Hindawi
Security and Communication Networks
Volume 2022, Article ID 6012232, 19 pages
https://doi.org/10.1155/2022/6012232

Research Article

Network Penetration Intrusion Prediction Based on Attention Seq2seq Model

Tianxiang Yu,[1,2] **Yang Xin**,[1,2] **Hongliang Zhu**,[1,2] **Qifeng Tang**,[3,4] **and Yuling Chen**[2]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guizhou 550025, China*
[3]*National Engineering Laboratory for Big Data Distribution and Exchange Technologies, Shanghai 200436, China*
[4]*Shanghai Data Exchange Corporation, Shanghai 200436, China*

Correspondence should be addressed to Yang Xin; yangxin@bupt.edu.cn

Received 9 October 2021; Revised 1 March 2022; Accepted 16 March 2022; Published 4 May 2022

Academic Editor: AnMin Fu

Intrusion detection is a critical component of network security. However, intrusion detection cannot play a very good role in the face of APT and 0 day. It needs to combine intrusion prevention, deception defense, and other technologies to ensure network security. Intrusion prediction is an important part of intrusion prevention and deception defense. Only by predicting the next possible attack can we prevent the corresponding intrusion or cheat adversary more efficiently. However, the current research on intrusion prediction has not received much attention. Most of the existing intrusion prediction research focuses on the prediction of security situation, specific security events, system calls, etc., having limitation in applicability and sequence dependency. In order to supplement this part of research, this paper reports the prediction of network penetration intrusion sequence for the first time. By introducing the ATT&CK framework, this paper builds a dictionary for the penetration intrusion types and builds three different seq2seq models. The experiment runs on the public and generated sequence data based on real APT events and adversary groups resulting that the model can predict future penetration intrusion sequence with an accuracy of up to 0.90.

1. Introduction

Nowadays, the threat of business network penetration, cyberattack, is one of the major cybersecurity challenges. These cyberattacks have specific targets and clear goals. The attackers who carry out these attacks are highly organized and well-resourced. They work in groups, using stealthy and multi-step attack techniques to achieve their goal [1]. These technologies are of heterogeneous nature for different tactic purposes, from spearphishing attachment [2] or supply chain compromise [3] for initial access to vulnerabilities exploit for execution [4], to Hijack or Token manipulation for persistence and privilege escalation [5], to covert channel for exfiltration [6]. Different intrusion detection systems (IDS) have been invented to detect and mitigate these attacks, such as malware detection [7], spearphishing detection, and covert channel detection. Now detection systems have become the basement of cybersecurity. However, due to the covert feature and 0-day vulnerability exploit of these penetration cyberattacks, detection systems cannot mitigate these threats completely. Thus, more recently, the computer security community has devoted more attention to intrusion prevention [8] and deception defense [9]. Intrusion prevention systems (IPS) protect valuable assets from cyberattacks by detecting anomalous behavior and taking mitigation actions with the help of IDS. However, if we add intrusion prediction modules into the IPS, the IPS can take some preventive actions before the attack really happens. For example, prevention systems can adjust the protection level of specific network assets if these systems have gathered anomalous behavior or predicted upcoming intrusion related to these assets. Deception defense systems attract attackers to focus on the exposed fake network and fake service so that the damage caused by the intrusion will not occur in the real business network. The information about attackers and attack tools gathered by the deception system will be of

great help to intrusion detection systems and other network security tools. With the help of intrusion prediction, deception defense systems can flexibly change and adjust its services and deception strategies based on the upcoming attacks.

As mentioned above, we have introduced the functionality and necessity of intrusion prediction. Recently, the community has also devoted attention to intrusion prediction and similar research to help intrusion prevention systems and deception defense systems work better. Approaches for addressing the intrusion prediction task include discrete model approaches, continuous model approaches, and machine learning approaches, as discussed by this paper [10]. Different approaches are used in different fields of intrusion prediction research, such as data mining and attack libraries building used in the analysis of attack patterns for attack projection [11, 12], adaptive grey model used in network security situation prediction [13], and deep learning model used in security events prediction [14], system-call prediction [15], web attacks prediction [16], and alert information prediction [17]. Although there have been many kinds of intrusion predictions, most intrusion prediction models will encounter tremendous obstacles in practical application. Many intrusion prediction models only predicted intrusion events that happened in a single terminal computer or server [14, 16, 17], ignoring the relevance among intrusion events that happened in different nodes in the real network, which brings uncertainty to the real business network application of these models. Although, due to the great performance of deep learning, these models actually predict specific security events or alerts that will happen later, there is still inconvenience in real applications because the effectiveness of security events will change rapidly with the release and spread of patches, which results in the need to update the security events library and retrain the model frequently.

To address the problems mentioned above, we propose a deep learning-based sequence-to-sequence (seq2seq) model that predicts network penetration intrusion sequences from the intrusions that happened already. We use the ATT&CK enterprise matrix [18] to map the security events and intrusion alerts in the same network to penetration intrusion technologies for different tactics. The model predicts upcoming penetration intrusion techniques such that the model only needs to care about the mapping of the ATT&CK framework and the patterns of network penetration intrusion, which brings convenience to its application. After obtaining the information of upcoming penetration intrusion technologies for specific tactics, the administrators or different kinds of network security monitoring systems can take actions easily, e.g., the administrator or intrusion prevention system can adjust the protection level of specific assets and network access control level after prediction to avoid further asset damage. Deception defense systems can adjust the service, information, and structure of the business network and the honeynet to attract attackers continuously.

The data of intrusion prediction are hard to obtain; many models of papers run on data collected from security corporations, and hence it's almost impossible to make these data public. To address this problem, we use public data from the field of incident response research, provided by Yusuke Takahashi [19]. This paper generates targeted attack sequences data based on reports of eight actual security incidents. However, the amount of these data is not enough to support the training of the model; thus, we investigate the groups listed in the ATT&CK page [20], collect the techniques used by these groups, and use the data generation tools provided by Takahashi [19] to generate the attack sequence data for meeting the requirements of model training. Finally, we run three different seq2seq models on these attack sequences data and show that our approach is effective in predicting the upcoming penetration intrusion sequence with an accuracy of up to 0.90.

The contribution of this paper is listed below.

(i) By introducing and building the seq2seq prediction model, we extend the intrusion sequence prediction problem from predicting next one element to predicting one sequence, which is more suitable for the penetration intrusion prediction scenario.

(ii) We firstly run the intrusion prediction model on penetration attack sequence data to prove its effectiveness in learning features and completing prediction tasks in penetration intrusion prediction scenarios. We also discuss how to deploy the model in practical applications.

(iii) We propose some designs and mechanisms about the usage scenario of the intrusion prediction.

Although the model we built is just a simple seq2seq model and the data are generated by a greedy algorithm and manually fixed, we still want to publish our findings, hoping this can inspire more researchers to divert their attention to intrusion prediction and deception defense.

2. Background and Motivation

In this section, we will propose the necessary background information to better explain our model, its application scenario, and the motivation behind it.

2.1. Prediction and Forecasting in Cybersecurity. There are several use cases of prediction and forecasting in cybersecurity, like attack projection, attack intention recognition, intrusion prediction, and network security situation forecasting, as discussed in this survey [10].

Attack projection focuses on recognizing a specific adversary's action pattern and predicting the adversary's next action category, as discussed in [21]. The approaches in attack projection include attack graph building, attack plan, and attacker estimating. These methods need to created prior by experts. Attack intention recognition's final task is to figure out the ultimate goal of an adversary [22]. The approaches for handling the recognition problem include causal networks, path analysis, graphical modeling, and Dynamic Bayesian Network (DBN). Network security situation forecasting is proposed to predict how the overall network security situation will evolve [23] and the system-

level security situation will evolve, e.g., whether the website will become malicious [24] and the machine will become infected [25].

Intrusion prediction needs to predict what type of attack will occur in a network or a single server [26]. Due to the motivation we will talk later, we choose intrusion prediction as our use case in the prediction of cybersecurity. As we have chosen intrusion prediction as the solution to our problem, we will talk about the methodology and model in detail about intrusion prediction and explain why we choose the seq2seq model as our final model.

There are several methods that will be used in intrusion prediction, such as alerts correlation, sequence of action, statistic methods, probability methods, and feature extraction. Alerts correlation interprets multiple alarms to semantic information content associated with the message to identify the relationship among alerts [27]. Sequence of action records the information in the ordered set of many kinds of security incidents, such as system call sequences [28] and network packet sequences [29]. Statistic methods investigate the collected data to reveal the underlying patterns and trends.

Although statistic methods might not be suitable for cybersecurity problems because of their linear analysis nature, there are still many statistic algorithms that solve these problems to suit different solutions, like linear regression, weighted moving average, and exponential smoothing [30, 31]. Probability methods represent the probability distributions of cybersecurity parameters, with the hidden Markov model (HMM) and Bayesian network being the most representative models [32, 33].

Our application scenario is network penetration intrusion prediction. Due to the adversary groups taking multiple steps to attack the network, network penetration intrusion can be considered as a sequence. And considering the previous comparison studies' results in these papers [14, 17] that deep learning-based statistic methods are much better than the probabilistic methods, we use statistic methods to learn the features in sequence data of penetration intrusion and build the seq2seq model to solve this penetration intrusion prediction problem.

2.2. Deception Defense. The need for deception defense systems is the main motivator for us to perform the network penetration intrusion prediction research; so, we will talk about the deception defense in this section.

As defined in this paper [34], cybersecurity deception misleads and confuses attackers to thereby cause them to take (or not to take) specific actions that aid cybersecurity. The main deception tools are honey-based tools, including honeypot, honeynet [35], honeytoken [36], and honeywords. There are other deception tools in network and physical layers [37, 38], like fingerprint hopping and address space randomization.

In this section, we will mainly talk about honeynet because there are many limitations of isolated use of deception tools, like the lack of interaction with the adversary and insecurity brought by high-interaction honeypot. The honeynet can use other deception tools and cooperate with other cybersecurity systems in a network. To continuously deceive adversaries that they are attacking the real system and there are some cybersecurity leakages in the system, the honeynet needs to adjust its node components, information components, and services configurations according to some deception strategies such that the adversaries will reside in the honeynet for a long time and thereby help collect information about the adversaries and the tools they use more easily.

One of the most important deception strategies is deception interaction based on gathered information, as discussed in the honeypot deception strategy survey [39]. However, the research of the deception interaction strategy between an adversary and a defender is still at an early stage and focuses on very simple scenarios, as discussed in this survey [40]. Many game theoretical models of interaction between the attackers and the system are based on only a few environment parameters and system actions.

