Enabling Efficient and Interpretable Cybersecurity Reasoning through Hyperdimensional Computing

Ali Zakeri, Hanning Chen, Narayan Srinivasa, *Fellow, IEEE*, Hugo Latapie, *Member, IEEE*, and Mohsen Imani, *Member, IEEE*

***Abstract*—Knowledge graphs play a crucial role in addressing the complexities of cybersecurity, as the increasing frequency and sophistication of cyber threats pose significant challenges to traditional defense technologies. In this paper, we propose a novel reasoning model, called INCYSER, that is tailored for cybersecurity. By leveraging Hyperdimensional Computing (HDC) as a symbolic and transparent computational model, INCYSER offers efficient and interpretable reasoning capabilities, ensuring reliable and trustworthy outcomes. Our model combines embedding-based unsupervised learning and HDC-based graph representation learning to construct a general representation for cybersecurity knowledge graphs, enabling diverse tasks including reasoning and general graph operations. Experimental evaluations demonstrate the effectiveness and efficiency of INCYSER, surpassing state-of-the-art models in link prediction and triple classification tasks. Additionally, a comprehensive ablation study examines the impact of various hyperparameters, showcasing the versatility of INCYSER. This work contributes to advancing the field of cybersecurity by introducing an interpretable and representation-based reasoning model for cybersecurity knowledge graphs.**

***Impact Statement*—In the face of increasingly sophisticated cyber threats, our work introduces INCYSER, a novel Interpretable Cybersecurity Reasoning model that leverages HDC for enhanced efficiency and interpretability in cybersecurity analysis. This model represents a significant leap over existing methodologies by delivering state-of-the-art performance in critical cybersecurity reasoning tasks and showcasing remarkable gains in inference speed and energy efficiency. INCYSER's advancements, including an average improvement of 1.3% in mean reciprocal rank (MRR) over leading models and up to 25.1% faster inference times, position it as a pivotal development in real-time cybersecurity applications and energy-constrained settings. Beyond its core capabilities, INCYSER's proficiency in general graph-related tasks underscores its versatility and potential to address a broad spectrum of challenges in cybersecurity and knowledge graph-based reasoning. By providing a robust, efficient, and interpretable model, INCYSER marks a significant stride towards next-generation cybersecurity systems equipped to combat the dynamic and complex landscape of cyber threats.**

***Index Terms*—Hyperdimensional Computing, Knowledge Graph Reasoning, Cybersecurity, Artificial General Intelligence**

I. Introduction

Submitted on 27 March 2024. This work was supported in part by DARPA Young Faculty Award, National Science Foundation #2127780 and #2312517, Semiconductor Research Corporation (SRC), Office of Naval Research, grants #N00014-21-1-2225 and #N00014-22-1-2067, the Air Force Office of Scientific Research under award #FA9550-22-1-0253, a grant from Intel, and generous gifts from Xilinx, AMD, and Cisco.

Ali Zakeri, Hanning Chen and Mohsen Imani are with the Bio-Inspired Architecture and Systems (BIASLab), UC Irvine, California, USA. E-mail: {azakerij, hanningc, m.imani}@uci.edu. Narayan Srinivasa is with Intel Labs, San Jose, Calofornia, USA, Email: narayan.srinivasa@intel.com. Hugo Latapie is with Cisco Systems, Inc. San Jose, California, USA. E-mail: hlatapie@cisco.com.

GIVEN the interconnected nature of the modern world, the significance of cybersecurity has reached unprecedented levels, driven by the escalating frequency and complexity of cyberattacks. Reasoning plays a paramount role in the realm of cybersecurity, empowering security analysts to derive valuable insights from extensive datasets, identify emerging threats, evaluate vulnerabilities, and make well-informed decisions to proactively address risks. The pivotal role of reasoning cannot be emphasized enough in ensuring effective mitigation of cyber risks and safeguarding digital assets in this dynamic landscape.

Cyberattacks and counterattacks unfold in environments that are both dynamic and complex, where a multitude of factors influence the success of attacks and the impact they have on missions. These environments within cyberspace are characterized by constant change, as applications are installed or removed, machines are added or removed, and various other factors contribute to the ever-evolving nature of the digital landscape. An avenue in the academic research in this domain aims at a high-level conceptualization of information security knowledge, which has has led to the creation of numerous ontologies [1]–[6]. As attackers continuously adapt their techniques and the information asymmetry between offensive and defensive forces in cyberspace grows, cybersecurity systems must evolve accordingly. To meet this demand, a range of novel tools and techniques has emerged to detect and counteract attacks. Knowledge graphs, in particular, have proven to be a powerful framework for enabling intelligent reasoning in cybersecurity. By capturing the intricate relationships and interdependencies among entities in cybersecurity domains, knowledge graphs offer valuable contextual information, enhancing the detection and prevention of cyber threats.

The field of knowledge graphs has witnessed notable advancements in various domains, indicating its growing maturity. Integrating knowledge graph principles with cybersecurity has become increasingly important, as some scholars have emphasized the need to build cybersecurity knowledge bases using knowledge graphs [7]–[12]. This integration is particularly vital due to the exponential increase in cybersecurity-relevant data generated within the cyberspace. The current challenge in cybersecurity analysis is not the lack of available data, but rather the task of integrating disparate, heterogeneous information from various sources into a cohesive model that enables reasoning for effective decision-making. With data that is diverse, fragmented, and varied, the focus lies on merging non-homogeneous data sources into a unified framework, allowing for comprehensive analysis and informed cybersecurity

decision-making.

The enduring nature and concealed attributes of contemporary attacks expose the shortcomings of conventional defense technologies relying on expert rules, machine learning, and deep learning. Traditional approaches, such as feature extraction [13], [14] and anomaly detection [15], [16], fall short in capturing the complete landscape of attack behavior. The persistent and covert nature of these attacks necessitates advanced techniques that surpass the abilities of simplistic tools, as they fail to provide a comprehensive understanding of the intricacies involved. Traditional symbolic models often suffer from limitations such as inaccuracies and non-scalability when dealing with large and dynamic cybersecurity datasets. On the other hand, while Graph Neural Network (GNN)-based models have shown promise in learning from graph-structured data, they can be slow, computationally costly, and lack interpretability, thereby undermining their trustworthiness for critical cybersecurity applications. Moreover, despite the extensive work in cybersecurity and knowledge graph reasoning, there remains a notable gap in models that effectively bridge these two fields. As discussed in this section and elaborated in Section V, existing methods often focus on isolated aspects, such as knowledge graph construction or machine learning models applied to non-graph-based data, lacking comprehensive integration.

In this paper, we address these limitations by proposing a novel reasoning model specifically designed for cybersecurity. Our model is efficient, interpretable, ensuring reliable and trustworthy reasoning outcomes. To achieve these goals, we leverage the power of Hyperdimensional Computing (HDC) as a symbolic and transparent computational model. HDC operates on a set of well-defined and explicitly represented symbols or entities, which are represented as high-dimensional vectors that encode various attributes and properties of the symbols. The representations are constructed using a set of predefined rules or operations, such as bundling and permutation, that are easy to understand and interpret. These properties enable efficient encoding, manipulation, and reasoning with symbolic representations, providing a promising avenue for cybersecurity knowledge graph reasoning.

