 and 9 from the final dataset so as to avoid any bias during the training of the machine learning models.

C. Text Cleaning

The text data in the attack descriptions needs to be pre-processed before it can be useful for further processing. Text cleaning or pre-processing involves removing stopwords (such as articles), punctuation, digits, stemming and lemmatization. We used clean-text library[1] to convert the text into lower case and remove stopwords and punctuation. After this step, the text descriptions can be used for feature extraction.

D. Feature Extraction

We used `TfidfVectorizer` from sklearn library to generate the word-term matrix. We allowed for unigrams and bigrams to be included in the vocabulary and considered only words that have a document frequency of at least 2. Additionally, we used gensim's `Doc2Vec` library, to get feature vectors for attack descriptions. We used a vocabulary size of 300 based on preliminary experiments.

E. Experimental Setup

The final dataset consists of 72 cyberattacks after excluding physical attacks, such as using a USB Killer, as we wanted to focus only on attacks in the cyber-space. The dataset includes the attack's name, attack description, technical and non-technical consequences and the cluster number and label. The entire dataset was divided into training and test sets with a 70% and 30% split, respectively. Additionally, we used stratified samples based on the number of samples per class to account for the class imbalance.

We used both tf-idf and Doc2Vec features described in Section II to compare results across different machine learning models. For the tf-idf method, we included only words that have a minimum document frequency of 2 and max document frequency of 0.98. We also allowed for unigrams and bigrams to be included in the tf-idf matrix. For Doc2Vec, we used the distributed bag of words model with vocabulary size of 300 and learning rate 0.065. The Doc2Vec model was trained for 50 epochs. We used four machine learning models from the sklearn python library: *LinearSVC, Logistic Regression, Multinomial Naive Bayes, Random forest*, and *Multilayer Perceptron*. The results are described in the next section.

[1]https://pypi.org/project/clean-text/

VI. Results

Table III shows performance of the various models for both tf-idf and Doc2Vec methods. After text cleaning, each text description was converted to the appropriate format that the underlying library required. For example, in the case of tf-idf, the text descriptions were converted to a matrix with rows representing the documents and columns containing the words. Similarly, for Doc2Vec, the training and testing documents were tokenized and tagged with a label using the `TaggedDocument` class from Doc2Vec library. The features were then used to train the ML models to learn and predict the cluster number. This problem is an instance of the general multi-class classification wherein, the target has multiple class labels instead of binary labels.

TABLE III
Results of the classification models for predicting Cluster number.

tf-idf				
Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.60	0.45	0.60	0.50
Linear SVC	**0.60**	0.49	0.60	**0.53**
Multilayer Perceptron	0.60	0.51	0.60	0.50
MultinomialNB	0.40	0.28	0.40	0.26
RandomForestClassifier	0.40	0.16	0.40	0.23
Doc2Vec				
Logistic Regression	0.48	0.53	0.48	0.46
Linear SVC	**0.57**	0.57	0.57	**0.53**
Multilayer Perceptron	0.38	0.43	0.38	0.37
GaussianNB	0.32	0.23	0.32	0.25
RandomForestClassifier	0.42	0.37	0.42	0.35

1) tf-idf. In the case of the tf-idf method, the *Linear-SVC* method is selected as the best performing model over *Logistic Regression* and *Multilayer Perceptron* as it has a higher F1-score compared to the other two models. Surprisingly, the worst performing model is the random forest classifier. This is against the supporting literature in prediction and classification modeling in some other security related topics such as fake reviews identification [24] and zero-day malware detection [25]. A possible reason might be due to the diversity of cyber security descriptions and thus difficulty in building decision trees when conducting ensemble prediction models.

2) Doc2Vec. Similarly, for the Doc2Vec method, *LinearSVC* outperformed the other classifiers. *LinearSVC* achieved an accuracy score of 57% and an F1-score of 53%. While the performance of random forest classifier has not been improved, the performance of the other classifiers such as multilayer perceptron and Gaussian NB has been degraded. Here, we studied simple multilayer perceptron. It would be intriguing to investigate the performance of deeper versions of Convolutional [26] or Recurrent Neural Networks [27] and optimize the prediction models accordingly.

As Table III indicates, *Linear-SVC* model had the best performance for both the tf-idf and Doc2Vec methods with 0.6 and 0.57 accuracy, respectively. It is worth noting that

tf-idf and Doc2Vec are different in that tf-odf considers the frequency of the terms in the document (i.e., email); whereas, Doc2Vec focuses on the semantics of documents. Given that we obtain a slightly better result using tf-idf than Doc2Vec, it may indicate that for phishing detection, email embedding through extracting frequency features might provide a better classification. The results of predicting consequences of attacks are promising and it indicates that with some additional hypertuning and optimization it is possible to hit a higher accuracy.

