TSEE: A Novel Knowledge Embedding Framework for Cyberspace Security

Angxiao Zhao
Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China
Zhaoquan Gu (guzhaoquan@hit.edu.cn)
School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
Yan Jia
Department of New Networks, Peng Cheng Laboratory
Wenying Feng
Department of New Networks, Peng Cheng Laboratory
Jianye Yang
Cyberspace Institute of Advanced Technology, Guangzhou University
Yanchun Zhang
Department of New Networks, Peng Cheng Laboratory

Research Article

Keywords: Cyber security, MDATA model, spatio-temporal knowledge representation, dynamic knowledge embedding

Posted Date: September 7th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3308655/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at World Wide Web on December 20th, 2023. See the published version at https://doi.org/10.1007/s11280-023-01220-9.

TSEE: A Novel Knowledge Embedding Framework for Cyberspace Security

Angxiao Zhao[1,3], Zhaoquan Gu[2,3*], Yan Jia[2,3], Wenying Feng[3], Jianye Yang[4], Yanchun Zhang[3]

[1]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China.
[2]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China.
[3]Department of New Networks, Peng Cheng Laboratory, Shenzhen, China.
[4]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China.

*Corresponding author(s). E-mail(s): guzhaoquan@hit.edu.cn;
Contributing authors: 202122280745@std.uestc.edu.cn; jiay@pcl.ac.cn; fengwy@pcl.ac.cn; jyyang@gzhu.edu.cn; zhangych03@pcl.ac.cn;

Abstract

Knowledge representation models have been extensively studied and they provide an important foundation for artificial intelligence. However, the existing knowledge representation models or related knowledge embedding methods mostly aim at static or temporal knowledge, which are not suitable for highly spatio-temporal relevant knowledge, such as the cyber security knowledge. In this paper, we propose a knowledge embedding framework called TSEE to handle this problem, which builds on the MDATA model to represent and utilize dynamic knowledge for cyber security. TSEE is composed of knowledge extraction module, knowledge representation module, knowledge embedding module, and situational awareness module. There modules can obtain, transform, and embed cyber security knowledge from different sources, improving the detection capabilities of various complicated attacks. We conduct experiments on the cyber range for evaluation, and the experimental results validate the higher prediction accuracy and stronger extendability than existing embedding methods. The framework can effectively improve the cyber security defense capabilities in the future.

Keywords: Cyber security, MDATA model, spatio-temporal knowledge representation, dynamic knowledge embedding.

1 Introduction

Knowledge is the accumulated intellectual asset, which contributes potential value to science and technology. In recent years, knowledge has been applied in scientific research, e-commerce, wise medicine, etc. With the maturation and advancement of information technology, knowledge can be obtained more conveniently and comprehensively. For example, artificial intelligence (AI) can maximize the linkage between knowledge and business information. By analyzing and processing big data, the revenue of companies can be expanded. In cyberspace security, vast amounts of knowledge need to be stored in intrusion detection system (IDS), such as known attacks, vulnerabilities, and attacker behavior characteristics. This enables IDS to accurately detect and report threats, and take timely measures [1].

Knowledge representation [2] aims to design a method to transform knowledge into a template, so that computers can understand human language and assist humans in judgment and thinking. Complex AI tasks such as knowledge reasoning will become easier by means of knowledge representation. Among numerous knowledge representation researches (see Part 2.1), knowledge graph [3] is the most popular method due to its friendly structure and human-like expression. Specifically, entities are typically represented by nodes, and relations are represented by edges. Knowledge can be described as *(head entity, relation, tail entity)*. For example, there are various types of attacks that exploit vulnerabilities to achieve their goals, such as phishing [4], XSS [5], SQL injection [6], adversarial attack [7], etc. An instance of knowledge in cyberspace security can be represented as *(SQL injection, exploit, CVE-XXX)*.

While knowledge graph can represent facts clearly [8], most of these facts are static and not easily changed. In the real world, dynamic facts typically dominate and may be distorted by the passage of time and changes in space. The complexity of dynamic knowledge is not only reflected in social or public opinion network scenarios, but is particularly prominent in cyberspace security [9]. Due to the classical structure of triples, knowledge graph is not competent for dynamic knowledge representation. In order to make up this deficiency, many researchers focus on temporal knowledge graph. However, temporal information and spatial information are highly correlated, the lack of spatial information can seriously affect the authenticity and validity of knowledge. In order to solve this problem, the MDATA (**M**ulti-dimensional **D**ata **A**ssociation and in**T**elligent **A**nalysis) model [10] is proposed. MDATA redefines the overall architecture of knowledge representation and associates spatio-temporal information with knowledge, which can effectively solve the representation problem of dynamic knowledge.

The computability of knowledge is the key to effectively utilize potential information in knowledge [11], so we need to embed MDATA model into vector space. However, there is nearly no embedding method based on MDATA model. Moreover,

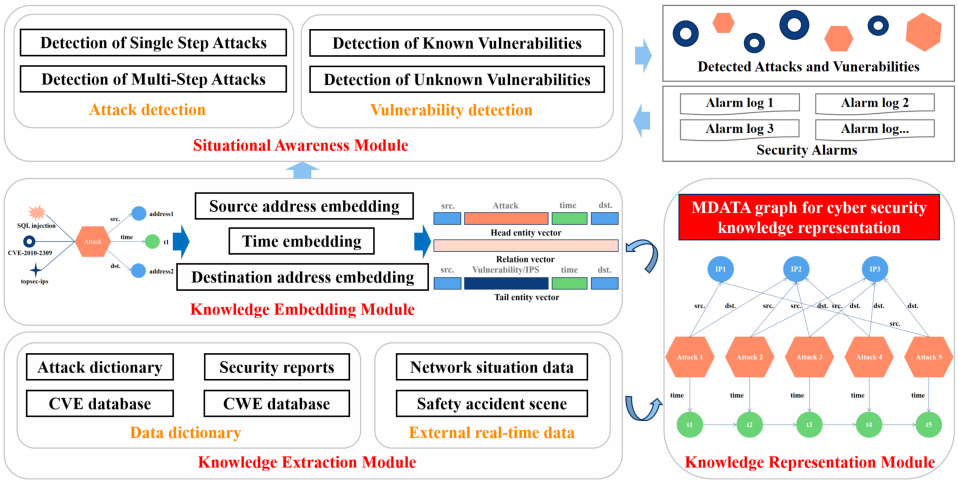

Fig. 1 An overview of the TSEE framework

most of the embedding methods for dynamic knowledge fail to make full use of temporal information. In order to improve and extend the existing embedding methods, Goel etc. [12] discover the problem and take the timestamp information as a part of entity vector and change knowledge structure at the vector space, achieving the utilization of temporal information. However, the existing embedding methods still have great defects in the integrity of knowledge expression due to ignoring spatial information. As a result, they are not suitable for embedding the MDATA model.

In this paper, we propose an advanced cyberspace security knowledge embedding framework based on the MDATA model. Since this is the first work about **T**emporal and **S**patial information **E**mbedding for **E**ntity, we name the framework TSEE. The intuitive idea is to map the cyberspace security knowledge to MDATA graph and apply the embedded model to detect the attacks and vulnerabilities by knowledge computation. The TSEE framework contains the following modules as depicted in Fig. 1. First, *the knowledge extraction module* extracts various kinds of cyberspace security knowledge from exposed repositories and external data. *The knowledge representation module* converts the obtained knowledge to MDATA graph. *The knowledge embedding module* is the core of the framework, which embeds the spatio-temporal information into the entity vector through an embedding layer. Finally, *the situational awareness module* can respond to dangerous attacks and potential vulnerabilities through embedding calculation.