The key contributions of this paper include the following:

- We develop INCYSER: INterpretable CYberSEcurity Reasoning, which is to the best of our knowledge, the first interpretable and representation-based reasoning model for Cybersecurity Knowledge Graphs (CSKGs). It addresses the need for a model that can provide insights into the reasoning process, while also achieving state-of-the-art performance in the context of cybersecurity.
- By fusing embedding-based unsupervised learning of knowledge graphs and HDC-based graph representation learning, INCYSER is capable of learning a general representation for CSKGs. This representation captures the underlying structure and relationships in the graph, and can be effectively used for a wide range of tasks, including cybersecurity reasoning tasks and general graph-related tasks.
- Our experiments demonstrate the effectiveness and efficiency of INCYSER on reasoning tasks. We show improvements on link prediction and triple classification tasks, through both accuracy and resource utilization evaluations, over various knowledge graph reasoning models, including state-of-the-art GNN-based models. We also display the versatility of INCYSER with a thorough ablation study over a selection of model's hyperparameters.

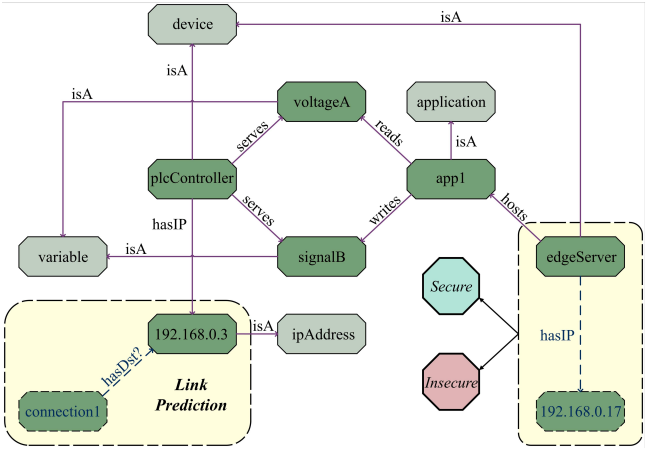

Fig. 1. An example of a cybersecurity knowledge graph. Blocks and arrows represent entities and relations in the knowledge graph, respectively. The two knowledge graph tasks of link prediction and triple classification are shown as well, with the former trying to predict the relation between the entities *192.168.0.3* and *connection1*, and the later classifying the triple (*edgeServer*, *hasIP*, *192.168.0.17*) as either "secure" or "insecure".

We evaluate the performance of INCYSER with a detailed comparison on cybersecurity knowledge graph reasoning tasks, including link prediction and triple classification, perform on publicly available datasets utilized and validated by the cybersecurity community. Our model shows an average improvement of **1.3%** in mean reciprocal rank (MRR) when compared to state-of-the-art GNN-based knowledge graph reasoning model.

Additionally, we illustrate the efficiency of INCYSER through comprehensive resource utilization evaluations, including speedups in latency and improvements in energy efficiency. Our model shows up to **25.1%** speedup and **24.2%** energy efficiency improvement over the state-of-the-art knowledge graph reasoning model.

By advancing reasoning capabilities in cybersecurity through the fusion of knowledge graphs and hyperdimensional computing, our work paves the way for more sophisticated and reliable cybersecurity systems capable of tackling the ever-evolving landscape of cyber threats.

II. Preliminaries

A. Knowledge Graphs

A formal representation of a knowledge graph involves structuring it as a directed graph $G = \{(e_s, r, e_o) \mid e_s, e_o \in \xi,\ r \in R\}$, where ξ is the set of entities and R is the set of relations. An example of a knowledge graph is demonstrated in Figure 1. In a knowledge graph, each factual statement is denoted by a directed edge, referred to as a triple $l = (e_s, r, e_o)$. It is important to note that entities can be

associated with multiple triples, as demonstrated in the given example. Additionally, relations within the knowledge graph are represented as directed edges, with the direction indicating the subject and object of the factual statement encapsulated by a triple.

Knowledge graphs serve as valuable structures for integrating data from diverse knowledge domains, often incorporating specialized vocabularies and ontologies that capture various categories, relationships, rules, and constraints within specific knowledge areas. Furthermore, knowledge graphs can leverage machine learning techniques, such as knowledge graph embeddings, to transform the symbolic representations of graph entities into vector representations that are better suited for processing with neural networks and similar methodologies. More specifically, embedding techniques for knowledge graphs facilitate the transformation of their components, including entities and relations, into continuous vector spaces. This process streamlines the manipulation of knowledge graph data while retaining the essential structure of the graph. Knowledge graph embeddings have garnered significant interest due to their potential to enhance various downstream tasks, such as graph completion and link prediction, opening up new avenues for research and application in the field [17].

Within the field of cybersecurity, knowledge graphs provide a versatile representation of information, which can be expressed using the subject-predicate-object format, denoted as $\{e_s, p, e_o\}$ for brevity, where the subject (e_s) corresponds to an entity within the graph, the predicate (p) represents a specific relationship, and the object (e_o) can be either another entity or a literal, e.g., a specific data value. This representation aligns with the Resource Description Framework (RDF) graph model, widely adopted for structured data representation on the web, which offers a flexible and extensible framework for capturing complex relationships and dependencies in cybersecurity domains. It enables the integration of diverse data sources and leverages standardized vocabularies, such as domain-specific ontologies, to capture the rich semantics and contextual information. By harnessing the power of RDF graphs, cybersecurity knowledge graphs facilitate efficient information integration, querying, and reasoning, empowering security analysts to extract actionable insights and make informed decisions in the dynamic and ever-evolving landscape of cyber threats.

B. Hyperdimensional Computing Basics

Hyperdimensional Computing (HDC), also known as Vector Symbolic Architecture (VSA) [18]–[20], offers a unique approach to manipulating data structures by transforming them into distributed representations. HDC achieves this by encoding data structures holographically within high-dimensional vector spaces. In this framework, random hypervectors, i.e., high-dimensional vectors, serve as the foundation for representing data. The encoding process involves utilizing three key operations that form an algebraic system within the high-dimensional space. By leveraging these operations and the power of high-dimensional representations, HDC enables efficient and effective manipulation of data structures, providing a promising computational model for a wide range of applications.

In this research, atomic primitives are represented by independent random bipolar vectors with a dimension of D. These base vectors, which represent specific entities, are stored in a codebook matrix of size $D \times n$, where D refers to the dimension of each entity hypervector, and n corresponds to the total number of entities in the knowledge graph. The similarity between vectors is evaluated through their dot product, $\langle \vec{x}, \vec{y} \rangle = \sum_{i=1}^{D} x_i y_i$, which determines whether the atomic objects are identical and captures the structural similarity between complex objects. To build structured objects, the HDC framework defines three fundamental operations as follows:

a) Bundling ($+$): operation is performed by element-wise addition of hypervectors, represented as $s = \vec{x}_1 + \vec{x}_2$. This operation combines the elements, functioning as a memory mechanism that retains the information from the input data in the resulting vector. The bundled hypervector maintains similarity to its constituent atomic hypervectors, indicated by $\langle \vec{s}, \vec{x}_1 \rangle \gg 0$, making it suitable for representing sets. In the case of bipolar vectors, each component in the result of a bundling operation takes an integer value, indicating that $\vec{s} \in \mathbb{Z}^D$. Importantly, it is possible to quantize the components, without impacting any of the aforementioned properties, to ensure that the bundled hypervector remains bipolar.

b) Binding ($\odot$): of $\vec{x}_1$ and $\vec{x}_2$ is achieved through the Hadamard product, which involves component-wise multiplication between the two hypervectors and is denoted as $\vec{s} = \vec{x}_1 \odot \vec{x}_2$. The resulting conjunct hypervector $\vec{s}$ differs significantly from its individual constituent vectors, as indicated by $\langle \vec{s}, \vec{x}_1 \rangle \approx 0$. Due to its ability to maintain information from its constituents without increasing the size, the binding operation can effectively serve as a means of variable binding.

c) Permutation (ρ): operation, denoted as $\vec{s} = \rho(\vec{x})$, involves cyclically shifting the components of the hypervector $\vec{x}$. The permutation operation exhibits certain properties: it distributes over bundling $\rho(\vec{x_1} + \vec{x}_2) = \rho(\vec{x}_1) + \rho(\vec{x}_2)$, as well as binding, $\rho(\vec{x}_1 \odot \vec{x}_2) = \rho(\vec{x}_1) \odot \rho(\vec{x}_2)$. Similar to binding, permutation causes dissociation between hypervectors, as indicated by $\langle \rho^l(\vec{x}), \vec{x} \rangle \approx 0$ when $l \neq 0$. The permutation operation is reversible and introduces a natural order through repeated application over the hypervector. As a result, it is utilized to represent sequences and different hierarchical levels within a structure.