We conclude the reason of game theory model being impractical is the lack of interaction-related information. This way the model cannot choose so much parameters and system actions because these parameters are just raw data which are not easy for the model to handle.

We try to solve the problem by introducing penetration intrusion prediction information to the deception strategy model. The penetration intrusion prediction information is a highly abstract information of interaction between adversaries and network defenders. Deception defense systems can use the intrusion prediction information and other cybersecurity information as a summary of the network situation and adversary status.

By introducing the prediction information and using the docker-based honeypot [41], we believe more deception strategies and actions can be proposed to keep the adversaries believing that they are in the real network and there is some leakage in the network. That's the main motivation for us to perform the network penetration intrusion prediction. We can use firewall and routers to redirect anomaly access to a honeynet server whose computing nodes and open services are all run by docker containers. The honeynet server collects adversary information and takes deceptive actions to let the adversary believe this is the real business network, such as create honey tokens, create docker-based honeypot such that the highly available interaction in the honeypot can increase the reality of the network, change the network status after one honeypot is attacked using docker control tools like Kubernetes, and create a new honeynet if adversaries want to move through subnets. The design of penetration intrusion prediction used in deception defense is shown in Figure 1.

2.3. Intrusion Prevention. The need for intrusion prevention systems (IPSs) is another motivation for us to dwell on the penetration intrusion prediction research; we will talk about the intrusion prevention system and the application of intrusion prediction in this section. The intrusion prevention system just has one more intrusion prevention engine module than the intrusion detection system (IDS), based on

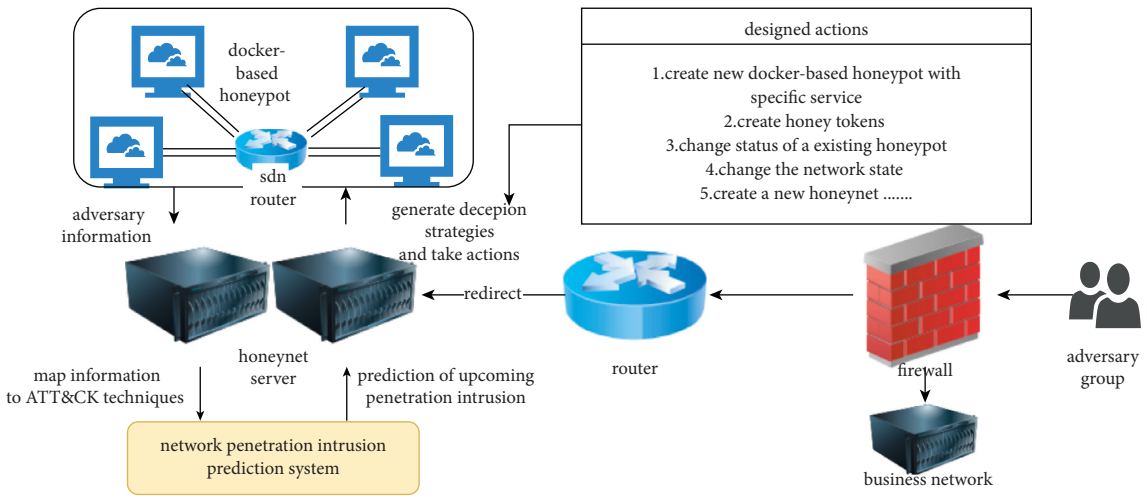

Figure 1: The design of penetration intrusion prediction used in deception defense.

this survey paper [42]. IPSs can be seen as an advanced combination of IDS, firewall, anti-virus software, and other cybersecurity control tools. IPSs still need to rely on the result of IDS, just preparing some actions for security devices to take to keep the damage minimal. However, IPSs cannot take actions before initial damage happens; it also cannot play a good role in the face of APT and 0-day attack due to its strong dependency on intrusion detection engines. By adding the intrusion prediction module to the IPS, the IPS can take some preventive actions to avoid future damage, such as credential access protection, application isolation and sandboxing, boot integrity lock, limitations on installation, adding authentication, data backup, and limitations on access to resource. Even if the damage cannot be avoided, the IPS with the intrusion prediction module still can confuse and perplex adversaries, giving defenders more time to react. That's also a motivation for us to perform the network penetration intrusion prediction research. We can add an intrusion prediction module between the intrusion detection engine and the prevention engine. After obtaining the result of the intrusion prediction module, the prevention engine can take some preventive actions as mentioned above and can work normally to mitigate the existing damage caused by the attacks and avoid further attacks. The design of the IPS with intrusion prediction is shown in Figure 2.

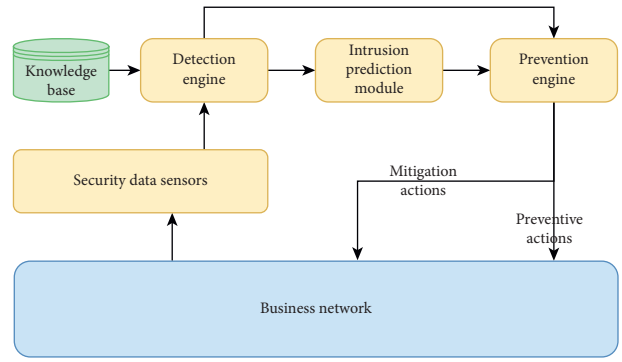

Figure 2: The design of penetration intrusion prediction used in IPSs.

2.4. ATT&CK Framework. As mentioned above, our specific intrusion prediction is network penetration intrusion prediction. However, the types of penetration intrusions are hard to define because the vulnerabilities and patches about cybersecurity change rapidly. Thus, we use the Mitre ATT&CK framework as our penetration intrusion knowledge base. The ATT&CK framework categories classify the adversary tactics and techniques based on real-world observation. By defining the types of penetration intrusion and building the cybersecurity threats dictionary, we can predict unobserved intrusion based on observed ones due to the associations in these tactics and techniques, as discussed in [43]. After introducing the ATT&CK framework, the rest of the job is extracting these associations' information in the observed sequence using the sequence model. There are many math models capable of handling sequence data, including the autoregressive integrated moving average (ARIMA) model, the Markov Chain model, and the seq2seq model. We already know that the penetration intrusion sequence will vary rapidly; thus, the ARIMA model is not suitable for this application scenario. As a shallow learning model, the Markov Chain model performs worse that the deep learning model in some research [17]. As mentioned, if we can predict a sequence rather than a single output, it will be helpful for other corporation systems to react more precisely; hence, the Markov Chain model may be not suitable for this application scenario due to the lack of future environment information. The seq2seq model is an encoder–decoder model and has been proven effective in many fields like machine translation and text summarization, which is similar to our application scenario, i.e., building dictionary for elements and learning the relationship information in sequences.

3. Materials and Methods

In this section, we will first introduce the methodology such that the readers can easily understand the problems we want to solve, why we need the seq2seq model, what kinds of data

the seq2seq model needs, and how data are processed through the model, which will greatly help understand the data material and the whole paper.

3.1. Methodology. In the methodology section, we will first introduce the application scenario of our model; then, we will introduce the ATT&CK framework which categories the specific intrusion techniques, and finally we will introduce our seq2seq model and its three different components.

The application scenario of our model is business network penetration intrusion prediction, which is predicting the upcoming network penetration intrusion based on the penetration intrusion that happened before. To make our model more practical, we use the seq2seq model to solve this prediction problem. The seq2seq model, first introduced by Cho et al. [44] and Sutskever et al. [45], predicts the future sequence based on the sequence that occurred already, which is also a categorical model considering sequence dependency and future prediction capability. Knowledge about business network penetration intrusion is complicated because the hardware and software environment and existing vulnerabilities differ in different networks and change rapidly through time. To satisfy the requirements of the scenario and the model, we have to use some methods to map these nearly countless network penetration intrusion types to some specific categories such that these seq2seq models can be trained with the data collected in different business networks and applied to more scenarios. To solve this mapping problem, we will introduce the ATT&CK framework in the next paragraph.

The ATT&CK framework is proposed by the MITRE Corporation [18], which is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. We can map network intrusion types to specific adversary techniques for different tactics using this knowledge base, as discussed by Aghaei and Al-Shaer [46]. This way, we can map cybersecurity incidents' reports and raw cybersecurity data collected in the network to ATT&CK techniques and tactics. Now that we have information of the ATT&CK techniques and tactics used in cybersecurity incidents by adversaries, we can just sort these incidents, techniques, and tactics by the occurrence time and get the network penetration intrusion technique sequence that happened in a specific incident. We can also list these techniques and tactics in a table and generate sequence intrusion sequence data from the table using the data generation tool proposed by Takahashi et al. [19]. Detailed information about this data will be explained in the data material section. Till now, we have introduced the application scenario of our model and the ATT&CK framework which helps generate the training data of our model. In the following paragraphs, we will introduce our seq2seq model and its components.

The seq2seq model is an encoder–decoder machine learning framework that predicts future sequences based on sequences that have already happened, such as word sequence, image sequence, and voice sequence. The seq2seq model has many implementations and components, such as the RNN (Recurrent Neural Network) seq2seq, the LSTM (Long Short-Term Memory) seq2seq, and the GRU (Gate Recurrent Unit) seq2seq. In the seq2seq model, the encoder reads the input sequence and produces a feature representation in continuous space, while the decoder generates a sequence based on the representation produced by the encoder. The basic architecture of our seq2seq model is depicted in Figure 3.

The network penetration intrusion sequence data can be collected from cybersecurity incident reports or generated from information about adversary groups and intrusion techniques they used. We use $s_i = \{t_1^{(i)}, t_2^{(i)}, \ldots, t_n^{(i)}\}$ to represent the network penetration intrusion sequence, while $t_n^{(i)}$ represents the specific intrusion technique that is used by the adversary at timestamp n during the whole network penetration. We can collect or generate these sequences that have different penetration purposes in different conditions from cybersecurity incident reports and adversary groups' information. We use $D = \{s_1, s_2, \ldots, s_m\}$ to represent these sequences' dataset. Of course, we will build a training dataset D_T and a validation dataset D_V from D, where $D_V \cap D_T = \varnothing$.