We implement the proposed framework on the Peng Cheng Cyber Range. Since our framework is based on the MDATA model, we extensively evaluate the embedding of MDATA on link prediction and compare the performance with knowledge graph embedding methods in the same scenario. The result shows that TSEE outperforms other knowledge graph embedding methods. Notably, we observe that the training time overhead only increases linearly, despite the more diverse embedded structure. In addition, TSEE can make the embedding model quickly converge to a stable state.

Our specific contributions are summarized as follows:

- This is the first embedding work based on the MDATA model, which can effectively embed dynamic knowledge. We apply the embedded MDATA graph to the cyberspace security situational awareness system, which improves the performance of attack and vulnerability detection, and also enhances the ability of intrusion detection (prevention) system (IDS/IPS) to deal with risks;
- The proposed TSEE framework potentially integrates spatio-temporal features into cyberspace security knowledge through a nonlinear function. Further, we extend the embedding of MDATA graph to link prediction, which confirms the capability of TSEE in the cyberspace security situational awareness system;
- We conduct extensive experiments with TSEE on the Peng Cheng Cyber Range. After evaluation, test and verification, TSEE shows better results than previous knowledge graph embedding methods in terms of prediction accuracy, efficiency, and extendability.

The remainder of our paper is organized as follows. The next section focuses on the related work about knowledge representation and knowledge graph embedding. Preliminaries are provided in Section 3. We describe our TSEE framework in detail in Section 4, which can be used in cyberspace security such as situational awareness. In Section 5, we show the experimental results and analysis on the Peng Cheng Cyber Range. Finally, we conclude the paper in Section 6.

2 Related Work

In this section, we introduce related work from three aspects: knowledge representation, static KG embedding, and temporal KG embedding.

2.1 Knowledge Representation

Knowledge representation is a method of representing human knowledge in a certain form, so that computers can understand and use this knowledge. Common knowledge representation methods include logical representation, production rule representation, frame representation, semantic network representation, knowledge graph (KG), etc. These methods are suitable for different types of knowledge and application scenarios.

Logical representation [13, 14] uses predicates or functions to represent knowledge, which support natural language conversion and complex logical reasoning. However, the method is difficult to handle uncertainty and fuzziness when dealing with professional knowledge, making reasoning accuracy and efficiency low. **Production rule representation** [15] is also called IF-THEN representation. IF is the premise of the rule, and THEN indicates the conclusion or action that should be taken when the condition indicated by IF is satisfied. In general, rule representation can clearly reflect the interaction mechanism between knowledge, but it is difficult to express structured knowledge. **Frame representation** [16, 17] solves this problem. However, due to the formalization of the framework structure, procedural knowledge is difficult to express, which makes reasoning difficult. **Semantic network representation** [18] is composed of nodes and a number of directed arcs, where nodes represent various things, concepts, events, actions, etc. Directed arcs between nodes represent the semantic

relations between nodes, which can describe the relation between objects naturally. Unfortunately, due to complex network structure and unclear expression semantics, it is difficult to achieve reasoning under effective rules. **Knowledge graph** [3] is a semantic network in the form of graph, and represent knowledge as triples. Knowledge graph supports the storage of semi-structured data, and is more flexible in the expression pattern. Besides, it has great advantages in the correlation of related relations and performs well in knowledge reasoning. **MDATA** [10] is a multi-dimensional data association and intelligent analysis model based on knowledge graph, which can represent rich feature information such as time and space. MDATA supports more complex professional knowledge scenarios.

2.2 Static KG Embedding

Static KG contains knowledge that cannot easily be changed as time passes. Static KG embedding (KGE) mainly uses scoring function to measure the representation effect of knowledge, so as to learn the embedding of entities and relations for static knowledge. The models for KGE can be classified into the following three categories.

Translation Models. **TransE** [19] is the first KGE model based on translation inspired by the phenomenon of translation invariance. The model transforms head entity vector $\mathbf{h}$ into tail entity vector $\mathbf{t}$ through relation vector $\mathbf{r}$, and finally each piece of knowledge can be modeled as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ in vector space. Due to less computation and parameters, TransE achieves good results in *1-1* relation modeling. However, complex relation types such as *1-N*, *N-1*, and *N-N* cannot be modeled well using TransE. To solve this problem, **TransH** [20] projects head and tail entities to the hyperplane of relation, so that the same entity has different representations in different relations, and different entities can also have equivalent semantics in the same relation. **TransR** [21] thinks that entities have multiple attributes and may emphasize different attributes in forming relations. To capture this complexity, relations are divided into separate semantic spaces, each focusing on a specific set of attributes. **TransD** [22] holds that considering both the type of relation and the interaction between head and tail entities is crucial. To this end, it defines separate projection matrices for the head and tail entities in relation space, which is shown to be effective in improving the performance of KGE models.

Bilinear Models. **Rescal** [23] is the first bilinear model. It solves the problem of metaphorical information interaction between head and tail entities by representing each relation as a matrix. Although it enhances the expressive force of the model, the huge parameters become the burden of the model. **DistMult** [24] restricts the relation matrix to diagonal matrix, reducing the parameters of the model by half. Unfortunately, the model simultaneously loses the ability to model asymmetric relations. In order to remedy this defect, **ComplEx** [25] introduces the concept of complex-valued vector and realizes the modeling of asymmetric relation through complex-valued dot product. **SimplE** [26] initializes two vectors for each entity, and uses the inverse of the relation to cleverly set the symmetry term in the scoring function, so that the two vectors learn from each other. **RotatE** [27] represents relation vectors as rotation matrices and uses a bilinear function to calculate the dot product between head (tail) entity vector and relation vector. The dot product is then used as input to a

rotation function, which produces a new entity vector. In this way, *N-N* relations and symmetric relations can be effectively captured by rotation.

Neural Network Models. Numerous KGE methods that rely on neural networks have been introduced in the past few years. **ConvE** [28] uses 2D convolutional neural network to model the interaction between entity and relation. It first reshapes and concatenates the embeddings of head entity and relation from a triple, and then captures their features through the convolutional layer. Finally, the tail entity can be predicted through fully connected layers. However, due to ignoring interactions with tail entity, ConvE fails to capture the full integrity of knowledge. To solve this problem, **ConvKB** [29] eliminates the reshape operation in ConvE, and uses 1D convolution to capture feature interactions between entities that share the same dimension. By doing so, ConvKB can model deeper feature interactions. **ConvR** [30] also recognizes the insufficient interaction in ConvE. Considering this problem, ConvR enables convolution to be performed across entity representations by constructing filters adaptively from relation representations, thus improving the quality of embedding learning. **Conv3D** [31] extends ConvE and ConvKB. Specifically, the model reshapes the embedding of entity and relation, and then stacks them into a cube for 3D convolutional input instead of concatenation. Finally, a feature map can be generated.

R-GCN [32] introduces graph convolutional network (GCN) for modeling multi-relational data of KG. R-GCN represents each node by its neighbor nodes, and the convolution works on nodes of different relation types by weight sharing, enabling the capture of complex relations between nodes. However, when the quantity of relations becomes too large to manage, computation may suffer from overload. To solve this problem, **CompGCN** [33] uses a vectorized relation encoder to encode relations, including three different directions: positive, negative, and self-loops. Through polynomial parameterization in the encoder, high-order relation can be better handled. **KBGAT** [34] employs graph attention mechanism to adaptively capture the complex semantic information between entities and relations in KG, thereby improving the performance of knowledge reasoning. **ParamE** [35] points out that although neural network has powerful fitting capability, it ignores the translational nature of knowledge representation. Therefore, ParamE uses head entity vector as the input of neural network and the parameters of neural network as relation vector. After transformation, tail entity vector is finally output. **CapsE** [36] adopts capsule network to model relation triples, using 1D convolution and multiple filters to generate capsules, while modeling entities in a triple at the same dimension. **InteractE** [37] believes that the level of interaction between knowledge embedding vectors is insufficient, which greatly limits the representation ability of the current model. Therefore, InteractE adopts three methods named feature permutation, feature reshaping and circular convolution and improve the effectiveness of embeddings.