III. Interpretable Cybersecurity Reasoning

We now propose our interpretable cybersecurity knowledge graph representation learning scheme, INCYSER. Figure 2 presents a high-level view of our design in practice. As the main learning approaches used in INCYSER, we will go over HDC-based graph representations, followed by the architecture of the unsupervised knowledge graph learning model.

A. Graph Representation in Hyperspace

Despite variations in algorithms and node ordering, all existing HDC-based graph representations share a common approach of utilizing edge correspondence as a measure of

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

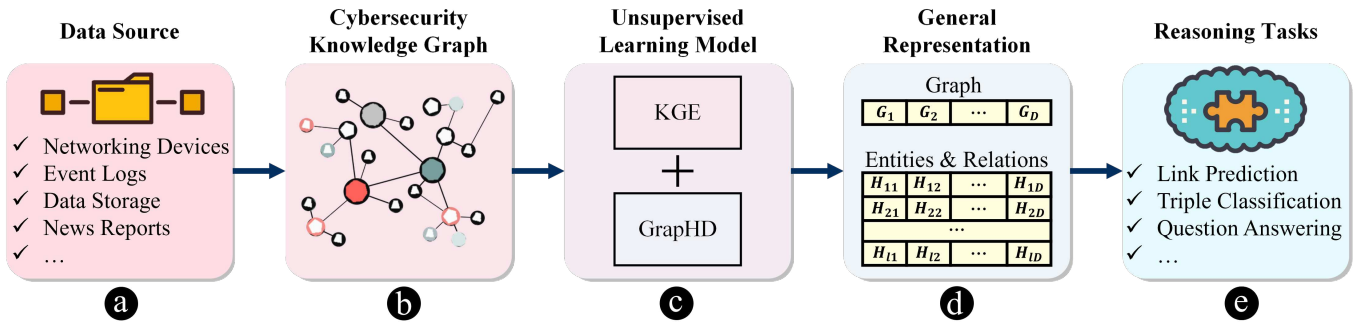

Fig. 2. Overview of the workflow for INCYSER. (a) The raw data used for learning the model can be gathered from various sources such as network systems and event logs. (b) Assembled dataset is the turned into a cybersecurity knowledge graph designed for representation using a list of entities and relations. (c) The CSKG is then passed through an unsupervised learning model, which is built using the structures of knowledge graph embedding model and HDC-based graph representation. (d) Training the mentioned model will allow INCYSER to transform the initial CSKG into a series of embeddings and hypervectors, which act as a general representation of the CSKG. (e) This representation can be used to perform various knowledge graph reasoning tasks, as well as general graph-related tasks.

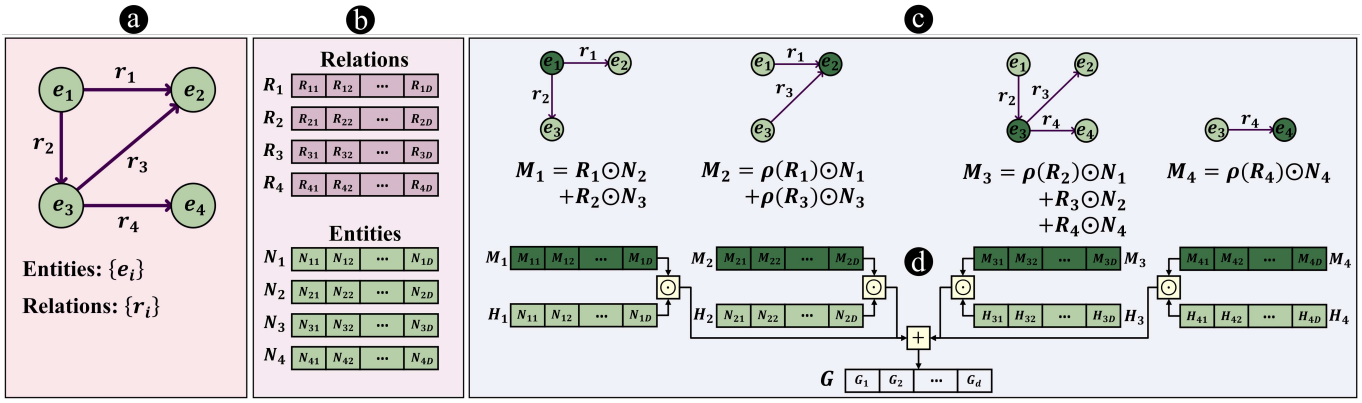

Fig. 3. Overview of the hyperdimensional graph representation. (a) A sample knowledge graph with 4 entites and 4 relations is given. We perform an encoding on such graph going through following steps: (b) The hypervectors for each entity $\vec{N}_i$ and relation $\vec{R}_i$ are generated. (c) A node memory hypervector is calculated for each graph node by bundling the association of its neighboring nodes and their respective edges. Note that the direction of the edges matter; an edge pointed inwards with be noted with a permutation on its hypervector. (d) The node memories are binded with their respective node hypervectors and bundled to result in the final graph hypervector, denoted as $\vec{G}$.

structural similarity. Figure 3 provides a general overview of these methods.

For a graph $G = (V, E)$, HDC first assigns a random hypervector $\vec{N}_i$ to each node $i \in V$ and $\vec{R}_j$ to each edge $j \in E$, both of size D. It then generates the codebook matrices $N = [\vec{N}_1, \vec{N}_2, ..., \vec{N}_n]$ with the size $D \times n$, and $R = [\vec{R}_1, \vec{R}_2, ..., \vec{R}_m]$ with the size $D \times m$ as the high-dimensional representation of graph nodes and edges, respectively. Due to their random generation, the codebook hypervectors exhibit a high degree of orthogonality, with the similarity between each pair of hypervectors approaching zero: $\langle \vec{N}_i, \vec{N}_j \rangle \approx 0, \langle \vec{R}_i, \vec{R}_j \rangle \approx 0, i \neq j$.

For each node l, the algorithm constructs a node memory hypervector by combining the hypervectors of its neighbors: $\vec{M}_l = \sum_{i \in Nbh(l)} \vec{R}_{li} \odot \vec{N}_i$, where $Nbh(l)$ represents the nodes in the neighborhood of node l, and $\vec{R}_{li}$ denotes the directed edge from node l to node i. The binding operation ensures that entities and their associated relationships retain unique context, enabling accurate differentiation and effective handling of ambiguities. To further preserve relational distinctions, permutation is applied for reverse edges, allowing the model to represent and differentiate opposite relationships by maintaining the direction of edges. Additionally, bundling combines related information to create a comprehensive representation of each node and its connections, capturing the collective context of neighboring relationships.