Then, we will explain how the sequence data are processed in our model. First, we need to specify the length of the input and output sequences, which has a big impact on the performance of the model. After that, we get one input data $x = \{t_{x1}, t_{x2}, \ldots, t_{xn}\}$ and output data $y = \{t_{y1}, t_{y2}, \ldots, t_{yn}\}$. We use h_t and s_t to represent the hidden state of the encoder and the decoder and use Unit_{enc} and Unit_{dec} to represent the processing unit of the encoder and the decoder, such as LSTM_{enc}, RNN_{enc}, and GRU_{enc}. Now, we have the input and output process formulas listed as follows:

$$\begin{aligned} h_t &= \text{Unit}_{\text{enc}}(x_t, h_{t-1}), \\ s_t &= \text{Unit}_{\text{dec}}(y_{t-1}, s_{t-1}). \end{aligned} \tag{1}$$

Considering the high relevance between intrusion techniques, which is caused by adversaries using multi-step attack techniques to achieve penetration, we decided to add the attention mechanism to our model. The attention mechanism introduces the score function to compute the score between the encoder hidden states and decoder hidden states for capturing location information. Then, we can use the score to compute the weight of the encoder hidden states and the weighted average of the encoder hidden states. These formula are listed as follows:

$$\begin{aligned} e_{ij} &= \text{score}\left(s_{i-1}, h_j\right), \\ \alpha_{ij} &= \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{n_x} \exp\left(e_{ik}\right)}, \\ c_i &= \sum_{j=1}^{n_x} \alpha_{ij} h_j. \end{aligned} \tag{2}$$

Now, we get the weighted average of the encoder hidden states for specific decoder hidden states and the decoder hidden state. We can concatenate them and obtain the

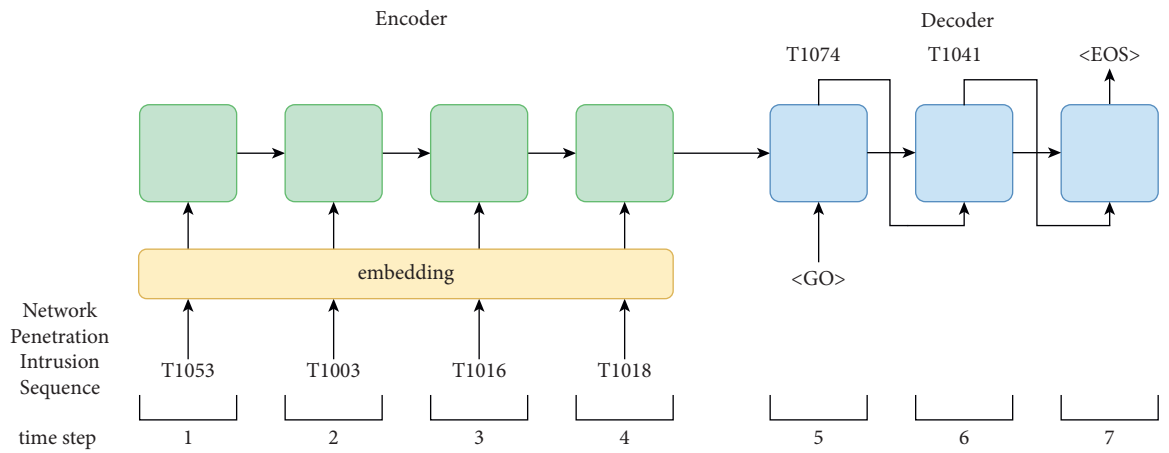

Figure 3: The basic architecture of our seq2seq model. "TXXXX" represents the network penetration intrusion techniques that are listed through time and form a network penetration intrusion sequence.

attention hidden state, which can be used to compute the final output at specific timestamp. These formula are listed as follows:

$$\widehat{s}_t = \tanh\left(W_c\left[c_t; s_t\right]\right). \tag{3}$$

After we get the attention hidden state, we can input the attention hidden state to the dense layer which uses softmax as its activation function.

$$y_t = \text{softmax}\left(\widehat{s}_t\right). \tag{4}$$

The attention mechanism of our seq2seq model is depicted in Figure 4.

We have introduced the seq2seq model and its attention mechanism. To implement the seq2seq model, we can choose different components. In this paper, we use the RNN, LSTM, and GRU to implement the seq2seq model and test its performance. We will talk about the details of these three components.

Simple RNN uses a matrix W to connect its hidden state to itself to implement the function of recurrence and relationship information memory.

$$h_t = \text{activation}\left(x_t \cdot U + h_{t-1} \cdot W\right). \tag{5}$$

Its structure is shown in Figure 5.

LSTM and GRU are the most popular variants of recurrent neural network models for sequential prediction scenarios. Instead of using a simple matrix to perform the sequence information extraction task, they design different mechanisms to do the memory task. As for LSTM, its hidden state computing formula are as follows:

$$h_t = \left[h_t', c_t'\right]. \tag{6}$$

The h_t' represents the short memory of sequence information and c_t' represents the long memory of sequence information. LSTM uses the gate function to protect and control the state of the memory. The structure of LSTM is shown in Figure 6.

The hidden state computing formula of LSTM are as follows:

$$\begin{aligned}
f_t &= \sigma\left(W_f \cdot \left[h_{t-1}', x_t\right] + b_f\right),\\
i_t &= \sigma\left(W_i \cdot \left[h_{t-1}', x_t\right] + b_i\right),\\
c_t &= \tanh\left(W_c \cdot \left[h_{t-1}', x_t\right] + b_c\right),\\
o_t &= \sigma\left(W_o \cdot \left[h_{t-1}', x_t\right] + b_o\right),\\
c_t' &= f_t * c_{t-1}' + i_t * c_t,\\
h_t^{'} &= o_t * \tanh\left(c_t'\right).
\end{aligned} \tag{7}$$

Different from LSTM using long-memory variant and short-memory variant as the hidden state, the GRU component uses one variant as the hidden state, designing the reset gate and update gate to control the state of memory. The structure of GRU is shown in Figure 7.

The hidden state computing formula of the GRU component are as follows:

$$\begin{aligned}
r_t &= \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right] + b_r\right),\\
z_t &= \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right] + b_z\right),\\
h_t' &= \tanh\left(W_h \cdot \left[x_t, h_{t-1} * r_t\right]\right),\\
h_t &= \left(1 - z_t\right) * h_{t-1} + z * h_t'.
\end{aligned} \tag{8}$$

3.2. Data Material. Data collection is hard. Many similar intrusion prediction research come from network security companies. They define and collect some specific security events and perform prediction on those data. Due to the confidentiality and severity after publishing the dataset, almost no dataset was published in previous research. We also want to use standard attack techniques (like "T1002") defined by the public knowledge base instead of using specific security events, which makes collection harder. Fortunately, after reading papers about security incident response, we got part of our data and data generation tool from Takahashi et al. [19]. They investigate eight actual cybersecurity incident reports and collect techniques and tools adversary groups use in these incidents. They generate 800 network penetration scenarios and corresponding attack sequences based on this information. In order to supplement the

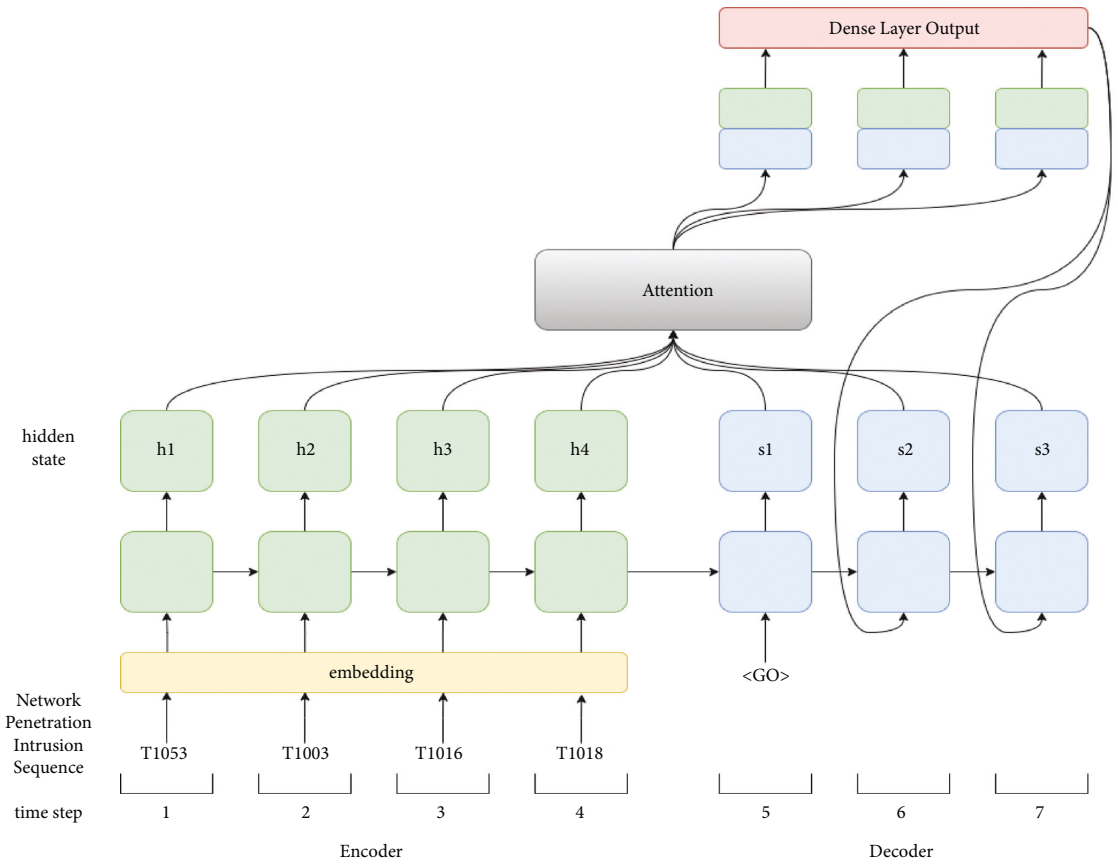

Figure 4: The attention mechanism of our seq2seq model.

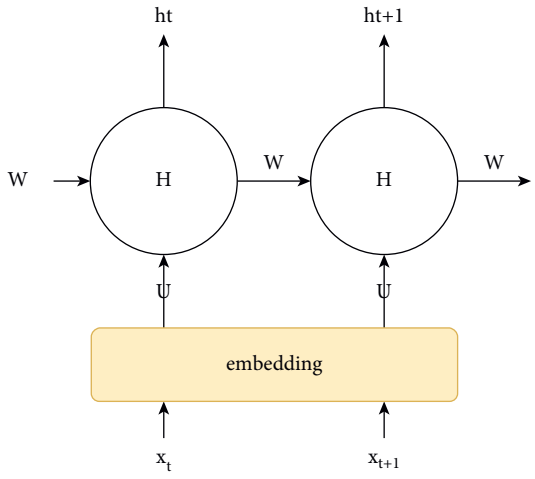

Figure 5: The structure of a simple RNN component.