2.3 Temporal KG Embedding

Temporal KG enhances static KG with temporal information, enabling the description of the evolution of entities and relations over time. Temporal KG embedding enhances the effectiveness of static knowledge representation learning by modeling temporal

interactions. In recent years, embedding methods on temporal KG have appeared to describe knowledge more accurately.

TransE-TAE [38] extracts pairs of relations from large factual data and uses them to learn an evolution matrix over time. This matrix captures the objective occurrence sequence between relations and enables to understand how relations transform from prior to subsequent states. For example, *graduation* typically occurs before *work* for the average person. **HyTE** [39] maps static knowledge to corresponding time hyperplane, and finally obtains the embedded vector through a training method similar to TransE, thereby modeling the temporal changes of knowledge. **Context-Aware** [40]) uses fact-aware mechanism to determine the useful context for target fact by measuring time consistency with selected context to embed knowledge. **RE-NET** [41] models event sequences via neighborhood aggregator R-GCN [32] and event encoder RNN. R-GCN aggregates neighbor information of nodes in different time by GCN and attention mechanism, and RNN learns to capture global graph feature information at specific timestamp. **RE-GCN** [42] thinks that RE-NET neglects the structural dependencies that exist within KGs at different timestamps, as well as the static properties of entities. Therefore, the evolution units are set to model the KG sequence recurrently. To capture sequential patterns, RE-GCN uses gate recurrent components to model the historical KG sequence and incorporates static entity properties via a static graph constraint. This improves the representation of evolutional entities and relations. Given that most models treat temporal features as external information, **DE** [12] changes the structure of vector through a nonlinear entity embedding function, which incorporates time information into entity vector and achieves remarkable effect in the task of link prediction.

Most of knowledge representation methods can only represent static or temporal knowledge. Moreover, when dealing with professional knowledge in highly evolving fields, these methods cannot simulate human thinking to achieve association and analysis. MDATA solves the problem for dynamic knowledge representation, especially in cyberspace security. However, the existing embedding methods are mostly based on knowledge graphs and have great shortcomings in dynamic knowledge modeling. Since there is no embedding work for MDATA, we propose the new framework TSEE for embedding MDATA model.

3 Preliminaries

In this section, we introduce the definitions and notations in the paper. In addition, we formulate the problem of cyberspace security knowledge embedding and alarm detection formally.

3.1 System model

Consider a large cyberspace security situational awareness system G = (V, E) where each node $v \in V$ represents an attack, vulnerability, or detection (prevention) device, and each edge (v_i, v_j) represents that two nodes v_i and v_j are linked. If two nodes are connected, it implies the attacker may exploit the vulnerability to attack or the particular attack can be detected by IDS/IPS devices. A typical graphic of the

Fig. 2 A graphic of the cyberspace security situational awareness system

system is illustrated in Fig. 2. For each node, denote the attack on node v_i as $A_G(v_i) = \{a_1(v_i), a_2(v_i), ..., a_n(v_i)\}$ where each $a_k(v_i) \in A_G(v_i)$ represents a specific attack such as SQL injection. Since many attacks exploit the vulnerabilities or weaknesses of specific networked system, we denote the set of vulnerabilities and weaknesses as $W_G(v_i) = \{\omega_1(v_i), \omega_2(v_i), ..., \omega_m(v_i)\}$ where each $\omega_k(v_i) \in W_G(v_i)$ are detected by vulnerability scanning. IDS/IPS are devices that have the ability to perceive attacks, they are denoted as $D_G(v_i) = \{d_1(v_i), d_2(v_i), ..., d_n(v_i)\}$.

The goal of the system is to perceive effective attacks or potential vulnerabilities at any time and place. Normally, the attack is composed of multiple attack steps to hide itself, such as APT (advanced persistent threat) attacks. Therefore, we can treat multi-step attack as a sequence and divide them into multiple single-step attacks. We denote the complex attack as a sequence $S = \{s_1, s_2, ..., s_t\}$, where each step s_k is regarded as a specific attack step, such as phishing mail attack, trojan horse attack, etc. Each step related to the attack can be represented as two types of tuples referring to MDATA:

$$\begin{aligned} s_k(1) &= (a_k(v_i),\ exploit,\ \omega_k(v_i),\ T,\ S) \\ s_k(2) &= (a_k(v_i),\ instanceOf,\ d_k(v_i),\ T,\ S) \end{aligned} \tag{1}$$

where T denotes the denote the time of the attack, S denotes the collection of source and destination IP.

Every single-step attack can be described as above. For example, the attacker carries out brute force cracking attack from one server (assume $IP_1 = 10.XX.XX.7$) to another (assume $IP_2 = 10.XX.XX.8$) at $TIME_1$. Meanwhile, the attack is detected by

the device in the system. All of these security knowledge can be expressed as follows:

$$\begin{aligned} s_k(1) &= (brute\ force\ cracking,\ exploit,\ CVE\text{-}XXX,\ TIME_1,\ IP_1,\ IP_2) \\ s_k(2) &= (brute\ force\ cracking,\ instanceOf,\ topsec\text{-}ips,\ TIME_1,\ IP_1,\ IP_2) \end{aligned} \tag{2}$$

The system usually deploys and uses IDS/IPS to detect attacks, and finally store them into alarm logs to provide service. Normally, alarm logs record sufficient text information such as attack time, source IP, destination IP, destination port, attack description, etc. Although numerous security alarms assist cyberspace security situation awareness every year, the core tasks such as attack detection and vulnerability detection still face great challenges due to alarm missing and inaccuracy.

3.2 Problem definition

In the large cyberspace security situational awareness system G, we can obtain the vulnerabilities set W_G and intrusion detection (prevention) device set D_G. Devices can produce alarm logs, which contain various kinds of attacks and related vulnerabilities. We denote the alarm set as follows:

$$ALARM_G = \{alarm_1, alarm_2, ..., alarm_n\} \tag{3}$$

Our objective is to gain the ability to efficiently perceive attacks and vulnerabilities with embedding methods and apply it to the situational awareness system. We formulate the problem as:

Problem 1: A Temporal and Spatial knowledge Embedding operation:

$$((V, T, S), R) \rightarrow \psi, \tag{4}$$

is a function that maps every alarm $((v_h, v_t, t, s), r)$ in $ALARM_G$ into a hidden representation ψ, where V, T, and S represent the aggregate of entity, time, and space, respectively. ψ is the embedding vector corresponding to non-empty tuples, v_h and v_t are head entity and tail entity of the tuple. The problem is to detect potential attacks or vulnerabilities according to alarm data.

4 TSEE: Temporal and Spatial information Embedding framework for Entity

In this section, we describe the **T**emporal and **S**patial information **E**mbedding framework for **E**ntity (TSEE) for cyberspace security situational awareness in detail. First, we present an overview of the framework. Then we describe these modules in the framework. Finally, we discuss about the advantages and disadvantages of the proposed framework.

4.1 Overview of the framework

The framework is illustrated in Fig. 1, containing four main modules as follows.

- The knowledge extraction module is introduced to extract knowledge from various cyberspace security data sources, such as security reports, vulnerability databases, and threat intelligence;
- The knowledge representation module represents the extracted security knowledge as the MDATA graph, and converts the knowledge to a large-scale MDATA graph;
- The knowledge embedding module maps the entity and related dynamic attributes in MDATA graph to vector space, and simulates the dynamic process of knowledge changing in vector space;
- The situational awareness module associates alarm data with the extracted knowledge based on embedding of the MDATA graph, it is designed to detect the potential attacks and vulnerabilities through matching algorithm.