The use of bundling ensures that the similarity between the node memory hypervector and each bundled hypervector remains significantly greater than zero, given the corresponding edge hypervector between the two nodes: $\forall i \in Nbh(l), \langle \vec{M}_l, \vec{R}_{li} \odot \vec{N}_i \rangle \gg 0$. This characteristic facilitates the retrieval of each neighbor during the reconstruction process by performing similarity comparisons. Consequently, the model follows a two-step process to construct a single hypervector for the entire graph. In the first step, it binds each node hypervector with its corresponding node memory, resulting in $\vec{N}_i \odot \vec{M}_i$. This binding operation introduces a distinction between the node memories and enables the model to correctly associate each node with its corresponding memory during the graph reconstruction. In the second step, all the associated pairs are bundled together, creating the graph hypervector $\vec{G}$:

$$\vec{G} = \sum_{i=1}^{n} \vec{N}_i \odot \vec{M}_i = \sum_{l=1}^{n} \vec{N}_l \odot \sum_{i \in Nbh(H_l)} \vec{R}_{li} \odot \vec{N}_i \quad (1)$$

The final equality in the expression leverages the distributivity property of binding over bundling, enabling the transparent compression of all edges in the graph. Each edge is represented by the binding of the hypervectors corresponding to its endpoints. This compressed representation serves as a powerful tool for reconstructing the original graph and facilitating various reasoning tasks.

To extract the graph information from the graph hypervector, we initiate the retrieval process by recovering the memory of each individual node. This can be approximately done by binding the graph hypervector $\vec{G}$ with the corresponding node hypervector:

$$\tilde{\vec{M}}_i = \vec{G} \odot \vec{N}_i = \vec{M}_i + \sum_{j \neq i}^{n} \vec{N}_i \odot \vec{N}_j \odot \vec{M}_j \quad (2)$$

It is important to note that the recovered node memory is denoted as $\tilde{\vec{M}}_i$, and it contains some additional terms compared to the original node memory hypervector $\vec{M}_i$. As we progress with the decoding process, these extra terms are disregarded due to the orthogonality of hypervectors.

Having estimated the node memory, we can inspect the connection between nodes j and i by measuring the similarity between the node hypervector of one, the node memory hypervector of the other, and the edge between them $S = \langle \vec{N}_j \odot \vec{R}_{ij}, \tilde{\vec{M}}_i \rangle$. If there exists an edge between i and j, then $S \gg 0$. Otherwise, $S \approx 0$.

In conclusion, the proposed representation algorithm utilizes hyperdimensional computing to encode and reconstruct graph structures efficiently while preserving contextual and relational distinctions among nodes and edges. By leveraging operations such as binding, bundling, and permutation, the approach achieves a compact yet expressive representation that supports reasoning tasks and scales effectively across diverse graph configurations. These characteristics align with broader advancements in hyperdimensional algorithms, as explored in [21], where complexity and accuracy are analyzed in greater depth. The transparent compression of graph edges and the efficient retrieval mechanisms further highlight the versatility of this method for applications requiring both interpretability and computational efficiency

B. Unsupervised Learning Model

At the core of INCYSER lies a learning model that is specifically designed to train on the knowledge graph and obtain a comprehensive representation that is not task-specific. This model is responsible for learning representations of the entities and relations within the knowledge graph. It incorporates an HDC encoder and a loss function, which undergo changes during training to facilitate the learning process of these representations. Figure 4 provides an overview of the architecture of the learning model, illustrating its components and their interactions. The algorithm utilized for model training is also presented in Algorithm 1.

Algorithm 1 INCYSER Training

Require: Training set S, Loss function L_{fn}, Embedding dimension d, Hypervector dimension D, Encoding activation function f, Encoding matrix $B_{d \times D}$
$\vec{e}_i, \vec{r}_j \leftarrow \mathcal{N}(0, \frac{4}{d})$, $i \in [1, n]$, $j \in [1, m]$
loop
  $S_{batch} \leftarrow$ NextBatch(S)
  $\vec{N}_i = f(\vec{e}_i B)$, $i \in [1, n]$ ▷ Encode entities
  $\vec{R}_j = f(\vec{r}_j B)$, $j \in [1, m]$ ▷ Encode relations
  if L_{fn} = TransE **then**
    $\mathcal{L} = \sum_{(i,j,k) \in S_{batch}} ||\vec{N}_i + \vec{R}_j - \vec{N}_k||$
  end if
  if L_{fn} = DistMult **then**
    $\mathcal{L} = \sum_{(i,j,k) \in S_{batch}} ||\vec{N}_i diag(\vec{R}_j) \vec{N}_k^T||$
  end if
  Update $\vec{e}_i$, $\vec{r}_i$ w.r.t. $\nabla \mathcal{L}$
end loop

The encoder takes the initial embeddings of the nodes in the graph and maps them into high-dimensional space, where the encoded vectors can be utilized capture the structural information of the graph, as shown in Section III-A. Once the encoded vectors are obtained, the model computes a loss based on the distance between the encoded vectors of adjacent nodes in the graph. The loss encourages the model to learn embeddings that capture the structural relationships between nodes, such as similarity, proximity, and context. We experiment with TransE [22] and DistMult [23] in our study, both of which are essential knowledge graph embedding scoring functions, and can be used individually or as a part of more complex model for performing reasoning tasks. An in-depth comparison of the performance of INCYSER using each of the mentioned scoring functions is discussed in Section IV-D

Finally, the model updates the initial embeddings based on the computed loss, using an optimization algorithm such as stochastic gradient descent or Adam [24]. This process is repeated until the embeddings converge to a stable solution. As the output of this model, the updated embeddings, as well as their high-dimensional representations, can then be used for various downstream tasks, including reasoning tasks of link prediction and triple classification, and general graph tasks such as graph reconstruction. The flexible nature of this general representation allows for the model being able to perform various tasks defined on knowledge graphs, rather than being designed for a single specific reasoning task.

IV. Experiments

In this section, we will display the experiments carried out on INCYSER and discuss their significance. We conduct three types of experiment on our model. First, we analyze the performance of our model in knowledge graph reasoning tasks of link prediction and triple classification and compare the results with several recent studies, including state-of-the-art