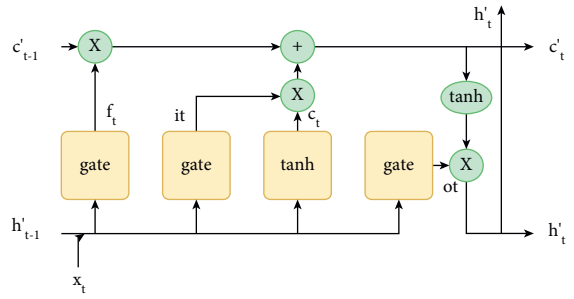

Figure 6: The structure of LSTM.

dataset, we also investigate several adversary groups about the attack techniques and tools they use. These adversary groups include APT32, Dragonfly, Leviathan, Patchwork, and other groups. Detailed information about these adversary groups and attack techniques can be found on this web page [20]. After investigating the adversary groups and cybersecurity incidents, we get their included attack techniques and estimate the number of penetration scenarios according to the complexity of the techniques.

After that we still need to perform some supplements and corrections to these attack technique lists such that the data generation tool can use these lists to generate complete penetration scenarios. These supplements and corrections improve the techniques lists' dataset from multiple perspectives, such as checking missing tactics in these lists and choosing appropriate techniques that achieve these penetration tactics to supplement the lists, checking inappropriate techniques for data generation tool and then adding them to the public excluded list, and investigating some classic penetration intrusion techniques and then adding them to the public included list. On the other hand, we do these supplements and correlations for adding mutual information to these datasets of different groups due to the lack of data amount and necessary mutual information to

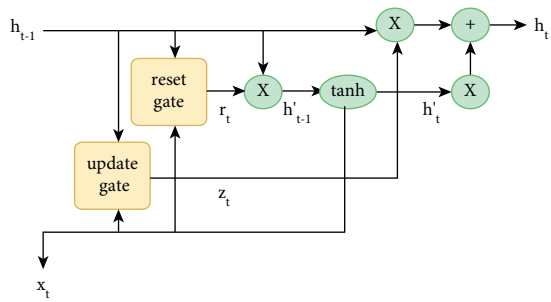

Figure 7: The structure of GRU.

perform the training and prediction task. Mutual information can be removed after enough investigation is done. Researchers and corporations can also edit their own supplementary techniques and public included techniques' list to make the prediction model appropriate for their own business applications. In this paper, we will only list the core ATT&CK attack techniques of groups and incidents because of the data usage in our business commercial system.

The overall information about Yusuke's dataset and our self-made dataset is depicted in Tables 1 and 2. The information includes the name of the investigated group or cybersecurity incident, the number of generated penetration scenarios, the core ATT&CK attack techniques in each group or incident, and the data source.

After preparing the techniques' lists, we still need to configure the data generation tool. The tool needs an atomic red team repository [47] to obtain information about the tools, tactics, supported platforms, input arguments, and executors related to one specific attack technique.

As the tool requires such information, we have to confirm that this tool works with ATT&CK version 3. Thus, we need to use this repository in a state before being converted to Mitre ATT&CK sub-technique schema.

After we reset the version of the atomic red team repository, we still need to add corresponding tactics to the software definition file of specific technique, about 200 files which contain about 1000 technique applications that need to be defined. The instruction of adding tactic definition and preparing the red team repository can be found in the files of the data generation tool.

We also need to edit the tool's configurations on environment as the tool requires. We use the network environment learning tool to gather the information of active entities and services in a network. The environment information gathering can be completed in three steps: environment data collection, environment data processing, and environment data transformation. We input the network segment information to the environment learning tool. Then, the environment data collection module will probe all IP contained in the network segment to obtain the active host IP in the network segment and use port scanning and services detection to determine what services are running in the TCP or UDP ports. After that, the data processing module will process the information and build relationships among services, hosts, and network segments. Finally, the environment data transformation module will use the entity model and property model to transform the processed data to the environment yaml file which is needed by Yusuke's data generation tool. If the network is small, the environment information can be typed manually. If we want to generate the data in a really complicated business network, the network environment mechanism we designed will be helpful. Our designed environment learning mechanism is shown in Figure 8.

After these configurations, the data generation tool will operate and generate the sequence data. The generation tool first selects the attack techniques from the included ATT&CK attack techniques list by the configurations of tools about how to do enterprise penetration attack, and then uses the atomic red team repository and network environment configurations to judge the correctness of the selection. The architecture of the data generation tool is depicted in Figure 9.

Now, have enough network penetration intrusion sequence data, but we still need to process data before training. First, we need to create a dictionary for all intrusion techniques which can map the technique to number. After we get the number sequence data, we need to cut the number sequence data according to the input length and output length of the seq2seq model and add an initial and end flag to it. Finally, we pad the data. We show the data processing in Figure 10.

Till now we have all training data prepared. The overall information about the training dataset is depicted in Table 3. Using the data generation tool, we generate 3600 network penetration scenarios. Counting Yusuke's 800 network penetration scenarios, we have a total of 4400 network penetration scenarios for training and validation. Each of the network penetration scenario is a sequence composed of attack techniques for different tactics. The max length of sequence is 36, i.e., in our generated network penetration scenario, the adversary reached his goal or the end network penetration is up to 36 steps. This "max length" feature is controllable to some extent. We try to set the max length not too big or not too small during the configuration to imitate the real network penetration as much as possible because when the max length is too small, the adversary cannot use enough attack techniques to perform an effective network penetration and when the max length is too big, the adversary is just stuck at a certain stage and cannot go on. Size of the vocabulary represents the number of different attack techniques in these sequences. For every step when

Table 1: Overall information about Yusuke's dataset.

Id	Investigated Group (G) or cybersecurity incident (I) name	Number of penetration scenarios	Core ATT&CK attack techniques	Data source
1	APT29 (G)	100	T1003, T1016, T1018, T1022, T1033, T1041, T1050, T1063, T1074, T1082, T1083, T1105, T1107, T1122, T1140	Yusuke Takahashi
2	Bronze Butler(G)	100	T1003, T1018, T1022, T1041, T1053, T1056, T1060, T1074, T1083, T1105, T1107, T1124, T1140, T1183	Yusuke Takahashi
3	Clinton campaign (I)	100	T1002, T1003, T1018, T1036, T1037, T1041, T1053, T1056, T1070, T1074, T1076, T1083, T1105, T1179, T1201	Yusuke Takahashi
4	Japan pension Service (I)	100	T1003, T1016, T1018, T1041, T1053, T1069, T1074, T1075, T1082, T1083, T1087, T1098, T1105, T1107, T1114, T1135, T1214	Yusuke Takahashi
5	National institute of advanced industrial science and Technology (I)	100	T1003, T1007, T1012, T1018, T1041, T1050, T1070, T1074, T1076, T1081, T1083, T1107, T1110, T1114, T1124, T1135, T1210	Yusuke Takahashi
6	SingHealth (I)	100	T1003, T1018, T1037, T1041, T1074, T1075, T1076, T1081, T1083, T1101, T1105, T1128, T1135, T1140	Yusuke Takahashi
7	South Korean banks and broadcasting organizations (I)	100	T1012, T1015, T1018, T1033, T1041, T1049, T1053, T1056, T1074, T1083, T1087, T1103, T1105, T1140	Yusuke Takahashi
8	Ukrainian electricity distribution companies (I)	100	T1003, T1004, T1007, T1016, T1018, T1028, T1041, T1056, T1074, T1083, T1084, T1087, T1105, T1124, T1135, T1214	Yusuke Takahashi

Table 2: Overall information about our self-made dataset.

Id	Investigated Group (G) or cybersecurity incident (I) name	Number of penetration scenarios	Core ATT&CK attack techniques	Data source
1	Patchwork (G)	400	T1119, T1009, T1088, T1059, T1003, T1132, T1022, T1005, T1074, T1073, T1189, T1173, T1203, T1083, T1107, T1066, T1036, T1112, T1027, T1086, T1093, T1060, T1076, T1093, T1060, T1076, T1105, T1053, T1064, T1063, T1045, T1193, T1192, T1082, T1033, T1204, T1102	Self-made
2	Leviathan (G)	400	T1009, T1197, T1116, T1059, T1074, T1140, T1203, T1027, T1086, T1060, T1117, T1105, T1064, T1023, T1193, T1192, T1204, T1078, T1102, T1047, T1084	Self-made
3	APT32 (G)	800	T1017, T1009, T1094, T1073, T1189, T1068, T1070, T1036, T1050, T1027, T1086, T1117, T1105, T1053, T1216, T1193, T1071, T1082, T1033, T1099, T1204, T1078, T1100	Self-made
4	Redbaldknight (G)	400	T1087, T1009, T1088, T1059, T1003, T1024, T1002, T1132, T1022, T1005, T1039, T1140, T1189, T1203, T1083, T1107, T1036, T1097, T1086, T1060, T1105, T1018, T1053, T1113, T1064, T1193, T1071, T1032, T1124, T1204, T1102	Self-made
5	MagicHound (G)	400	T1059, T1043, T1003, T1002, T1114, T1083, T1107, T1056, T1027, T1086, T1057, T1060, T1105, T1113, T1064, T1193, T1192, T1194, T1071, T1082, T1016, T1033, T1065, T1204, T1102	Self-made
6	Cobalt (G)	400	T1088, T1191, T1173, T1203, T1068, T1107, T1046, T1050, T1027, T1086, T1055, T1108, T1060, T1117, T1219, T1076, T1105, T1053, T1064, T1193, T1192, T1071, T1032, T1204, T1220	Self-made
7	Dragonfly (G)	800	T1087, T1098, T1110, T1059, T1043, T1136, T1003, T1002, T1005, T1074, T1089, T1189, T1114, T1133, T1083, T1107, T1187, T1070, T1036, T1112, T1135, T1069, T1086, T1012, T1060, T1076, T1105, T1018, T1053, T1113, T1064, T1023, T1193, T1192, T1071, T1016, T1033, T1221, T1204, T1078, T1100	Self-made

predicting, the model outputs a 76-dimensional vector. The attack technique represented by the largest scalar number in the vector is the result of the prediction at that step. The number of training sequences is the sequences that we put into the model when training. It is much bigger than the number of penetration scenarios because we split each scenario to dozens of short sequences such that the model can run efficiently and make the most of data. The length of the input and output sequences affects the accuracy, which is reflected in the later experimental results. We will talk about the limitation of the dataset in the discussion section.