4.2 Knowledge Extraction Module

The emergence of new attack methods and vulnerabilities has led to an evolving trend in threat forms, which in turn drives continuous accumulation of cyberspace security knowledge. Moreover, the sources of cyberspace security knowledge can be diverse, including vulnerability database, virus database, threat intelligence, detection results of security vendors, security forums and security incident reports, etc. In order to solve this problem, many countries and institutions have built abundant knowledge bases to effectively manage knowledge of attacks and vulnerabilities. For example, the

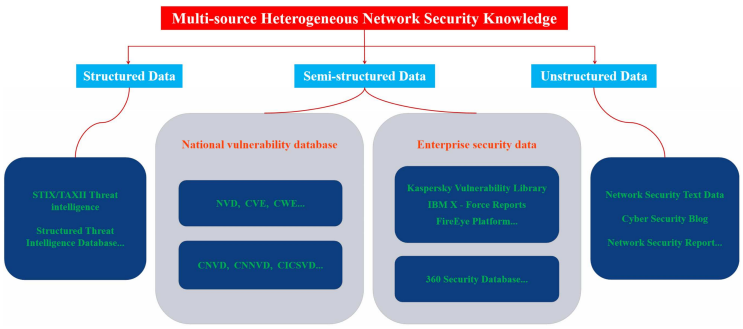

Fig. 3 The sources of cyberspace security knowledge

Cyber Threat Intelligence (CTI) program aims to collaborate and share intelligence information about cyberspace security threats to enhance defenses and help countries better respond to cyber threats. Security Incident Response Teams (SIRT) are security response teams established by multiple organizations, which typically collaborate with other security organizations and vendors to coordinate and share knowledge about vulnerabilities and security incidents.

The cyberspace security knowledge can be extracted or utilized from the multi-source heterogeneous sources. As is shown in Fig. 3, the Common Vulnerabilities and Exposures (CVE) database records the disclosed vulnerabilities and the Common Weakness Enumeration (CWE) database lists the weaknesses of softwares and

hardwares. In addition, many countries and enterprises have published many security reports on cyber attacks, from which knowledge such as attack process can be extracted. Furthermore, information about cyber attack and defense released by Security Forum also plays an important role in cyberspace security knowledge extraction. In addition, external real-time data are also extracted for further association and analysis, such as cyberspace security incident scene data and situational awareness system data in this module.

4.3 Knowledge Representation Module

The extracted cyberspace security knowledge is represented by MDATA model, which can express the spatio-temporal characteristics. Then, the represented knowledge is converted to MDATA graph, so as to simulate the dynamic changes in cyberspace security. MDATA graph is composed of nodes (entities) and edges (relations) that

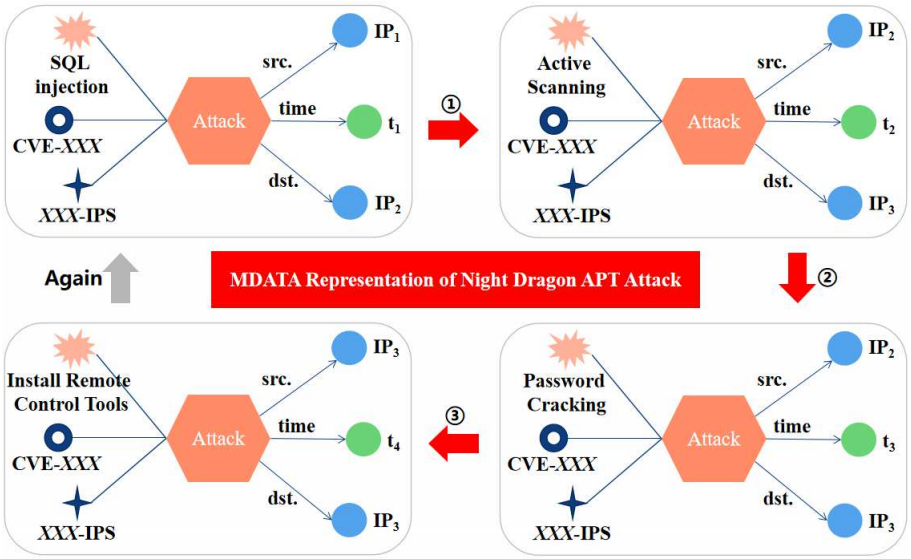

Fig. 4 A graphic of the MDATA Representation of Night Dragon APT Attack

can be obtained through Natural Language Processing (NLP) methods such as entity extraction, relation extraction, etc.

The MDATA graph of typical Night Dragon APT attack is shown in Fig. 4. From the graphic for Night Dragon, multi-step attacks are included, each of which exploits a specific vulnerability. First, the attacker (assume IP_1) utilizes SQL injection to invade a web server (assume IP_2) at t_1. Then the attacker utilizes active scanning to find the specific fragile host (assume IP_3) at t_2, and utilizes password cracking to control the host at t_3. Finally, the sensitive files will be transmitted to the attacker at t_4 by installing remote control tools.

Many attacks can be showed in more detail by adding the occurrence time of the single-step attack. Moreover, the cyber attacks usually be launched through different

IP addresses. Through these IP addresses, we can restore and simulate the real occurrence scene. After being represented by MDATA graph, the fine-grained attack scene of Night Dragon APT attack is fully displayed.

4.4 Knowledge Embedding Module

To enable the computability of cyberspace security knowledge, we design a universal function to embed spatio-temporal features into entities, and use it for knowledge embedding in the MDATA graph. Normally, we take entities (mainly about attacks, vulnerabilities, devices, etc.), time, source IP address, destination IP address as input, and obtain multi-dimensional entity vectors with rich spatio-temporal features. Considering the sparsity of relations in cyberspace security, we do not add features to relation vectors. We also provide optional parameters for the embedding function to ensure that the method can adapt to scenarios similar to cyberspace security. The general definition of our function is as follows:

$$E_v^{t,s}[n] = \begin{cases} \boldsymbol{\alpha_v}[n]\sigma_s\left(\boldsymbol{\omega}_v[n]s_{src} + \boldsymbol{b}_v[n]\right), & \text{if } 0 \le \mathrm{n} \le pd \\ \boldsymbol{\alpha_v}[n], & \text{if } pd \le \mathrm{n} \le qd \\ \boldsymbol{\alpha_v}[n]\sigma_t\left(\boldsymbol{\omega}_v[n]t + \boldsymbol{b}_v[n]\right), & \text{if } qd \le \mathrm{n} \le (1-p)d \\ \boldsymbol{\alpha_v}[n]\sigma_s\left(\boldsymbol{\omega}_v[n]s_{dst} + \boldsymbol{b}_v[n]\right), & \text{if } (1-p)d \le \mathrm{n} \le d \end{cases} \tag{5}$$

where $\boldsymbol{\alpha_v}$, $\boldsymbol{\omega_v}$, $\boldsymbol{b_v} \in R^n$ are the entity-related vector, α_v is used to capture the global features of the entity, $\omega_v s$ and $b_v s$ is the influence factor of spatio-temporal characteristics to balance the needs of scenes. They are parameters that can be updated by learning. σ_t and σ_s are activation functions about time and space respectively. p and q are responsible for regulating the proportion of features in entity vectors, where $0 \le p \le q \le 1$. Specifically, in the entity vector of n-dimensional, [pd, qd] is used to preserve the original features of the entity, [qd, (1-p)d] is used to capture temporal features, [0, p] and [(1-p)d, d] are used to capture source IP address and destination IP address features. $0.5 \le q \le 1-2p$ needs to be satisfied to ensure that spatio-temporal information does not occupy the main part of the entity vector. The original features of entity will be fully expressed when some elements of ω_v are set to zero, and then the embedding method degenerates into a static model.