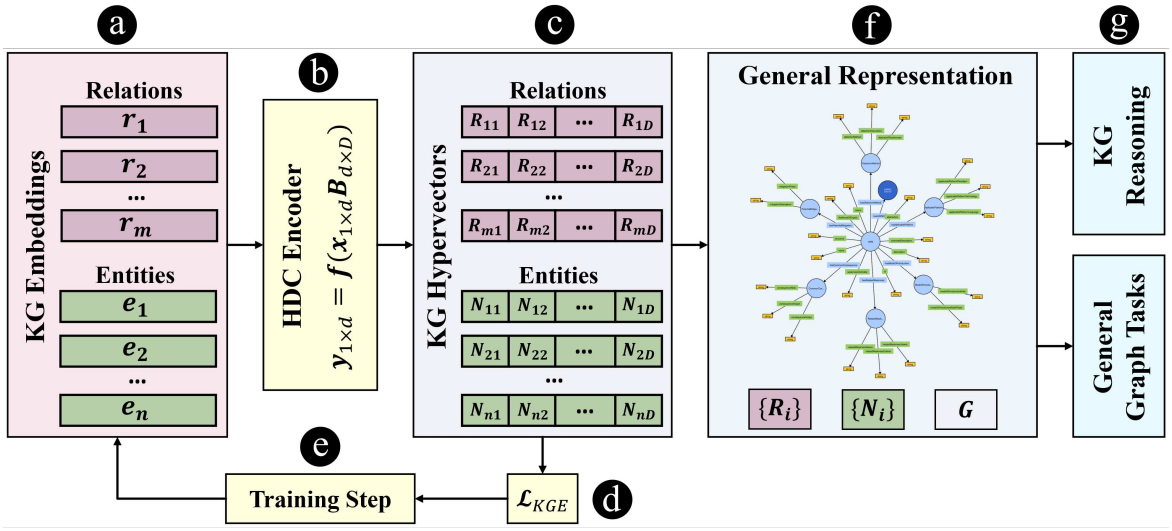

Fig. 4. The structure of the unsupervised learning model. (a) The model starts with generating low-dimensional embeddings for entities and relations in the knowledge graph, denoted as $\vec{e}_i$ and $\vec{r}_i$. (b) The embeddings are transformed into a high-dimensional representation using a fixed HDC encoder with the encoding matrix B and activation function f. (c) After the completion of training, the outcome is a collection of hypervectors that can be utilized for various tasks on the knowledge graph. (d) The model calculates an embedding loss function on the hypervectors generated for the entites and the relations. (e) The loss function is employed to learn and update the low-dimensional embeddings, aiming to enhance the representation of the knowledge graph. (f) The output of the model is a general representation for the graph, which includes hypervectors for each entity and relation in $\vec{N}_i$ and $\vec{R}_i$, and a graph hypervector $\vec{G}$. (g) The acquired representation can be leveraged to tackle a wide range of knowledge graph reasoning tasks as well as general graph-related tasks.

TABLE I
STATISTICS FOR THE SELECTED CYBERSECURITY KNOWLEDGE GRAPH DATASETS. THE FIGURES UNDER TRAIN, VALID, AND TEST COLUMNS REPRESENT THE NUMBER OF TRIPLES IN EACH DATASET SPLIT.

Dataset	Entities	Relations	Train	Valid	Test
SEPSES	32092	105	69743	23214	23058
IAD	4346	38	24359	7546	5614

designs, on the task. Second, we demonstrate the effectiveness of INCYSER in performing general graph-related tasks such as graph reconstruction and shortest path problem. Lastly, we assess our model empirically by measuring several performance metrics in an ablation study on model hyperparameters. In all experiments, INCYSER was implemented using Python and PyTorch, and evaluations were conducted on a platform equipped with an Intel i9-12900KF CPU and an NVIDIA RTX 4090 GPU, except when mentioned. The runtime environment included Ubuntu 20.04 with Python 3.8. To ensure a fair and consistent evaluation of all compared models, we used the same initialization state (i.e., consistent seeding) across all experiments.

A. Datasets

In our study, we utilized publicly available datasets that have been used and validated by the cybersecurity community, properties of which is shown in Table I. The Semantic Processing of Security Event Streams (SEPSES) project [7] includes a cybersecurity knowledge graph that integrates and links critical information such as vulnerabilities, weaknesses and attack patterns from various publicly available sources, modeling intricate relationships among cyber threat entities, the diversity of which simulates the multifaceted nature of cyberattacks in practice. The knowledge graph is continuously updated to reflect changes in various data sources used as inputs, including Common Attack Pattern Enumeration and Classification (CAPEC), Common Platform Enumeration (CPE), Common Vulnerabilities and Exposures (CVE), Common Vulnerability Scoring System (CVSS), and Common Weakness Enumeration (CWE). Due to the size of the full knowledge graph, we utilized the subgraphs containing CWE and CAPEC ontologies for our study. The selected subset is preprocessed from the standard RDF form, removing the unnecessary description properties, to acquire the datasets for training, validation, and testing. An overview of the structure of this dataset is shown in Figure 5.

We also employ the dataset generated by [25], which is gathered using an Industrial Automation Demonstrator (IAD) that mimics the integration of OT and IT technologies. Activity from this system is recorded and subsequently transferred into a knowledge graph that models entities, like devices and IP addresses, and their relationships, such as network connections and data accesses. The dataset is composed of a training set, which is a recording of baseline activity of the automation system, and a testing set, where deviations from the baseline were added during testing time according to standard cybersecurity attack patterns. Novel activity is assigned five different degrees of severity thorough analysis of the available contextual information by a human expert, ranging from "observed" and "expected", to "highly-suspicious". To include more support data for each class, we combine the "highly-suspicious" and "suspicious" classes into one, as well as labeling "expected" and "unexpected" classes as safe. Hence the goal of the task is to separate "suspicious" connections (i.e.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

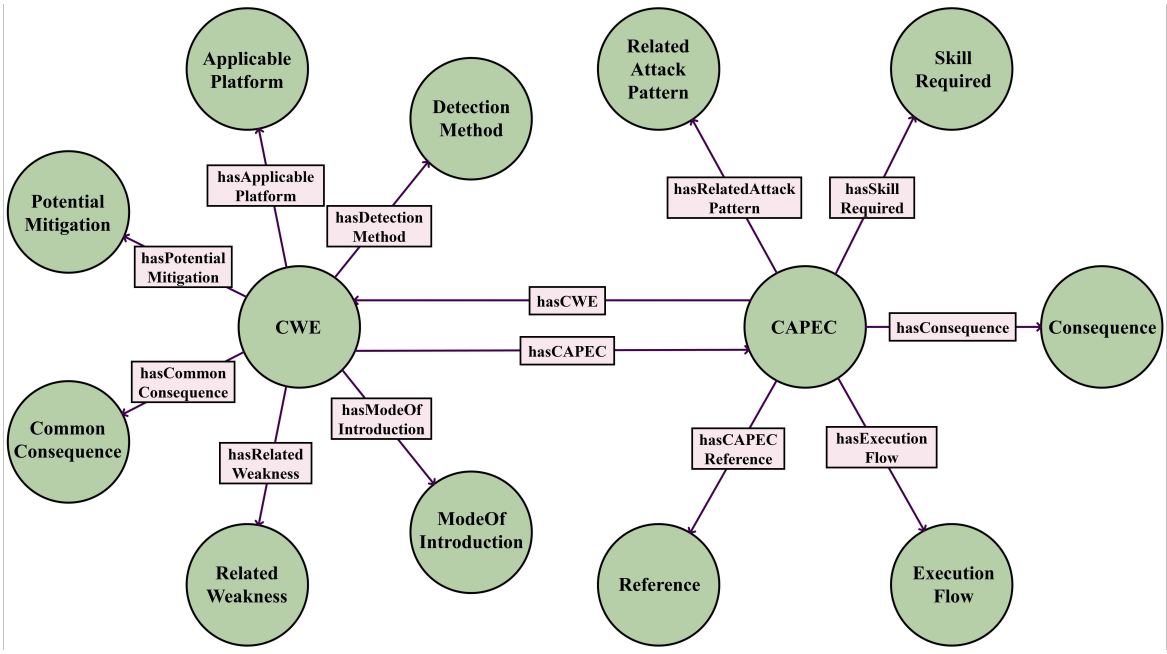

Fig. 5. Main classes and their relations in the utilized subgraphs of the SEPSES cybersecurity knowledge graph. The cybersecurity knowledge graph is initially in the form of RDF graphs, which is preprocessed into sets of triples for training, validation and testing datasets.

TABLE II
LINK PREDICTION PERFORMANCE OF INCYSER AND PREVIOUS MODELS ON SEPSES AND IAD DATASETS.