3.3. Software Material. We implement our seq2seq model in Python3.8 with TensorFlow2.4 package. We create an inherited encoder and decoder class from TensorFlow's "keras.Model" class and build the seq2seq model. We implement the data generation tool based on Yusuke's data generation tool in Python3.8 with Faker, fnvhash, tinydb, and PyYAML packages.

4. Results

It is worth noting that we do not provide much information on the comparison with other existing methods for the reason that there are no existing seq2seq methods used in the intrusion prediction field as far as we know. There are some existing methods in the prediction task based on the intrusion sequence data, and thus we compare our accuracy with that of the good existing method. For shallow learning methods, as they have been proven not effective in the intrusion prediction field in other papers, we do not provide these comparison experiments.

4.1. Overall Prediction Result. Our seq2seq model has three implementations, the LSTM-seq2seq, the RNN-seq2seq, and the GRU-seq2seq. In this section, we evaluate the overall performance of our seq2seq model in predicting the future intrusion techniques' sequence. The amount of our dataset is not big, so we do not separate the data by the data source at the first beginning and the main purpose of the experiment is to prove the ability of our model to capture the internal logical knowledge behind the intrusion technique sequence. We will talk about the model's performance when training and testing on completely different data source in the next section.

At the first stage of the experiment, we shuffle the data and split it in the ratio of 0.8 and 0.2. The first part is for training and the other is for validation. Then, we adjust the dropout, epoch, and hidden unit parameters to avoid overfitting while training and testing the three seq2seq models. The overall prediction result of the LSTM-seq2seq model is shown in Table 4.

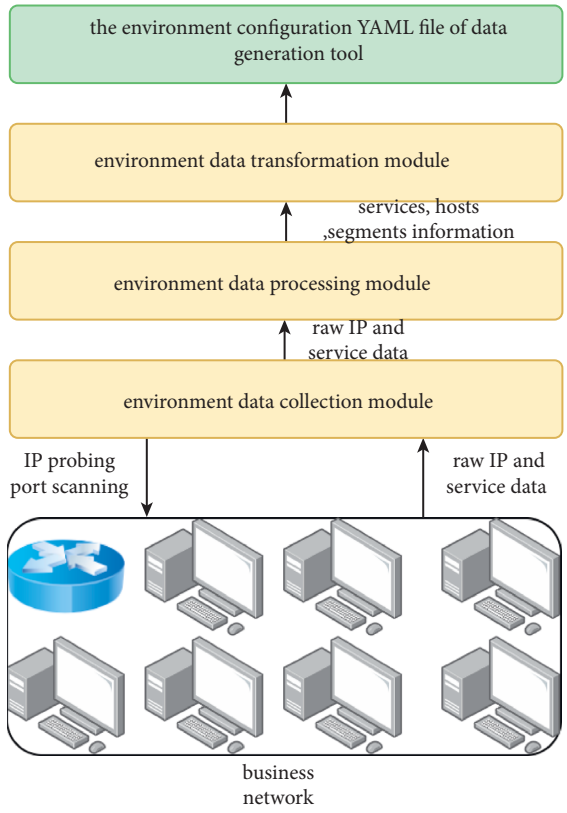

Figure 8: Our designed network environment data collection mechanism.

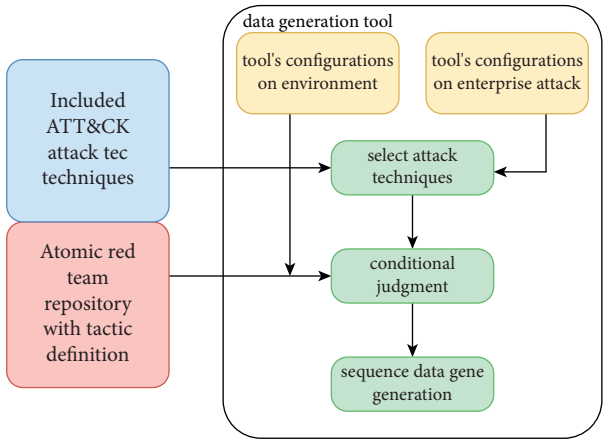

Figure 9: The architecture of the data generation tool.

We set the number of hidden units to 64 because overfitting happens when the number of hidden units is too big due to the limitation of data amount, and after experiments we find increasing the number of hidden units and dropout parameter will not bring great improvement to this model. We also find the accuracy on the validation dataset decreases slightly with the increasing of the dropout parameter, which helps avoid overfitting in the model.

We can observe from the result that the input length and the output length are critical to the performance of the model. When the output length is set to one, the LSTM-seq2seq model is nearly the same with the common LSTM model. The LSTM-seq2seq model with an output length 1 encodes the input data to a hidden state, its decoder decodes the hidden state, and start flag to a decoder hidden state which is used to generate the final output. The start flag has no meaning and will not bring much difference to the output. In this situation, the model reaches an accuracy of 0.90 which is not far from other results in the field of intrusion prediction, such as 0.93 for predicting the security event [14], proving that this model we built can extract and learn the information in penetration intrusion sequence data efficiently.

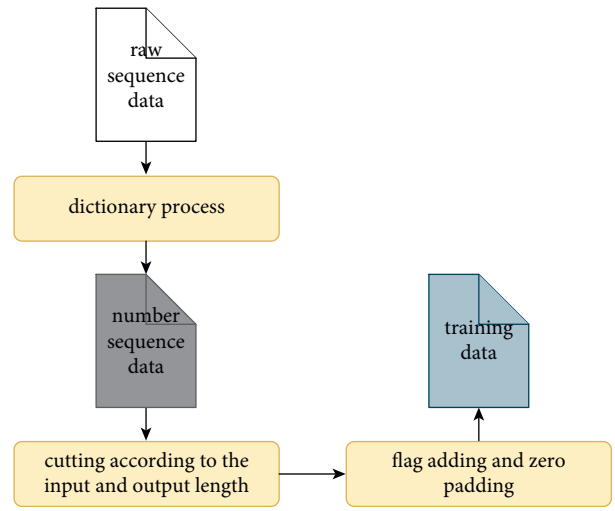

Figure 10: Data processing diagram.

When we change the input length of the model, we find that the validation accuracy of the model barely changed. The result proves that the input length of 5 is enough for the model to extract information about the penetration intrusion sequence. The result also proves the data limitation that sequences with a length of more than 5 contain almost no more information than shorter sequences. To better check that, we performed more experiments about the input length and the output length; the result is shown in Figure 11.

From the heatmap, we can observe that models with an input length of more than 5 perform not much better than models with an input length of 5, and models with an input length of less than 5 perform much worse than models with an input length of 5. It proves our previous thoughts that sequences with a length of more than 5 contain just few more information due to the limitations of the dataset and application scenario. We can easily understand this phenomenon; when the length of the input penetration intrusion sequence is more than 5, the relationship between the last and first penetration intrusion techniques is pretty weak. Although there is still a possibility that strong correlation exists between the two penetration intrusion techniques, like techniques for initial access tactic and techniques for lateral movement tactic, the limitations of the generated data make this improvement of long sequence input not too obvious. We will talk about the limitations of the dataset in the discussion section. Now considering the

Table 3: Overall information about the training dataset.

Number of penetration scenarios	4400
Number of sequences	About 100k, change slightly with the input and output length
Size of vocabulary	74
Max length of sequence	36

Table 4: The overall prediction result of the LSTM-seq2seq model.

LSTM-seq2seq model prediction result id	Input length	Output length	Accuracy on validation dataset	Number of hidden units
1	5	1	0.89	64
2	6	1	0.90	64
3	7	1	0.90	64
4	5	1	0.90	128
5	5	2	0.87	64
6	5	3	0.85	64
7	5	4	0.82	64
8	5	5	0.79	64
9	5	6	0.75	64
10	5	10	0.60	64
11	5	15	0.47	64

above analysis, we set the input length to 5 in the following experiments.

The accuracy decreases with an increase in the output length. When we change the output length of the model, the validation accuracy changes rapidly, from 0.89 to 0.47. It shows the limitation on the seq2seq model: long distance prediction and dataset; we will talk in detail about this phenomenon and the solution in the actual application in the following section.

We also use the RNN-seq2seq and GRU-seq2seq models to run the experiments. The overall prediction result of the RNN-seq2seq and GRU-seq2seq models is shown in Table 5. The comparison heatmap between the three models is show in Figure 12.

From the result, we observe that the information extraction and the inference ability of RNN and GRU are weaker than that of LSTM in the application scenario of sequence prediction. Especially, the validation accuracy of the RNN-seq2seq model drops dramatically with an increase in the output length, proving the basic RNN component is not suitable for the seq2seq model. We have confirmed that LSTM is the best component in these three for the seq2seq model; thus, further experiments on the influence of the training dataset and sequence length will only use the LSTM-seq2seq model. Readers can assume the RNN-seq2seq and the GRU-seq2seq as a weakened version of LSTM-seq2seq in this application scenario.

4.2. Influence of Sample Distribution. It's important to measure one model's inference ability in a completely new environment, and so-called sample distribution. In the experiment of the binary classification model, the distribution of positive and negative samples can produce a ROC curve, and finally prove the discrimination ability of the model. In a multi-classification model, the sample distribution mainly refers to the samples in different scenes or at different times. If the training data and validation dataset of the multi-classification model belong to different distributions, the change of accuracy will reflect the complexity of the problem solved by the model and the necessity of updating the model regularly. The measurement results will reflect the robustness and practicability of the model to new data and new environment to a certain extent.

Due to the limitation of the data amount, we shuffle the data before the overall experiments to prove the model's ability to catch and learn the internal knowledge in the sequence data. After the overall experiments, we now design some experiments to test the influence of the training dataset. For convenience, we only use the LSTM-seq2seq model with the input and output length of 5 and 3. First, we cut the whole dataset in Table 1 and Table 2 to separate the dataset. In our definition, dataset 1 contains data from APT29, Bronze Butler, Clinton campaign, and Japan Pension Service; dataset 2 contains data from National Institute of Advanced Industrial Science and Technology, SingHealth, South Korean banks and broadcasting organizations, and Ukrainian electricity distribution companies; dataset 3 contains data from Patchwork and Leviathan; dataset 4 contains data from APT32; dataset 5 contains data from Redbaldknight and Magic; dataset 6 contains data from Cobalt and Dragonfly. For a specific dataset, we train the model on the rest dataset and test the model on the specific dataset. We show the result in Figure 13.