Normally, we associate alarm knowledge and extracted knowledge (such as CVE database) represented by MDATA graph into vector space with formula 5. In the process of model training, we use mini-batch stochastic gradient descent (SGD) to learn. For each alarm knowledge $alarm_k = ((\mathrm{v_h}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r})$ in mini-batch, we generate two types of negative samples by replacing head entity or tail entity, 1 - $((\mathrm{v}_{h'}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r})$ and 2 - $((\mathrm{v_h}, \mathrm{v}_{t'}, \mathrm{t}, \mathrm{s}), \mathrm{r})$, where $\mathrm{v}_{h'}$ denotes the replaced head entity and $\mathrm{v}_{t'}$ denotes the replaced tail entity.. The related parameters of the model are updated by minimizing the cross entropy loss:

$$\mathcal{L} = -\left(\sum_{alram_k \in ALARM_G} \frac{\exp(\phi((\mathrm{v_h}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r}))}{\sum_{\mathrm{v}_{h'} \in \mathrm{C}_1} \exp(\phi((\mathrm{v}_{h'}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r}))} + \frac{\exp(\phi((\mathrm{v_h}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r}))}{\sum_{\mathrm{v}_{t'} \in \mathrm{C}_2} \exp(\phi((\mathrm{v_h}, \mathrm{v}_{t'}, \mathrm{t}, \mathrm{s}), \mathrm{r}))}\right) \tag{6}$$

Algorithm 1 The Learning Algorithm of TSEE based on Attack and Vulnerability Detection.

Require:
Alarm knowledge set $Alarm_G$ = {(V, T, S), R} in MDATA graph, related knowledge bases, embedding dimension d, learning rate α, epoch number n;

Ensure:
Mixed entity embeddings with spatio-temporal features and relation embeddings in MDATA graph;

Generate entity set E and relation set R;
epoch $\leftarrow$ 0;
while epoch $<$ n **do**
  // sample a mini-batch of size b from $Alarm_G$
  $Alarm_{G\_batch} \leftarrow Sample(Alarm_G, b)$;
  $T_{batch} \leftarrow \emptyset$; // tuples for learning
  for each tuple in $Alarm_{G\_batch}$ **do**
    // negative sampling from E with unif
    $(C_1, C_2) \leftarrow$ negSample (tuple);
    // set of tuples obtained by negative sampling for learning
    $T_{batch} \cup (C_1, C_2)$;
    Initialize embeddings of entity and relation;
    Implement segmented embedding for entities w.r.t formula 5;
    xavier_uniform (entity embeddings);
    xavier_uniform (relation embeddings);
  end for
  Update parameters with optimizer w.r.t formula 6;
end while
return Embeddings of entities and relations of MDATA graph;

where C_1 and C_2 are entity sets obtained by negative sampling head entity and tail entity from $alarm_k$, the size of the set is determined by the ratio of negative samples to positive samples. ϕ is a scoring function for tuples. For example, if TransE[19] is adopted, the function will be defined as $\phi((v_h, v_t, t, s), r) = \|v_h(ts) + r - v_t(ts)\|_{l_1/l_2}$, where $v_h(ts)$ and $v_t(ts)$ are the vectors of entities obtained by formula 5).

Similarly, other static KG embedding models can also be extended with our embedding framework for scoring and prediction, such as DistMult [24], SimplE [26], etc. We correspondingly name the combined methods TSEE-TransE, TSEE-DistMult, and TSEE-SimplE. Obviously, our embedding framework has good extendability and can be applied in many scenes in cyberspace security compared with previous single embedding methods. The algorithm of gaining knowledge embeddings of MDATA graph is shown in Alg. 1.

This module adds spatio-temporal information into entity vector based on MDATA graph. Compared with previous embedding methods, our method can achieve more accurate knowledge representation in a low-dimensional vector space.

4.5 Situational Awareness Module

After transforming the MDATA graph of cyberspace security knowledge into embedded vectors, the situational awareness module aims to detect and predict potential cyber attacks and vulnerabilities through calculations. Due to maintaining rich knowledge embeddings, the detection will be executed quickly. Finally, the potential threat information can be predicted and recorded in new alarm logs.

Algorithm 2 The Matching Algorithm based on the Situational Awareness System.

Require:
Alarm knowledge set $Alarm_G = \{(\mathrm{V}, \mathrm{T}, \mathrm{S}), \mathrm{R}\}$, Embedding dictionary Dic_E of entities in the situational awareness system;

Ensure:
Potential threat information;

1: **for** each tuple in $Alarm_G$ **do**
2: // replacing entity embeddings in tuple with Dic_E
3: $(Alarm_{G'}) \leftarrow$ sampleEntiyEmbeddings (tuple);
4: // caculating the scores of these tuples
5: Score all tuples in $Alarm_{G'}$ with ϕ in formula 6;
6: // verifying the tuples on the Peng Cheng Cyber Range
7: Evaluate the rationality of top-tuples;
8: // recording potential attacks and vulnerabilities
9: Write the probable threat information to new alarm logs;
10: **end for**

Many matching methods can be applied in this module. In this part, we introduce an efficient matching algorithm as Alg. 2. For each tuple in $Alarm_G$, we first initialize a set to store the restructured tuples that are replaced by the embedding dictionary. After obtaining the new tuple set, we score and rank the scores of all tuples in order. Finally, we evaluate the rationality of top-tuples on the Peng Cheng Cyber Range. By repeating the process, the algorithm outputs potential threat information to new alarm logs. Since MDATA model is associated with spatio-temporal features, this module can perceive possible attacks and vulnerabilities.

4.6 Advantages and Disadvantages

The proposed embedding framework can effectively detect the potential attacks and vulnerabilities based on the embeddings of MDATA graph. Compared to the extant methods in cyberspace security, the framework has the following advantages:

- The capability of detecting complex attacks. Many existing methods can only find out some specific attacks for lack of cyberspace security knowledge sources. The proposed framework can associate alarm data by performing correlation analysis on multiple security data sources, thereby providing a wider range of potential detection results;

- The detection accuracy of attacks and vulnerabilities. Current KG methods can only be utilized for analyzing static security knowledge, which results in relatively low accuracy of detection and many instances of missing or false cases. The proposed framework based on MDATA model can record spatio-temporal knowledge, contributing valid detection for the situational awareness system;
- The extendability of embedding methods. The classical embedding methods can be extended by TSEE to evaluate the knowledge embedding and predict potential entities, thereby enabling adaptation to sophisticated scenarios in cyberspace security.

Although the framework has good performance in cyberspace security situational awareness, there are also some disadvantages. First, the framework uses knowledge vector computation in threat detection. However, it is difficult for our framework to find out unknown threat that is not contained in the extracted knowledge. Second, the update iteration of cyber attack technologies leads to a large number of missing knowledge in cyberspace security knowledge base, and the volume of knowledge in computation will become huge. Therefore, it is necessary to establish a comprehensive knowledge base maintenance mechanism to reduce the impact of knowledge gaps. Furthermore, improving the framework's ability to detect unknown knowledge can be achieved by implementing automatic knowledge analysis and reasoning in future work.

5 Experiment results

In this section, we evaluate the proposed framework from four aspects. First, we evaluate the accuracy and extendability of cyberspace security knowledge embedding method based on MDATA graph. Second, we evaluate the efficiency and performance of the framework. Then we evaluate the detection effect of the framework with different activation functions. Finally, we demonstrate the importance of each component in the embedding method through ablation experiments.