	SEPSES				IAD			
	H@10	**H@3**	**H@1**	**MRR**	**H@10**	**H@3**	**H@1**	**MRR**
RESCAL [26]	0.002	0.001	0.001	0.001	0.003	0.002	0.002	0.002
SimplE [27]	0.185	0.114	0.040	0.090	0.728	0.325	0.121	0.282
DistMult [23]	0.211	0.117	0.042	0.096	0.716	0.311	0.099	0.269
TransE [22]	0.440	0.336	0.227	0.300	0.820	0.762	0.655	0.718
R-GCN [28]	0.315	0.246	0.184	0.229	0.773	0.736	0.692	0.722
ConvTransE [29]	0.436	0.355	0.284	0.337	0.779	0.760	0.701	0.736
SACN [29]	0.481	0.429	0.372	0.412	0.785	0.760	0.732	0.752
RAGAT [30]	0.514	0.459	0.396	0.437	0.838	0.814	0.774	0.799
CompGCN [31]	0.522	0.468	0.408	0.449	0.846	0.822	0.774	0.803
INCYSER (Proposed Method)	0.525	0.470	0.414	0.452	0.852	0.835	0.807	0.825

cybersecurity knowledge graph triples) from the ”safe” ones. This complements the other dataset by reflecting dynamic threat patterns and interconnected attack surfaces, making it particularly suitable for evaluating the model's adaptability to evolving cyber threat landscapes. Together, these datasets provide a comprehensive benchmark for assessing the practical relevance and efficacy of INCYSER in real-world scenarios.

B. Reasoning Tasks

We benchmark INCYSER's reasoning ability over the SEPSES and IAD datasets, mainly considering two tasks: link prediction and triple classification. Link prediction is performed over both datasets, while triple classification is done over IAD, since it includes the extra labeled data needed for the task. The performance of our model is compared with several benchmark models in knowledge graph reasoning, including the state-of-the-art models. Table II provides models' performances for the link prediction task, assessed using commonly used evaluation metrics, including hits@k and MRR. Hits@k measures the proportion of times the correct answer appears in the top k positions of the search results, indicating the model's ability to quickly provide relevant answers, with higher scores reflecting better accuracy. Meanwhile, MRR measures the average inverse rank of the first correct answer, highlighting the position of the first relevant result, with higher values indicating better performance. These metrics together evaluate a model's effectiveness in information retrieval. The triple classification results are also shown in Table III, where the comparisons are made through standard classification metrics.

It is evident that INCYSER outperforms all other models, which range from embedding based models [22], [23], [26], [27] to GNN-based models [28]–[31], for both cybersecurity knowledge graphs. More specifically, INCYSER achieved MRR scores of 0.452 and 0.825, surpassing the state-of-the-art model CompGCN by **0.003** and **0.022**, and the attention utilizing model RAGAT by **0.015** and **0.026**. There is also

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

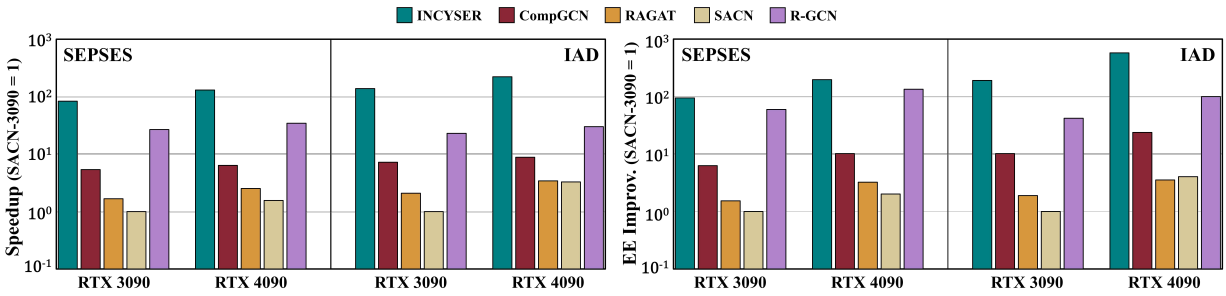

Fig. 6. Cross-model and cross-platform comparison of model speedup and energy efficiency improvement. The batch size across all experiments is fixed to 256.

improvements shown in across all measured hits@k metrics for both datasets, including **0.5%**, **1.3%**, and **5.6%** increase in hits@10 over CompGCN, RAGAT, and SACN, respectively.

These results highlight the superiority of INCYSER in capturing the complex relationships and patterns within the knowledge graph. The model's unique ability to leverage HDC techniques enables it to encode and reason with symbolic representations efficiently, leading to enhanced link prediction performance. The accuracy and precision of the link predictions can greatly enhance the usability and reliability of knowledge graph-based systems in cybersecurity applications.

One of the key advantages of leveraging hyperdimensional computing in INCYSER architecture, is the notable improvements it offers in terms of inference speedup and energy efficiency compared to GNN-based methods. Through extensive experiments, we observed a significant reduction in inference time with our reasoning model. INCYSER uses a fixed HDC encoding for the entity and relation embeddings, which allows for faster inference times, whereas GNN-based models models, such as CompGCN, need much more processing as a result of having trainable GNN layers. This improvement in speed is crucial for real-time cybersecurity applications where timely decision-making is vital.

Additionally, our INCYSER demonstrates enhanced energy efficiency compared to other approaches. By leveraging efficient encoding schemes and optimized computational operations inherent in hyperdimensional computing, we reduce the computational overhead and minimize energy consumption. This improvement is particularly valuable for resource-constrained environments or scenarios where energy conservation is a critical consideration.

To quantify the improvements, we conducted rigorous performance measurements and compared the inference speed and energy consumption of INCYSER with state-of-the-art GNN-based reasoning models. Our results, demonstrated in Figure 6, consistently showed a notable speedup in inference, specifically ranging from **15.4%** to **25.1%** over CompGCN, depending on the platform and dataset. Moreover, we observed energy efficiency improvements of **19.3%** on average, when compared to CompGCN, highlighting the energy-saving potential of our approach.

The significant gains in both inference speed and energy efficiency achieved by INCYSER contribute to its practical applicability and scalability in real-world cybersecurity scenarios. These improvements allow for faster response times, reduced latency, and increased overall system efficiency, leading to enhanced cybersecurity operations and resource optimization.

TABLE III
TRIPLE CLASSIFICATION PERFORMANCE OF INCYSER AND PREVIOUS MODELS ON IAD DATASET

	Precision	Recall	F1-score	Accuracy
TransE	0.678	0.659	0.669	0.542
DistMult	0.694	0.682	0.688	0.566
CompGCN	0.708	0.909	0.796	0.673
INCYSER	0.765	0.960	0.851	0.765

We also focused on the triple classification task within the realm of cybersecurity reasoning, aiming to accurately classify the labels accessed to the different triples within the knowledge graph. Through the utilization of HDC techniques, we achieved significant improvements in the performance of the triple classification task compared to existing approaches. As shown in Table III, INCYSER proves to be the best classifier across all metrics when compared with two commonly-used embedding-based approaches TransE and DistMult, and the state-of-the-art GNN-based model CompGCN. We selected these models for triple classification based on their relevance and strong performance in knowledge graph embedding tasks. TransE and DistMult serve as established baselines, while CompGCN represents a state-of-the-art GNN-based approach. This focused comparison ensures meaningful evaluation of INCYSER's classification capabilities, emphasizing models commonly applied to and fine-tuned for such tasks.