From the result, we can observe that all experimental accuracy results on completely separate datasets are much smaller than the overall experiment's accuracy. That's because there is penetration scenario information in the separate test dataset which does not exist in the training dataset.

We can also observe that the experimental accuracy results on dataset 1 and dataset 2 are much smaller than the rest. That's because of the different data generation methods. We get the public dataset 1 and dataset 2. As for datasets 3–6, we investigate the adversary groups about the attack ATT&CK techniques and tools they used. During the investigation, we must investigate the attack techniques for

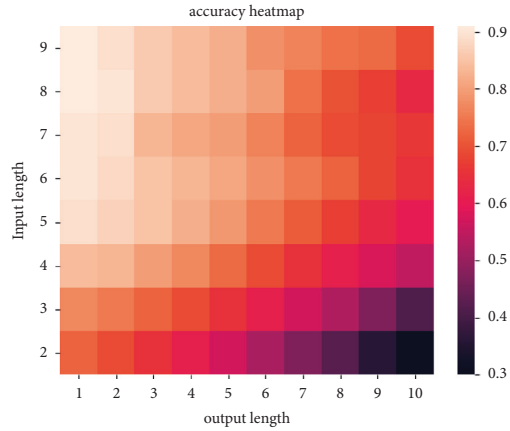

Figure 11: The accuracy heatmap of the LSTM-seq2seq model.

Table 5: The overall prediction result of the RNN-seq2seq and GRU-seq2seq models.

Model prediction result id	Model component	Input length	Output length	Accuracy on validation dataset
1	RNN	5	1	0.83
2	RNN	5	2	0.80
3	RNN	5	3	0.74
4	RNN	5	5	0.64
5	RNN	5	10	0.43
6	GRU	5	1	0.87
7	GRU	5	2	0.85
8	GRU	5	3	0.82
9	GRU	5	5	0.72
10	GRU	5	10	0.53

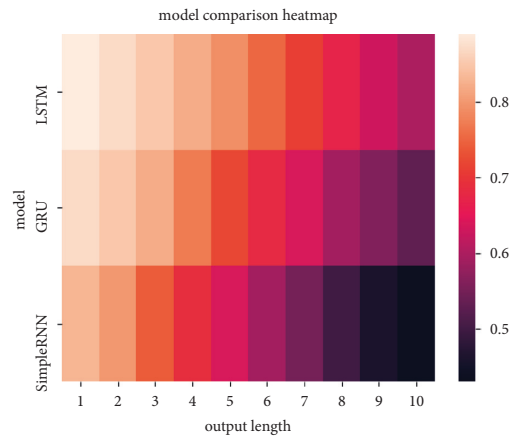

Figure 12: The comparison heatmap of models' performance.

different tactics such that these attack techniques can form a complete penetration scenario which is needed by the data generation tool. These tactics include persistence, privilege escalation, defense evasion, credentials access, discovery, lateral movement, collection, and exfiltration. Many groups do not have enough attack techniques to form complete penetration scenarios, and thus we add some techniques to the attack techniques list of these adversary groups, resulting in some similarity among datasets 3–6. That's the reason why the experimental result on separate datasets 3–6 is better than the rest.

In this section, we evaluate the performance of our model on completely separate datasets and analyse the reason for the accuracy decreasing. When applied to an actual business network environment, we still have to supplement the dataset.

4.3. Influence of Sequence Length. From the overall performance of our model, we can observe that sequence length is a critical parameter, especially the output sequence length. Seq2seq is actually a machine learning framework. After a seq2seq model gets one prediction result, it will put the result to the next input such that the model can predict a whole sequence. However, the prediction result of one timestamp contains not only the information of input and previous predictions, but also the error accumulated during previous predictions. The longer the sequence, the bigger the accumulated error. That's the reason why the prediction accuracy drops from 0.89 to 0.47 with an increase in the output length.

To solve the problem, in the real-life application of the model, we can set the length of the output sequence not too big and update the input sequence frequently. To implement this design, we should create a stack to store the ATT&CK techniques and an algorithm to decide whether we should replace the prediction technique with new techniques or create a new input sequence at the time of prediction; we should also set enough zero padding location in the input sequence such that we can enter the changeable sequence data to the input for model inference. It's worth noting that

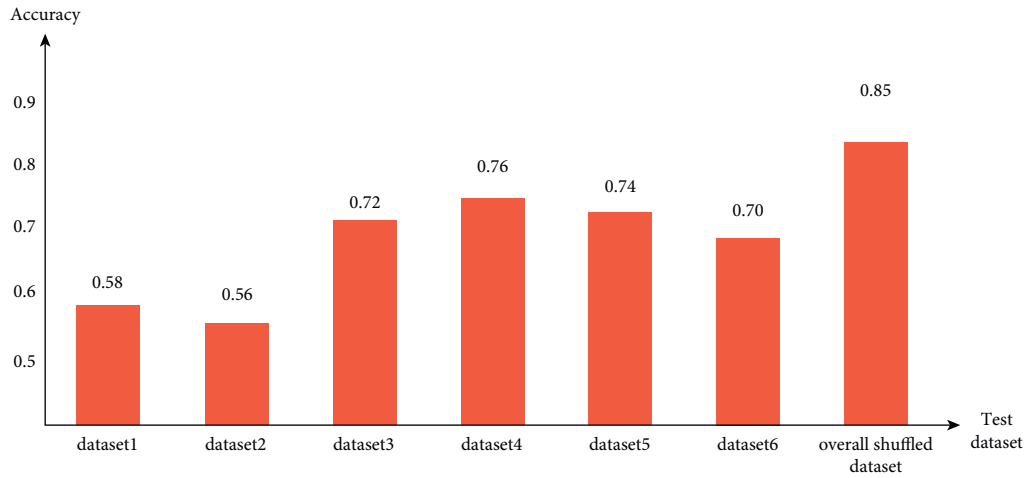

Figure 13: The experimental result on a completely separate dataset.

we still need to map the cybersecurity information, maybe including exploit samples and logs that occurred in the actual network to the ATT&CK techniques, but that's not the scope of this paper; readers can learn about the mapping on the ATT&CK web page. The application scenario is shown in Figure 14.

5. Discussion

5.1. Data Limitations. Seq2seq is a machine learning framework; it encodes the existing sequence and decodes the context information to predict the future sequence. The seq2seq model is a statistic model in a broad sense. The application scenario of our model is network penetration intrusion prediction. The model should be able to capture the internal information from these sequences about how adversaries run network penetration. Facing different network environments, the adversary will take different steps. The ATT&CK framework knowledge base turns the adversary intrusion behaviour to hundreds of attack techniques for several tactics such that the adversary intrusion behaviour can be modelled and predicted. However, the dataset about network penetration intrusion is so small; even in some big-scale and destructive cybersecurity incidents, the attack sequences that intruders have followed is not much and hard to collect. Even in this situation of the dataset, under consideration of insecurity brought by the exposure of software, hardware, and network conditions, most corporations will not publicize corresponding attack sequence information and logs after they are attacked, bringing inconvenience to the research of incident response and intrusion prediction.

To solve the data amount problem and meet the basic data amount requirement of the model, we have to use the public attack sequence data and generate more data based on the investigation of adversary groups. In fact, public data are also generated from investigating adversary groups and cybersecurity incidents. There are barely no public attack sequence data that only come from real network penetration.

The data generation tool is essential to the quality of the dataset. As mentioned above in the data material section, we just need to initialize the tool and give the tool a list of ATT&CK attack techniques. The data generation tool will select one technique at one step based on its choice strategy, which includes a lot of attack information, and then judge its legitimacy using the atomic red team repository and other functional codes.

The model is trained and tested on generated data, proving its ability to learn the sequence information. However, when used in real applications, the data need to be updated and fixed. The dataset still has some limitations.

First, more investigation needs to be done. The dataset needs to contain all attack techniques to make the model more practical. In real applications, we need to investigate more penetration scenarios to let the dataset contain information among all these attack techniques such that the model can imitate the existing attack patterns at least.

Second, the data generation material needs to be updated. All material related to the data is based on the ATT&CK framework version 3, including the atomic read team repository, the technique lists we collected, and the data generation tool. The latest version of the ATT&CK framework contains sub-techniques under one specific attack technique and decreases the number of techniques, making the framework more reasonable. Our paper just provides a model and generates some demonstrative data to solve the intrusion prediction problems. If used in real applications, the data material still needs to be updated, and the endeavour in this update will be quite big.

Third, data enrichment and augmentation need to be done. Under the framework of the ATT&CK cybersecurity threat knowledge base, we can try to generate other intrusion sequence dataset to supplement our dataset using other generation methods in the future work. There is some researchabout attack plans generation [48, 49] and red team emulation [50, 51]. We can modify these models using the threat definition of the ATT&CK framework and generate more targeted attack sequence data. In future works, we can also use expertise knowledge to score these datasets and use

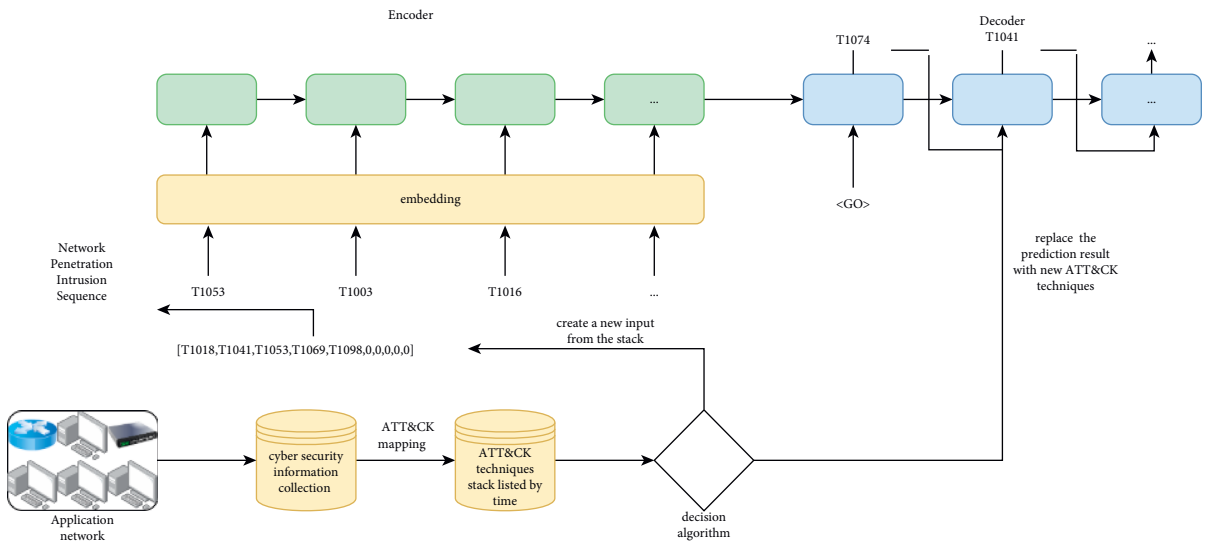

Figure 14: The designed prediction scenario of our model.

data augmentation for datasets with high scores, such as repeating, random cutting, sample pairing, and variation auto-encoder data augmentation.