5.1 Experiment Setting

The data is collected through the Peng Cheng Cyber Range, which supports many attack and defense drills. We collect the data within one hour during one drill and extracted about 6200 alarms with spatio-temporal characteristics (including 524 entities). After extracting the facts from these alarms, we divide them into three parts as 8:1:1. We implement our experiments in Pytorch and use Adam [43] to optimize parameters in mini-batch while learning embeddings. The hyperparameters are listed as follows: the embedding dimension $d \in \{100, 150, 200\}$, the proportion of static feature dimensions $no\_st\_prop \in \{0.25, 0.35, 0.45\}$, the proportion of spatial feature dimensions $space\_ent\_prop \in \{0.2, 0.3, 0.4\}$, the proportion of temporal feature dimensions $time\_ent\_prop = d - no\_st\_prop - space\_ent\_prop$, number of training rounds $epochs \in \{300, 400, 500\}$, ratio of negative and positive samples $negative\ ratio \in \{100, 300, 500\}$, the learning rate $lr \in \{0.1, 0.01, 0.001\}$, The finally adopted settings: $\{d = 100$, $no\_st\_prop = 0.25$, $space\_ent\_prop = 0.3$, $time\_ent\_prop = 0.45$, $epochs = 500$, $negative\ ratio = 500$, $lr = 0.001\}$. The batch size $b = 512$ and the l_2 norm is set

in the score function. Various types of activation functions are used in equation 1 (see more specifically in Table 2). The model was set to be saved every 20 epochs, and the model obtaining the optimal MRR on validation set is selected for test. In order to ensure the fairness of the experimental results, the same parameter settings are used in baselines based on alarm data.

5.1.1 Baselines

Our benchmarks contain the classical static KG embedding models, including translation models (such as TransE, TransH, and TransR), bilinear models (such as Rescal, DistMult, ComplEx, and SimplE) and neural network models (such as ConvKB, ConvE, and Conv3D). For temporal KG embedding models, HyTE, RE-NET, and RE-GCN are selected for comparison. DE is the main compared model because it also adopts the segmented embedding.

5.1.2 Link Prediction

Link prediction is usually used in KG to predict the missing fact. Specifically, the task is defined to predict missing head entity in the fact $((?, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r})$ or missing tail entity in the fact $((\mathrm{v_h}, ?, \mathrm{t}, \mathrm{s}), \mathrm{r})$. Through the accuracy of hitting the correct facts in the embedding space, the embedding performance can be measured. We use link prediction to indicate the effectiveness of cyberspace security knowledge embedding in MDATA graph.

Since the significant difference in the quantity of head and tail entities in cyberspace security scenario, we use $\mathrm{v_k} \in V$ to replace head entity $\mathrm{v_h}$ and tail entity $\mathrm{v_t}$ separately for each fact $((\mathrm{v_h}, \mathrm{v_t}, \mathrm{t}, \mathrm{s}), \mathrm{r})$ in the test set of alarm data, generating two query dictionaries to calculate the metrics in Part 5.1.3. This approach differs from the common practice in KG, which randomly replaces either the head or tail entities to generate a single query dictionary.

5.1.3 Evaluation Metrics

Two popular indicators are usually adopted in link prediction, one is *mean reciprocal rank (MRR)* and the other is $Hits@n$, which reflect the comprehensive embedding performance. In the case of generating only one query dictionary, two indicators are usually defined as:

$$MRR = \frac{1}{|\ \mathrm{test}\ |} \sum_{\mathrm{fact}\ \in\ \mathrm{test}} \frac{1}{\mathrm{rank}(\mathrm{fact},\ \mathrm{v_k})}; \tag{7}$$

$$Hits@n = \frac{1}{|\ \mathrm{test}\ |} \sum_{\mathrm{fact}\ \in\ \mathrm{test}} \mathrm{bool}(\mathrm{rank}(\mathrm{fact}\ , \mathrm{v_k}) \leq n), \tag{8}$$

where $rank(fact, \mathrm{v_k})$ represents the ranking of the fact after randomly replacing head or tail entity.

In order to ensure the rationality of experimental comparison, we make following modifications on the above two evaluation indicators:

$$MRR = \frac{1}{2 \mid \text{ test } \mid} \sum_{\text{fact } \in \text{ test}} (\frac{1}{\text{rank(fact, } \mathrm{v_h})} + \frac{1}{\text{rank(fact, } \mathrm{v_t})}); \tag{9}$$

$$\begin{aligned} Hits@n = \frac{1}{2 \mid \text{ test } \mid} \sum_{\text{fact } \in \text{ test}} (&\text{bool(rank(fact , } \mathrm{v_h}) \\ &\leq n) + \text{bool(rank(fact, } \mathrm{v_t}) \leq n)), \end{aligned} \tag{10}$$

where $rank(fact, \mathrm{v_h})$ represents the ranking of the fact after matching the head entity, $rank(fact, \mathrm{v_t})$ represents the ranking of the fact after matching the tail entity.

5.2 Accuracy Analysis of TSEE

We compare three variants extended by our framework with the baselines: 1-TSEE-TransE, 2-TSEE-DistMult, and 3-TSEE-SimplE. Since we focus on the embedding structure and representation capability, we train all the models with the same setting and dataset for common parameters. The best results of MRR and Hits@10 for each model are reported in Table 1.

Table 1 Comparison of Embedding Accuracy

Model	MRR	Hits@1	Hits@3	Hits@10
TransE [19]	0.0911	–	0.0968	0.3226
TransH [20]	0.0875	–	0.1048	0.2581
TransR [21]	<u>0.1718</u>	0.1290	0.1935	0.2419
Rescal [23]	0.0844	0.0484	0.0806	0.1613
DistMult [24]	0.0757	0.0323	0.0887	0.1452
ComplEx [25]	0.0189	0.0081	0.0161	0.0161
SimplE [26]	0.0506	0.0161	0.0565	0.1129
ConvKB [29]	**0.2010**	0.0968	<u>0.2661</u>	0.4234
ConvE [28]	0.1856	**0.1563**	0.1423	0.3861
Conv3D [31]	0.1619	0.0847	0.1533	0.3912
HyTE [39]	0.0758	0.1064	0.2627	0.4022
RE-NET [41]	0.1141	0.1263	0.2541	0.3965
RE-GCN [42]	0.0588	<u>0.1324</u>	**0.2794**	0.4021
DE-TransE [12]	0.1367	0.0323	0.1532	0.4113
DE-DistMult [12]	0.1321	0.0403	0.1532	0.3065
DE-SimplE [12]	0.0688	–	0.0645	0.2258
TSEE-TransE	0.1324	0.0242	0.1371	<u>0.4274</u>
TSEE-DistMult	0.1581	0.0538	0.1801	**0.4328**
TSEE-SimplE	0.0339	0.0081	0.0323	0.0887

When static KG embedding methods are applied to cyberspace security, convolutional neural network models exhibit best accuracy, followed by translation models, and lastly the bilinear models. For translation models, TransH and TransR have lower performance than TransE when the value of n in hits@n is too large. The reason is that

the number of relations is far less than that of entities in cyberspace security, while TransH and TransR mainly focus on relations modeling, making the model lose its original advantages. For bilinear models, Rescal outperforms better than other models for that the interaction of all pairs of entities can be fully captured. DistMult is the simplified model of Rescal, which experiences a slight decrease in accuracy. The performance of ComplEx is very poor because the introduction of complex-valued vectors significantly increases the number of parameters, making the complexity of embedding space and difficulty in optimization. The embedding ability of SimplE is affected as seen in the table due to the sparsity of relations in cyberspace security. For convolution neural network models, ConvKB, ConvE, and Conv3D achieve significant improvement in extracting feature information of static knowledge, which is closely tied to inherent advantages in interaction between entity and relation through filters.