C. General Graph Tasks

In addition to excelling in cybersecurity reasoning tasks, INCYSER can also be utilized to perform general graph tasks, and it demonstrates promising performances. Specifically, we evaluated its effectiveness in graph reconstruction and the computation of shortest paths within knowledge graphs. It is important to note that this is a direct result of generating a general representation vectors for the knowledge graph, rather than learning the representations for any specific task, as most other knowledge graph reasoning models, including the GNN-based models, tend to do.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

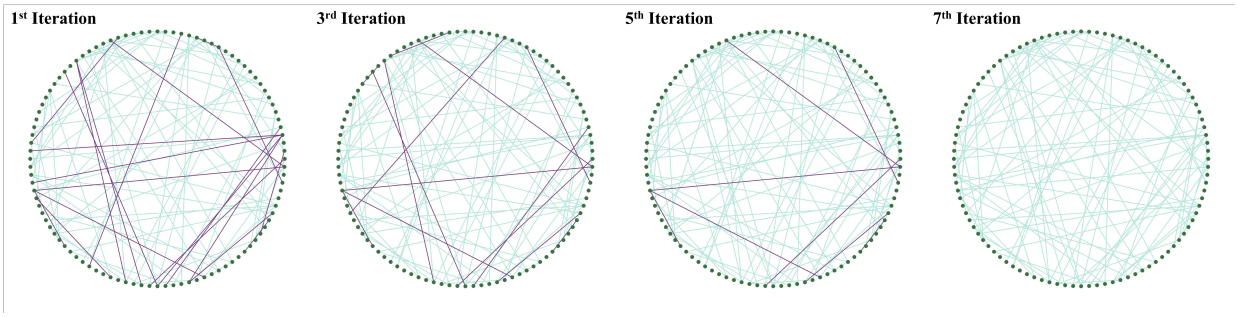

Fig. 7. Visualization of the graph reconstruction process, shown through iterations of error correction. Purple colored edges are mismatched edges, while cyan edges are the true predicted edges. It is worth noting that in graph-related tasks the results are primarily influenced by the graph structure itself, while the specific values assigned to entities and relations and their context can be disregarded.

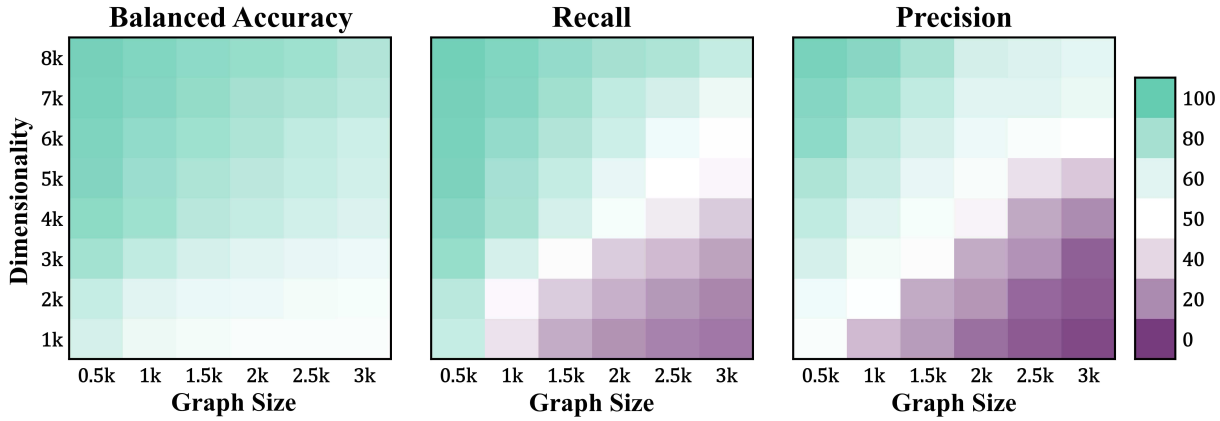

Fig. 8. Graph reconstruction report. The reconstruction task can be essentially seen as a classification problem, where the model aims at classifying whether an edge exist for each possible connection between the graph nodes. The experiments are presented in a heatmap format, where the horizontal axis represents the graph size, and the vertical axis represents the hypervector dimensionality. The heatmaps showcase the values of three classification metrics: balanced accuracy, recall, and precision. The graph density is set to be 0.001 across all runs. Furthermore, the use of balanced accuracy is employed to mitigate the impact of imbalanced data caused by the sparsity of graphs.

Graph reconstruction aims to reconstruct the original graph structure from a learned representation of graph ($\vec{G}$ hypervector in our case). This task is crucial in scenarios where partial or incomplete graph information is available, such as when dealing with missing or corrupted data. An example of the task is shown in Figure 7, while Figure 8 exhibits the reconstruction report of INCYSER across different model dimensionalities and graph parameters. By leveraging the rich symbolic representations and reasoning capabilities of HDC, our algorithm successfully infers the original connections between nodes, and proves to be highly effective in graph reconstruction. Across various parameter configurations, it's observable that increasing the dimensionality of the model enhances accuracy and further enables precise learning and retention of information for larger graphs. This improvement is attributed to the model's augmented ability to encode and process a richer set of nodes and connections within the graph through the use of larger hypervectors. Meanwhile, with dimensionality held constant, the model demonstrates higher accuracy in reconstructing smaller graphs.

Furthermore, we assessed the performance of our model in solving the shortest path problem, which involves finding the most efficient path between two nodes in a graph. Shortest path computation is a fundamental graph operation with diverse applications, ranging from network routing to resource optimization. The performance of our model on this task is demonstrated in Figure 9, which shows INCYSER's success rate in calculating shortest paths within knowledge graphs with different parameters, across different model dimensionalities. When working with a greater dimensionality, our model demonstrates increased capability to work with higher node depths in graphs. It efficiently navigates the graph structure and identifies the optimal paths, showcasing its effectiveness in solving this fundamental graph problem. While INCYSER demonstrates robustness across various graph configurations, increasing graph density presents additional challenges to maintaining accuracy. Dense graphs, characterized by a higher edge-to-node ratio, increase the complexity of distinguishing relationships due to the abundance of connections. The accuracy of bundled representations is inversely proportional to the number of vectors stored, assuming fixed vector dimensionality [21]. This effect is particularly evident in graph node memories, as the model is required to navigate and encode a greater number of interdependencies, i.e., edges, for graphs with higher densities. This behavior can be seen in the results shown in Figure 9, where success rates for the shortest path

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

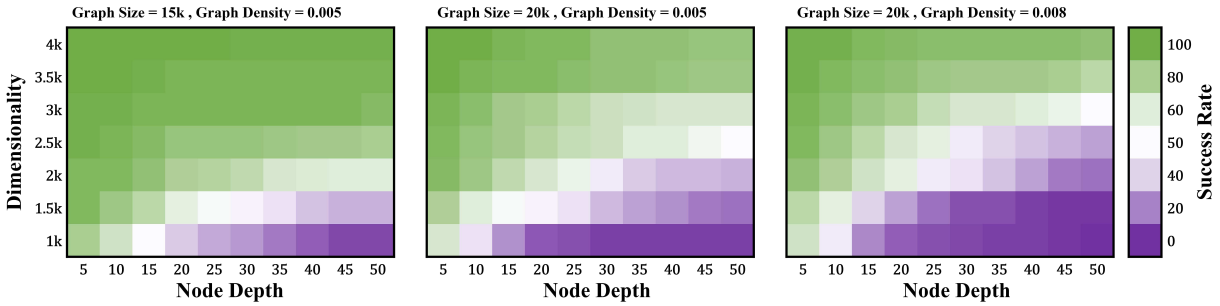

Fig. 9. Heatmap for success rate of the shortest path algorithm. The horizontal axis shows the distance between the two nodes, while vertical axis shows the hypervector dimensionality. Three set of experiments are performed with different parameters for graph structure, all of which closely mimic the graph structure of cybersecurity knowledge graphs used in the study.