5.2. Model Limitations. Since application scenarios need our model to capture the characteristics in the input techniques' sequences, we add the LSTM, RNN, and GRU components and attention mechanism to this model. Although previous experimental results have proven its ability to capture and learn the characteristics of the sequence data, our seq2seq model still has some limitations and can be improved. We now discuss the model limitations in this section so that researchers can modify and improve our model more easily.

First, there are still many components that can make up the seq2seq model, including CNN [52], transformer [53], and pointer generator network [54]. When using these models to solve the intrusion prediction problem, the dataset also needs to be fixed according to the requirements of the models. These components provide different ways to capture the characteristics in the sequences data. These models may be more suitable for penetration intrusion sequence prediction problems.

Second, more mechanisms can be added to the seq2seq model for the penetration intrusion prediction application scenario. Now, our model takes the penetration intrusion sequence as input, learning the characteristics in these sequences. However, the intrusion sequence information is highly related to the information of the network environment in which the penetration really occurs. Although the choice of intrusion techniques must have considered the network environment already, there is still missing environment information in the prediction scenario, which influences the performance of the model and brings uncertainty to the real-life application of the model. We need to add more mechanism to give the model the ability to use more information about the network environment, software environment, hardware environment, and even some more. However, most existing mechanisms about the seq2seq model care about the information in these sequence data, like attention and self-attention mechanism, not providing a way to compute more information parameters. If we want other environment parameters to be included in the model, we might use some reinforcement learning mechanism in the seq2seq model, as discussed in [55]. We can consider the environment information and other not directly related information as state and let the state influence the model. It still needs a lot of theoretical work.

5.3. Deployment and Application. As already discussed, besides dataset and model improvement, we still need to map the security information collected in a network to the ATT&CK attack techniques when we actually deploy the seq2seq model. The work of the actual deployment will be heavy and complicated and we have discussed it in the previous sections. To map the security information to techniques and tactics, we need to collect these kinds of information first, including shell commands, malware actions, network packets, user account information, device and network access information, alert information, and other related information. We can use some information collection tools like Elasticsearch Beats, Beats-based scripts, network information protocol, SNMP, and so on. After that, we need to extract the critical part using some data search engine or machine learning and search it in the ATT&CK threat knowledge base, atomic red team repository, or other cybersecurity knowledge database to judge what tactic and technique the intrusion belongs to. The designed raw data mapping mechanism in real applications is shown in Figure 15.

Now, we turn our attention back to the model's application with other cybersecurity systems after deployment. After we deploy the model, we can get the future several attack techniques with different tactics the adversary may take. If we input the information into the intrusion

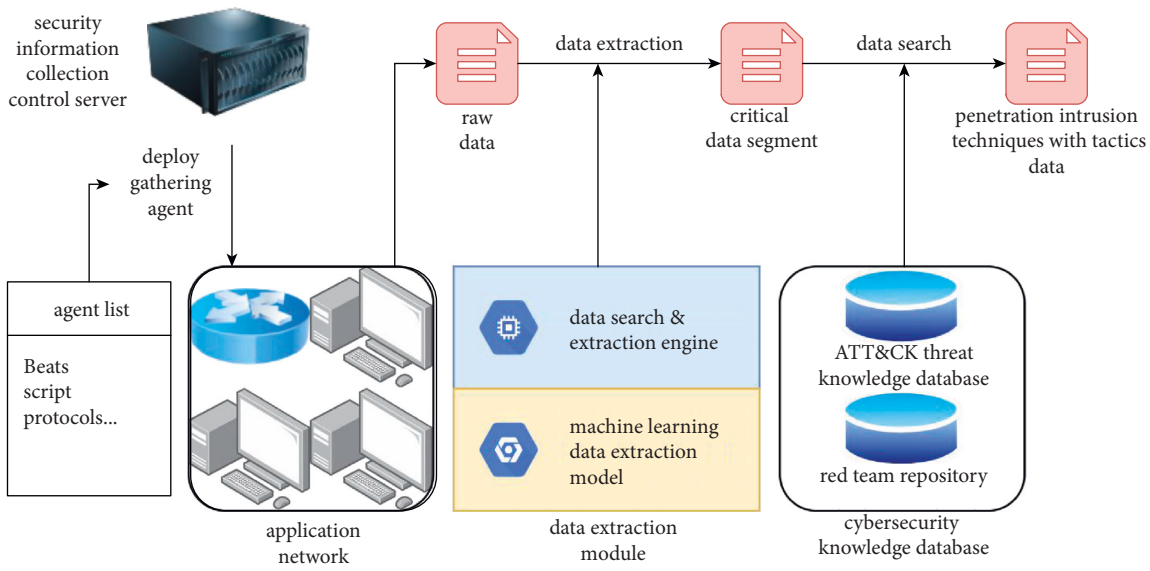

Figure 15: Our designed raw data mapping mechanism in actual application.

prevention system, the system can know the intrusion tactics the adversary wants to achieve and take actions accordingly; for example, the IPS can check or reset the user right after predicting the attack techniques for privilege escalation tactic; the IPS can restrict access to important assets after predicting the attack techniques for collection and exfiltration tactics; the IPS can limit software installation and check boot integrity after predicting some attack techniques for persistence tactic. The security system can provide many mitigation suggestions with the help of the ATT&CK framework, as discussed in [56]. More details about how to take mitigation actions to fight against the attack techniques can be seen on the ATT&CK mitigation web page. If we put the information into the deception defense system, the information can help the deception defense system in the same way. Honeynet is one kind of deception defense system. If honeypot in the system is based on a docker, as discussed in [41], it can change its service and asset information easily to attract the hacker according to the predicted attack techniques the adversary will take; this way the honeynet can trap the hacker and collect information, as discussed in [35].

6. Related Work

6.1. Intrusion Sequence Data Generation. Our intrusion prediction model is based on the ATT&CK enterprise-targeted attack sequence data generated by APTGen [19]. There are also many similar researches about attack sequence generation, attack tree generation, and automatic penetration of red team due to the prohibitive cost and requirements of deploying red team and the data collection difficulty. Attack plan generation and attack graph generation are researched in these papers [48, 49, 57]. Automatic penetration of red team is researched in these papers [50, 51, 58]. These papers [59, 60] discuss the strategy and the uncertainty details about penetration attack. These models and strategies are based on some theories and some cybersecurity threat frameworks like the Markov Decision Process, Cyber Kill Chain framework, and the ATT&CK framework. However, most existing papers have the problem that theories cannot be implemented or there are too few attack techniques and tactics to be considered. APTGen generated multiple attack sequences that an adversary may execute in a targeted environment with the help of the ATT&CK framework and the atomic red team repository. Thus, we choose APTGen as our dataset and data generation tool finally.

6.2. Intrusion Prediction. Our intrusion prediction approach is based on attention seq2seq model with the RNN, LSTM, and GRU components. In the past several years, there have been many efforts to use machine learning methods to solve intrusion prediction problems in many scenarios. For instance, Tiresias [14] uses the LSTM model to predict upcoming security events based on security events' sequence with comparison to the Markov Chain model and spectral learning model. Ansari [17] uses the Conv-LSTM model to predict the upcoming alert information based on the existing alert sequence with comparison to shallow learning. Liu [61, 62] collected external measurable features about an organization's business network to predict security incidents. As for the prediction of Cyber Kill Chain-related penetration intrusion prediction, there are just few researches. Danneman [63] built a model to predict lateral movement patterns. Noor [64] proposed a machine learning framework to identify cybersecurity threats based on observed attack patterns. Shaer [43] proposed a hierarchical clustering algorithm to provide statistically significant and explainable technique correlations of adversarial attack techniques and tactics of the ATT&CK framework. They also proposed one thought that constructing the associations will enable predicting unobserved attack techniques based on observed ones. As far as we know, we are the first to build models to predict specific adversarial techniques and tactics based on the ATT&CK framework.

7. Conclusions

In this paper, we built a seq2seq model with attention mechanism for the prediction of network penetration intrusion prediction. We evaluated the model on public and generated attack sequence data under the ATT&CK framework. The model reaches a high accuracy when the sequence length is short and the dataset is simple, showing its ability to catch information and learn features in these penetration intrusion sequence data. The model's accuracy decreases with an increase in the output length and data complexity due to the inherent data limitations and model limitations, which brings difficulties to the application of the model. We discussed these limitations, possible solutions, and model applications with other cybersecurity systems. We will then try to solve these problems and apply the prediction model to other systems, as our future work.

Data Availability

The data and data generation tool used to support this paper are all public; the readers can read the "data material" part of this paper to know how to obtain the data and the data generation tool to generate more data for further research.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the "National Key R&D Program of China (2020YFB1708602, 2020YFB1708600)", "Major Scientific and Technological Special Project of Guizhou Province (20183001)" and the "Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No.2017BDKFJJ015, No.2018BDKFJJ008, No.2018BDKFJJ020, No.2018BDKFJJ021)".

References

[1] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats, advanced information systems engineering," in *Communications and Multimedia Security. CMS 2014. Lecture Notes in Computer Science*, B. De Decker and A. Zúquete, Eds., vol. 8735, pp. 63–72, Springer, Berlin, Heidelberg, 2014.

[2] D. N. Pande and P. S. Voditel, "Spear phishing: diagnosing attack paradigm," in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2720–2724, Chennai, India, March 2017.

[3] S. Eggers, "A novel approach for analyzing the nuclear supply chain cyber-attack surface," *Nuclear Engineering and Technology*, vol. 53, no. 3, pp. 879–887, 2021.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, Article ID 41550, 2019.