VII. Conclusion and Future Work

In this paper, we employed machine learning models to predict the consequences of cyber attacks using two popular word embedding methods, that is, tf-idf and Doc2Vec. LinearSVC achieved the best performance for both the cases, which is consistent with past research that demonstrated LinearSVC to be well suited for multiclass-classification problems in natural language processing tasks. Although, the best accuracy obtained was only 60%, this could be attributed to the small data sample with an unequal number of samples per cluster label. We tried to accommodate for this problem by using stratified sampling during train-test splitting. A stratified approach balances the samples in the train-test split such that data instances for no single class can bias the ML models.

In future work, we would like to explore complex features using other word-embeddings, such as Word2Vec, to evaluate if the prediction scores can be improved. It would be also to reduce the dimensionality of the output of Doc2Vec using feature reductions techniques such as principal component analysis and encoder-decoder [28]. These emerging techniques have shown great performance in recent research work. Additionally, we would also like to explore whether we can use other natural language processing techniques to directly predict the actual consequence itself instead of the cluster number.

Acknowledgment

This research work is supported by National Science Foundation (NSF) under Grant No: 1564293.

References

[1] N. I. of Standard Technology. Cyber-physical systems. https://www.nist.gov/el/cyber-physical-systems.

[2] S. Walker-Roberts, M. Hammoudeh, O. Aldabbas, M. Aydin, and A. Dehghantanha, "Threats on the horizon: Understanding security threats in the era of cyber-physical systems," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 2643–2664, 2020.

[3] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, "Cyber-security incidents: A review cases in cyber-physical systems," *International Journal of Advanced Computer Science and Applications*, vol. 9, 2018.

[4] D. Serpanos, "The cyber-physical systems revolution," *Computer*, vol. 51, no. 3, pp. 70–73, 2018.

[5] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, "Cyber-physical systems security: Limitations, issues and future trends," *Microprocessors and Microsystems*, vol. 77, p. 103201, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0141933120303689

[6] M. R. Endsley, "Design and evaluation for situation awareness enhancement," *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, no. 2, pp. 97–101, 1988. [Online]. Available: https://doi.org/10.1177/154193128803200221

[7] R. S. Gutzwiller, S. M. Hunt, and D. S. Lange, "A task analysis toward characterizing cyber-cognitive situation awareness (ccsa) in cyber defense analysts," in *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2016, pp. 14–20.

[8] T. Debatty and W. Mees, "Building a cyber range for training cyberdefense situation awareness," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–6.

[9] J. Preden, "Generating situation awareness in cyber-physical systems: Creation and exchange of situational information," in *2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014, pp. 1–3.

[10] Z. Yang, T. Li, and W. Jiang, "Situation awareness for cyber-physical system: A case study of advanced metering infrastructure," in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2018, pp. 1–6.

[11] H. Orojloo and M. Abdollahi Azgomi, "Predicting the behavior of attackers and the consequences of attacks against cyber-physical systems," *Security and Communication Networks*, vol. 9, no. 18, pp. 6111–6136, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1761

[12] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on Future Directions in Cyber-physical Systems Security*. DHS, July 2009. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/601.html

[13] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to information retrieval*. Cambridge university press, 2008.

[14] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[16] MITRE. Cwe - common weakness enumeration. [Online]. Available: https://cwe.mitre.org

[17] ——. Capec - common attack pattern enumeration and classification. [Online]. Available: https://capec.mitre.org

[18] ——. Mitre att&ck. [Online]. Available: https://attack.mitre.org

[19] OWASP. Log injection software attack. [Online]. Available: https://owasp.org/www-community/attacks/Log_Injection

[20] P. University. Secure internet programming: Web spoofing. [Online]. Available: https://sip.cs.princeton.edu/WebSpoofing/

[21] Kaspersky. What is dns cache poisoning and dns spoofing? [Online]. Available: https://usa.kaspersky.com/resource-center/definitions/dns

[22] Microsoft. What is a dll. [Online]. Available: https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library

[23] S. Rao, "Denial of service attacks and mitigation techniques: Real time implementation with detailed analysis," SANS Institute, Tech. Rep., 2011.

[24] L. Gutierrez-Espinoza, F. Abri, A. S. Namin, K. S. Jones, and D. R. W. Sears, "Ensemble learning for detecting fake reviews," in *44th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2020, Madrid, Spain, July 13-17, 2020*, 2020, pp. 1320–1325.

[25] F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani, and A. S. Namin, "Can machine/deep learning classifiers detect zero-day malware with high accuracy?" in *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*, 2019, pp. 3252–3259.

[26] N. Tavakoli, "Seq2image: Sequence analysis using visualization and deep convolutional neural network," in *44th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2020, Madrid, Spain, July 13-17, 2020*, 2020, pp. 1332–1337.

[27] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and bilstm in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*, 2019, pp. 3285–3292.

[28] N. Tavakoli, S. Siami-Namini, M. A. Khanghah, F. M. Soltani, and A. S. Namin, "Clustering time series data through autoencoder-based deep learning models," *CoRR*, vol. abs/2004.07296, 2020.