The overall performance of temporal KG embedding methods perform well. The embedding accuracy of HyTE is relatively better than other models, because HyTE can establish different hyperplanes for timestamps, leading to a better understanding and differentiation of entity and relation embeddings. RE-GCN and RE-NET can be effective compared to traditional neural network models even with a small n in hits@n, as they prioritize the global structural information of the graph and focus on the impact of neighbor nodes in the graph. TransE, DistMult, and SimplE extended by DE is not particularly outstanding than other models, a deeper reason is that only temporal information is integrated. The comprehensive performance of TSEE-TransE and TSEE-DistMult is better than DE and other baselines due to the addition of spatio-temporal feature information. Since our data is based on cyberspace security scenario with numerous attacks and vulnerabilities, the dependency of head and tail entities is weak, so SimplE extended by DE or TSEE is difficult to achieve satisfactory results.

In summary, TSEE has higher accuracy in embedding knowledge compared to other models, making it a valuable tool for detecting potential attacks and vulnerabilities in cyberspace security situation awareness.

5.3 Training Efficiency of TSEE

We compare three static embedding models, TransE, DistMult, and SimplE, with three variant models extended from DE and TSEE under the same experimental setting (including activation function). The curves of time and loss with the number of epochs are shown in Figure 5.

From the loss-epoch curve in the graphic, TSEE-TransE and TSEE-SimplE require less time to converge compared with other models. The convergence speed of TSEE-DistMult is slightly slower than that of DE-DistMult. Moreover, with the same loss function, TSEE can reduce the loss of embedding models to a relatively lower position. From the time-epoch curve in the graphic, TSEE only increases linearly in the cost of training time compared with the other two baselines.

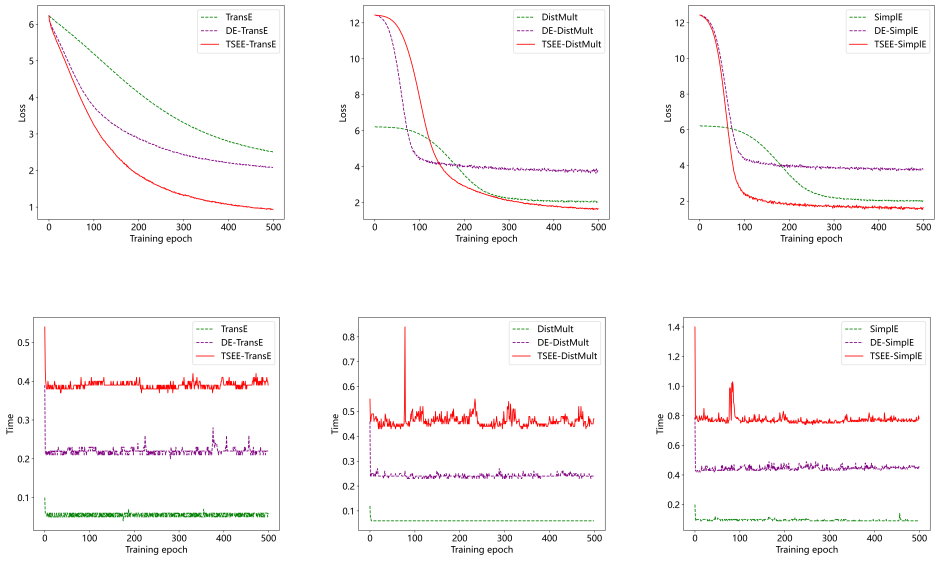

Fig. 5 The training curve for the three approaches.

5.4 Detection Effect with different Activation Functions

In this section, we explore the effect of different activation functions in Equation 5. *Sigmoid* has a smooth range from 0 to 1, which ensures that the neuron values will not jump during training. The output of *Tanh* is zero-centered, which is better than *Sigmoid*. *Leaky ReLU* is an improved version of *ReLU*, which can effectively solve the problem of zero-gradient caused by negative input of *ReLU*. *Sin* is an oscillating non-monotonic function, which can model various open and closed states. *SIREN* [44] is a sinusoidal periodic activation function, which shows good results on audio, video, and images restoring.

The influence of different activation functions for embedding cyberspace security knowledge cannot be ignored. From the observation in Table 2, *Sin* and *SIREN* can capture more complex feature information of entities in DE and TSEE, which is significantly superior to other activation functions.

5.5 Ablation Study

In Equation 5, the spatio-temporal information embedding contains three components: a_v, ω_v, and b_v. To learn the mechanism of them in detail, we run our embedding framework under three kinds of settings: 1- when a_v are removed (i.e. set to 1), 2- when ω_v are removed (i.e. set to 1), and 3- when b_v are removed (i.e. set to 0).

From the obtained results in Table 3, the absence of each component will cause a different degree of accuracy decline in the embedding performance of the model. It can be improved that all the three components are important and indispensable for capturing the spatio-temporal features.

Table 2 Embedding Performance with different Activation Functions

Model	σ_t	σ_s	MRR	Hits@1	Hits@3	Hits@10
TransE	-	-	0.0911	0.0000	0.0968	0.3226
DE-TransE	*Sigmoid*	–	0.1367	0.0323	**0.1532**	**0.4113**
	Tanh	–	0.1213	0.0081	0.1613	0.3871
	LeakyReLU	–	0.0311	0.0000	0.0242	0.0806
	Sin	-	0.1040	0.0081	0.1210	0.3226
	SIREN	-	0.1182	0.0108	0.1317	0.3952
TSEE-TransE	*Sigmoid*	*Sigmoid*	0.1143	0.0161	0.1048	0.3871
	Tanh	*Tanh*	0.0992	0.0081	0.0968	0.3387
	LeakyReLU	*LeakyReLU*	0.0380	0.0000	0.0081	0.1371
	Sin	*Sin*	0.1324	0.0242	**0.1371**	**0.4274**
	SIREN	*SIREN*	0.1032	0.0054	0.1129	0.3145
DistMult	-	-	0.0757	0.0323	0.0887	0.1452
DE-DistMult	*Sigmoid*	–	0.0586	0.0000	0.0484	0.1613
	Tanh	–	0.0677	0.0000	0.0968	0.1532
	LeakyReLU	–	0.1321	0.0403	**0.1532**	**0.3065**
	Sin	-	0.0605	0.0000	0.0242	0.2177
	SIREN	-	0.0654	0.0054	0.0699	0.1694
TSEE-DistMult	*Sigmoid*	*Sigmoid*	0.1458	0.0403	0.1694	0.3468
	Tanh	*Tanh*	0.1601	0.0565	0.1774	0.3871
	LeakyReLU	*LeakyReLU*	0.1531	0.0806	0.1532	0.3306
	Sin	*Sin*	0.1759	0.0806	**0.1855**	0.4194
	SIREN	*SIREN*	0.1581	0.0538	0.1801	**0.4328**
SimplE	-	-	0.0506	0.0161	0.0565	0.1129
DE-SimplE	*Sigmoid*	–	0.0307	0.0000	0.0403	0.0806
	Tanh	–	0.0294	0.0000	0.0323	0.0645
	LeakyReLU	–	0.0688	0.0000	0.0645	**0.2258**
	Sin	-	0.1062	0.0403	**0.1532**	0.2016
	SIREN	-	0.0570	0.0054	0.0699	0.1505
TSEE-SimplE	*Sigmoid*	*Sigmoid*	0.0264	0.0000	0.0321	0.0726
	Tanh	*Tanh*	0.0171	0.0000	0.0081	0.0484
	LeakyReLU	*LeakyReLU*	0.0225	0.0000	0.0242	0.0484
	Sin	*Sin*	0.0268	0.0000	**0.0323**	**0.0806**
	SIREN	*SIREN*	0.0227	0.0027	0.0162	0.0645