[5] S. Brookes and S. Taylor, "Containing a confused deputy on x86: a survey of privilege escalation mitigation techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, 2016.

[6] P. S. Nyakomitta and D. S. O. Abeka, "A survey of data exfiltration prevention technique," *International Journal of Advanced Networking and Applications*, vol. 12, no. 03, pp. 4585–4591, 2020.

[7] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Computing Surveys*, vol. 50, no. 3, pp. 1–40, Article ID 41, 2018.

[8] S. Das and M. J. Nene, "A survey on types of machine learning techniques in intrusion prevention systems," in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2296–2299, Chennai, India, March 2017.

[9] S. Das and M. J. Nene, "A survey on types of machine learning techniques in intrusion prevention systems," in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2296–2299, IEEE, Chennai, India, March 2017.

[10] M. Husak, J. Komarkova, E. Bou-Harb, and P. Celeda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2019.

[11] X. Wenke Lee and W. Lee, "Attack plan recognition and prediction using causal networks," in *Proceedings of the 20th Annual Computer Security Applications Conference*, pp. 370–379, Tucson, AZ, USA, December 2004.

[12] Z. T. Li, J. Lei, L. Wang, and D. Li, "A data mining approach to generating network attack graph for intrusion prediction," in *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, pp. 307–311, Haikou, Hainan, China, September 2007.

[13] Y. B. Leau and S. Manickam, "Network security situation prediction: a review and discussion," in *Proceedings of the International Conference on Soft Computing, Intelligence Systems, and Information Technology*, pp. 424–435, Springer, Chennai, India, March 2015.

[14] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias: predicting security events through deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, pp. 592–605, Association for Computing Machinery, Toronto Canada, October 2018.

[15] S. Lv, J. Wang, and Y. Yang, "Intrusion prediction with system-call sequence-to-sequence model," *IEEE Access*, vol. 6, pp. 71413–71421, 2018.

[16] A. Kulkarni, "Predicting security events using attention neural network," Scramento State, CA, USA, Ms-project, 2021.

[17] M. S. Ansari, V. Bartos, and B. Lee, "Shallow and deep learning approaches for network intrusion alert prediction," *Procedia Computer Science*, vol. 171, pp. 644–653, 2020.

[18] B. E. Strom, A. Applebaum, and D. P. Miller, "Mitre att&ck: design and philosophy," Technical report, 2018.

[19] Y. Takahashi, S. Shima, and R. Tanabe, "APTGen: an approach towards generating practical dataset labelled with targeted attack sequences," in *Proceedings of the 13th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 20)*MA, USA, August 2020.

[20] mitre, "MITRE ATT&CK V3 Group Documentation page," 2019, https://attack.mitre.org/versions/v3/groups/.

[21] S. J. Yang, H. Du, J. Holsopple, and M. Sudit, *Attack Projection*, Springer International Publishing, Cham, pp. 239–261, 2014.

[22] A. A. Ahmed and N. A. K. Zaman, "Attack intention recognition: a review," *IJ Network Security*, vol. 19, no. 2, pp. 244–250, 2017.

[23] Y.-B. Leau and S. Manickam, *Network Security Situation Prediction: A Review and Discussion*, pp. 424–435, Springer, Berlin, Heidelberg, 2015.

[24] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *Proceedings of the 23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 625–640, San Diego, CA, USA, August 2014.

[25] L. Bilge, Y. Han, and D. A. M. Riskteller, "Predicting the risk of cyber incidents," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1299–1311, Dallas Texas USA, October 2017.

[26] M. Abdlhamed, K. Kifayat, Q. Shi, and W. Hurst, *Intrusion Prediction Systems*, Springer International Publishing, Cham, pp. 155–174, 2017.

[27] R. Sadoddin and A. Ghorbani, "Alert correlation survey: framework and techniques," in *Proceedings of the 2006 international conference on privacy, security and trust: bridge the gap between PST technologies and business services*, pp. 1–10, Ontario, Canada, October 2006.

[28] L. Feng, X. Guan, S. Guo, Y. Gao, and P. Liu, "Predicting the intrusion intentions by observing system call sequences," *Computers & Security*, vol. 23, no. 3, pp. 241–252, 2004.

[29] G. Zhang and J. Sun, "A novel network intrusion attempts prediction model based on fuzzy neural network," *Lecture Notes in Computer Science*, vol. 3991, pp. 419–426, 2006.

[30] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Towards a forecasting model for distributed denial of service activities," in *Proceedings of the 2013 I.E. 12th international symposium networking and computer application*, pp. 110–117, Cambridge, MA, USA, August 2013.

[31] H. Park, S. Jung, and H. Lee, "H.P.: cyber weather forecasting unknown internet worms using randomness analysis," *IFIP Advances in Information and Communication Technology*, vol. 376, pp. 376–387, 2012.

[32] A. Shameli Sendi, M. Dagenais, M. Jabbarifar, and M. Couture, "Real time intrusion prediction based on optimized alerts with Hidden Markov Model," *Journal of Networks*, vol. 7, no. 2, pp. 311–321, 2012.

[33] J. Wu, L. Yin, and Y. Guo, "Cyber attacks prediction model based on Bayesian network," in *Proceedings of the 2012 I.E. 18th international conferences parallel and distributed systems*, pp. 730-731, Singapore, December 2012.

[34] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," in *Cyber Deception*, S. Jajodia, V. Subrahmanian, V. Swarup, and C. Wang, Eds., Springer, Cham, 2016.

[35] L. Spitzner, "The Honeynet Project: trapping the hackers," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 15–23, 2003.

[36] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici, "HoneyGen: an automated honeytokens generator," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI'11)*, pp. 131–136, IEEE, Beijing, China, November 2011.

[37] S. Antonatos, P. Akritidis, and E. P. Markatos, "Defending against hitlist worms using network address space randomization," *Computer Networks*, vol. 51, no. 12, pp. 3471–3490, 2007.

[38] Z. Zhao, F. Liu, and D. Gong, "An SDN-based fingerprint hopping method to prevent fingerprinting attacks," *Security and Communication Networks*, vol. 2017, Article ID 1560594, 12 pages, 2017.

[39] L. Zobal, D. Kolář, and R. Fujdiak, "Current state of honeypots and deception strategies in cybersecurity," in *Proceedings of the 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 1–9, IEEE, Dublin, Ireland, October 2019.

[40] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in network security," *Security and Communication Networks*, vol. 4, no. 10, pp. 1162–1172, 2011.

[41] D. Sever and T. Kišasondi, "Efficiency and security of docker based honeypot systems," in *Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1167–1173, Opatija, Croatia, May 2018.

[42] A. Patel, Q. Qassim, and C. Wills, "A survey of intrusion detection and prevention systems," *Information Management & Computer Security*, vol. 18, no. 4, pp. 277–290, 2010.

[43] R. Al-Shaer, J. M. Spring, and E. Christou, "Learning the associations of MITRE ATT & CK adversarial techniques," in *Proceedings of the 2020 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, IEEE, Avignon, France, July 2020.

[44] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 2014, pp. 3104–3112, 2014.

[46] E. Aghaei and E. Al-Shaer, "Threatzoom: neural network for automated vulnerability mitigation," in *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, pp. 1–3, Nashville, TN, USA, April 2019.

[47] githu, "Atomic Red Team Repository," 2021, https://github.com/redcanaryco/atomic-red-team.

[48] J. Hoffmann, "Simulated penetration testing: from" dijkstra" to" turing Test++"" in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling*, Jerusalem, Israel, June 2015.

[49] G. Falco, A. Viswanathan, C. Caldera, and H. Shrobe, "A master attack methodology for an AI-based automated attack planner for smart cities," *IEEE Access*, vol. 6, pp. 48360–48373, 2018.

[50] A. Applebaum, D. Miller, and B. Strom, "Intelligent, automated red team emulation," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 363–373, Los Angeles, CA, USA, December 2016.

[51] S. Randhawa, B. Turnbull, and J. Yuen, "Mission-centric automated cyber red teaming," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pp. 1–11, University of Hamburg, Germany, August 2018.

[52] G. Zhang, X. Bai, and Y. Wang, "Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism," *Machine Learning with Applications*, vol. 5, Article ID 100064, 2021.

[53] E. Egonmwan and Y. Chali, "Transformer and seq2seq model for paraphrase generation," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 249–255, Hong Kong, November 2019.

[54] H. Kumari, S. Sarkar, and V. Rajput, "Comparative analysis of neural models for abstractive text summarization," in *Proceedings of the International Conference on Machine Learning,*

Image Processing, Network Security and Data Sciences, pp. 357–368, Springer, Assam, India, April 2020.

[55] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2469–2489, 2020.

[56] W. Xiong, E. Legrand, and O. Åberg, "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *Software and Systems Modeling*, vol. 21, pp. 157–177, 2022.

[57] J. L. Obes, C. Sarraute, and G. Richarte, "Attack planning in the real world," 2013, https://arxiv.org/abs/1306.4044.

[58] H. T. Ray, R. Vemuri, and H. R. Kantubhukta, "Toward an automated attack model for red teams," *IEEE Security & Privacy*, vol. 3, no. 4, pp. 18–25, 2005.

[59] K. Durkota and V. Lisý, "Computing optimal policies for attack graphs with action failures and costs," in *Proceedings of the STAIRS 2014: Proceedings of the 7th European Starting AI Researcher Symposium*, pp. 101–110, Prague, Czech Republic, August 2014.

[60] C. Sarraute, O. Buffet, and J. Hoffmann, "POMDPs make better hackers: accounting for uncertainty in penetration testing," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada, July 2012.

[61] Y. Liu, A. Sarabi, and J. Zhang, "Cloudy with a chance of breach: forecasting cyber security incidents," in *Proceedings of the 24th {USENIX} Security Symposium ({USENIX} Security 15)*, pp. 1009–1024, Washington D. C, USA, August 2015.

[62] Y. Liu, J. Zhang, and A. Sarabi, "Predicting cyber security incidents using feature-based characterization of network-level malicious activities," in *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, pp. 3–9, San Antonio, TX, USA, March 2015.

[63] N. Danneman and J. Hyde, "Predicting adversary lateral movement patterns with deep learning," 2021, https://arxiv.org/abs/2104.13195.

[64] U. Noor, Z. Anwar, and A. W. Malik, "A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories," *Future Generation Computer Systems*, vol. 95, pp. 467–487, 2019.