Table 3 Results for Different Variations

Model	a_n	ω_v	b_v	MRR	Hits@1	Hits@3	Hits@10
TSEE-TransE	×	✓	✓	0.0559	0.0000	0.0377	0.1720
	✓	×	✓	0.1277	0.0323	0.1183	0.3656
	×	✓	×	0.1104	0.0108	0.1291	0.3306
TSEE-DistMult	×	✓	✓	0.1695	0.0726	0.1801	0.4027
	✓	×	✓	0.1778	0.0860	0.1962	0.3844
	✓	✓	×	0.1686	0.0780	0.1828	0.4113
TSEE-SimplE	×	✓	✓	0.0199	0.0027	0.0134	0.0377
	✓	×	✓	0.0167	0.0000	0.0162	0.0511
	✓	✓	×	0.0196	0.0000	0.0134	0.0484

6 Conclusions

In this paper, we propose an embedding framework TSEE for cyberspace security based on MDATA, which can effectively detect attacks and vulnerabilities. The framework consists of four main modules. The knowledge extraction module extracts cyberspace security knowledge from various data sources and the latest security alarm data. The knowledge representation module converts obtained knowledge into MDATA graph to support efficient retrieval and association. The knowledge embedding module can embed MDATA graph into vector space to provide effective representation for knowledge computability. The situational awareness module can discover potential attacks and vulnerabilities in alarm data, including undetected attacks and vulnerabilities in IDS/IPS device. The framework has response capability and plays an important role in cyberspace security defense. We validate the framework on the Peng Cheng Cyber Range and conduct experiments to evaluate its performance. The proposed framework can greatly improve the cyberspace security defense capability of the situational awareness system considering the detection accuracy. Since the method has limitations for increasing knowledge and unseen knowledge, we will continue to optimize the framework and embedding algorithm in the future.

Author Contributions Angxiao Zhao wrote the main manuscript text. Zhaoquan Gu programed the key project and provided guidance in writing the manuscript. Yan Jia supervised the work. Wenying Feng assisted in editing and proofreading this manuscript. Jianye Yang and Yanchun Zhang provided suggestions for revisions. All authors reviewed the manuscript.

Funding This work was supported in part by the Major Key Project of PCL (Grant No. PCL2022A03), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

Availability of data and materials The data and materials used to support the findings of this study are available from the corresponding author upon request.

Declaration

Competing Interests We declare that none of the authors have any competing interests in the manuscript.

Ethics Approval Not applicable.

References

[1] Wei, S., Jia, Y., Gu, Z., Shafiq, M., Wang, L.: Extracting novel attack strategies for industrial cyber-physical systems based on cyber range. IEEE Systems Journal (2023)

[2] Davis, R., Shrobe, H., Szolovits, P.: What is a knowledge representation? AI magazine **14**(1), 17–17 (1993)

[3] Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., Wahler, A., Fensel, D., et al.: Introduction: what is a knowledge graph? Knowledge graphs: Methodology, tools and selected use cases, 1–10 (2020)

[4] Aleroud, A., Zhou, L.: Phishing environments, techniques, and countermeasures: A survey. Computers & Security **68**, 160–196 (2017)

[5] Gupta, S., Gupta, B.B.: Cross-site scripting (xss) attacks and defense mechanisms: classification and state-of-the-art. International Journal of System Assurance Engineering and Management **8**, 512–530 (2017)

[6] Halfond, W.G., Viegas, J., Orso, A., *et al.*: A classification of sql-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering, vol. 1, pp. 13–15 (2006). IEEE

[7] Zhaoquan, G., Yushun, X., Weixiong, H., Lihua, Y., Yi, H., Zhihong, T.: Marginal attacks of generating adversarial examples for spam filtering. Chinese Journal of Electronics **30**(4), 595–602 (2021)

[8] Jia, Y., Gu, Z., Jiang, Z., Gao, C., Yang, J.: Persistent graph stream summarization for real-time graph analytics. World Wide Web, 1–21 (2023)

[9] Jia, Y., Gu, Z., Du, L., Long, Y., Wang, Y., Li, J., Zhang, Y.: Artificial intelligence enabled cyber security defense for smart cities: A novel attack detection framework based on the mdata model. Knowledge-Based Systems **276**, 110781 (2023)

[10] Jia, Y., Gu, Z., Li, A., *et al.*: Mdata: A new knowledge representation model. Springer International Publishing. doi **10**, 978–3 (2021)

[11] Pandey, B., Mishra, R.: Knowledge and intelligent computing system in medicine. Computers in biology and medicine **39**(3), 215–230 (2009)

[12] Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3988–3995 (2020)

[13] Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM (JACM) **12**(1), 23–41 (1965)

[14] Green, C.C., Raphael, B.: The use of theorem-proving techniques in question-answering systems. In: Proceedings of the 1968 23rd ACM National Conference, pp. 169–181 (1968)

[15] Davis, R., Buchanan, B., Shortliffe, E.: Production rules as a representation for a

knowledge-based consultation program. Artificial intelligence **8**(1), 15–45 (1977)

[16] Minsky, M.: A framework for representing knowledge. MIT, Cambridge (1974)

[17] Fikes, R., Kehler, T.: The role of frame-based representation in reasoning. Communications of the ACM **28**(9), 904–920 (1985)

[18] Sowa, J.F.: Semantic networks. Encyclopedia of Cognitive Science (2012)

[19] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)

[20] Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)

[21] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI Conference on Artificial Intelligence (2015)

[22] Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (volume 1: Long Papers), pp. 687–696 (2015)

[23] Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. In: Icml (2011)

[24] Yang, B., Yih, W.-t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)

[25] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080 (2016). PMLR

[26] Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. Advances in neural information processing systems **31** (2018)

[27] Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)

[28] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[29] Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model

for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)

[30] Jiang, X., Wang, Q., Wang, B.: Adaptive convolution for multi-relational learning. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 978–987 (2019)

[31] Feng, W., Zha, D., Wang, L., Guo, X.: Convolutional 3d embedding for knowledge graph completion. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 1197–1202 (2022). IEEE

[32] Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, pp. 593–607 (2018). Springer

[33] Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.: Composition-based multi-relational graph convolutional networks. arXiv preprint arXiv:1911.03082 (2019)

[34] Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. arXiv preprint arXiv:1906.01195 (2019)

[35] Che, F., Zhang, D., Tao, J., Niu, M., Zhao, B.: Parame: Regarding neural network parameters as relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2774–2781 (2020)

[36] Vu, T., Nguyen, T.D., Nguyen, D.Q., Phung, D., *et al.*: A capsule network-based embedding model for knowledge graph completion and search personalization. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2180–2189 (2019)

[37] Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., Talukdar, P.: Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3009–3016 (2020)

[38] Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., Li, S., Sui, Z.: Towards time-aware knowledge graph completion. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1715–1724 (2016)

[39] Dasgupta, S.S., Ray, S.N., Talukdar, P.: Hyte: Hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on

Empirical Methods in Natural Language Processing, pp. 2001–2011 (2018)

[40] Liu, Y., Hua, W., Xin, K., Zhou, X.: Context-aware temporal knowledge graph embedding. In: International Conference on Web Information Systems Engineering, pp. 583–598 (2020). Springer

[41] Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. arXiv preprint arXiv:1904.05530 (2019)

[42] Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., Cheng, X.: Temporal knowledge graph reasoning based on evolutional representation learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 408–417 (2021)

[43] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[44] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems **33**, 7462–7473 (2020)