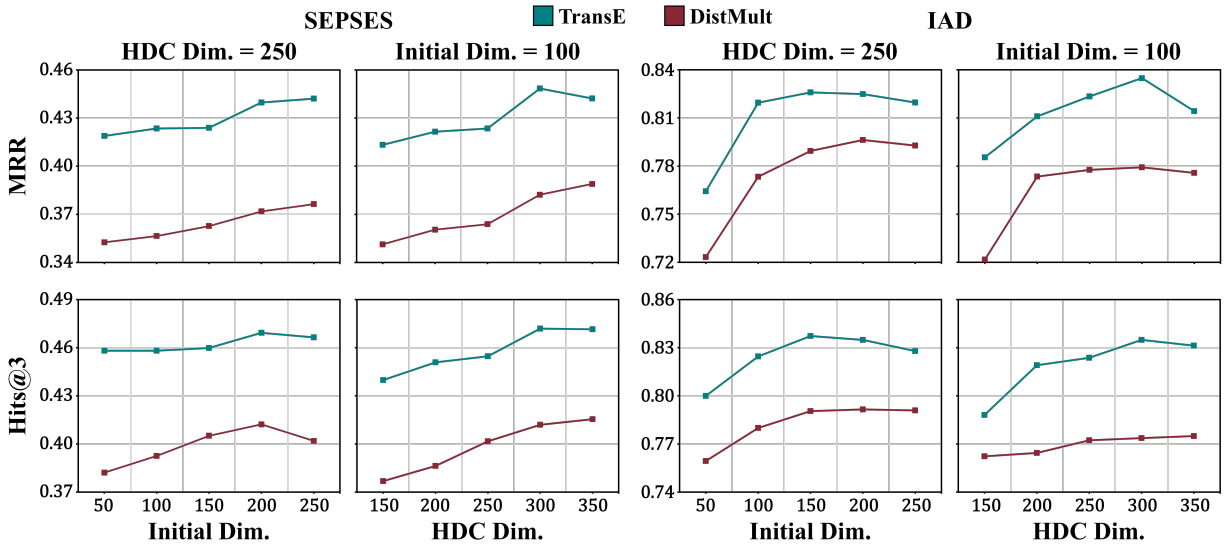

Fig. 10. Ablation study on INCYSER's most important hyperparameters: Initial embedding size, hypervector dimensionality and embedding scoring function. MRR measures the average reciprocal rank of correct predictions, reflecting how high they appear in the ranking list. Hits@3 indicates the proportion of correct predictions within the top three results, emphasizing precision.

task slightly decrease as graph density increases for graphs of the same size.

The effectiveness of INCYSER on general graph tasks extends beyond performance metrics, emphasizing its core strength—interpretability. By leveraging symbolic representations, INCYSER decodes graph structures into their constituent elements, providing direct and transparent insights into its decision-making process. Every decision and reasoning step can be traced back to high-dimensional representations that transparently encode relationships and attributes. This interpretability is quantified by INCYSER's capacity to accurately reconstruct graph structures and trace relationships, offering a clear view of how solutions are derived during graph-based reasoning tasks.

The success of INCYSER on general graph tasks highlights its versatility and broad applicability beyond cybersecurity reasoning. The interpretability, accuracy, and efficiency exhibited by our model provide valuable insights into its potential for addressing various graph-related challenges.

D. Ablation Study

The performance of INCYSER is influenced by the score function type as well as parameters in the graph structure encoder, such as entity and relation embedding dimension, encoded hypervector dimension, etc. In order to gain deeper insights into the performance and behavior of our proposed model, we conducted an ablation study where we systematically varied three key parameters: the embedding score function, initial embedding size, and hypervector dimensionality. We evaluated the impact of these parameter changes on the model's performance over both cybersecurity knowledge graphs in Table I, using two commonly used metrics: MRR and hits@3. All evaluations are depicted in Figure 10.

In our study, we examined the utilization of two frequently employed embedding score functions: TransE and DistMult. Additionally, we experimented with different initial sizes for entity and relation embeddings, as well as varying hypervector dimensionality. By systematically adjusting these hyperparameters, our objective was to evaluate their impact on the accuracy of link predictions for both datasets.

In all conducted experiments, TransE consistently outper-

formed DistMult across both metrics. Notably, the performance gap between the two approaches is more pronounced in the case of SEPSES compared to IAD. These findings align with our expectations, as DistMult, while computationally efficient, represents an oversimplified embedding model that may struggle with asymmetric relations. On the other hand, TransE exhibits superior performance in larger-scale knowledge graphs, further validating our observations. Moreover, the results indicate that increasing the initial or HDC vector dimensions generally improves the scores. However, there are exceptions where increasing one dimensionality while keeping the other fixed actually decreases performance. This can be attributed to the increased model complexity, which requires more training and may not necessarily lead to improved accuracy. These observations are particularly evident for larger dimension sizes, suggesting that the use of excessively large dimensionality may be unnecessary in certain cases.

V. Related Work

A. Cybersecurity Knowledge Graphs

Cybersecurity knowledge graphs play a crucial role in enhancing the representation of threat knowledge, enabling security researchers to effectively acquire diverse threat information for initial intelligent decision-making. Numerous academic studies have dedicated their focus to utilizing knowledge graphs in the field of cybersecurity.

[32] proposed the definition of distinct ontologies, utilizing expert knowledge, to depict the interconnections among entities, thereby facilitating a comprehensive understanding of the relationships within the domain. SEPSES [7] developed a knowledge graph dedicated for cybersecurity. Their work illustrated the process of generating security alerts and establishing connections between these alerts and the their cybersecurity knowledge graph. This integration aimed to provide a more comprehensive understanding of ongoing cyber threats by leveraging the rich information encapsulated within the knowledge graph. [33] utilized big data and natural language processing techniques to develop a knowledge graph of cybersecurity practices tailored for students. Their work involved constructing a comprehensive graph that captures the interrelationships and dependencies between various concepts in cybersecurity. With this knowledge graph, students are able to conveniently browse and search for relevant cybersecurity concepts, gaining a deeper understanding of the subject matter. [34] introduced a compact and human-readable threat intelligence recommendation system based on knowledge graph technology. Their system incorporates various features, including security items, network entities, and emerging hacker groups and attacks. By leveraging the power of knowledge graphs, the system provides valuable insights and recommendations in the field of threat intelligence, enabling users to make informed decisions and take appropriate actions to mitigate risks. [35] presented an ontology-based network security framework for the Internet of Things (IoT). The framework utilized IoT knowledge reasoning to monitor IoT devices, classify cyber threats or vulnerabilities, and implement effective countermeasures to safeguard the functioning of existing services. Open-CyKG [9] introduces an open cyber threat intelligence knowledge graph framework, which utilizes an attention-based neural open information extraction model to extract valuable cyber threat information from unstructured advanced persistent threat (APT) reports. The framework incorporates a neural cybersecurity named entity recognizer for entity identification and fusion techniques using word embeddings to construct a structured knowledge graph, enabling security professionals to query and retrieve valuable information with improved performance compared to existing models. CSKG4APT [12] focuses on constructing an APT knowledge graph using open-source cyber threat intelligence. Their platform is developed based on a knowledge graph model, incorporating deep learning and expert knowledge for knowledge extraction and updating. The proposed APT attack attribution method enhances network defense strategies by integrating fragmented intelligence and enabling active defense adjustments. Generally, APTs have been focus of many reasoning works in cybersecurity, including CONAN [36] and APT-KGL [37] as well. KRYSTAL [11] presents a modular framework based on a knowledge graph that addresses the limitations of existing approaches in attack graph-based methods. It enables the integration, extension, and reuse of techniques for threat detection, attack graph reconstruction, and analysis, improving detection capabilities and scalability, and also provides query options for contextualization and supports multiple detection techniques, such as tag propagation, attack signatures, and graph queries.

In real-world applications, numerous enterprises and institutions have adopted knowledge graphs to develop customized threat intelligence analysis systems tailored to specific scenarios, enabling them to enhance their cybersecurity measures and protect their assets.

In addition to specialized cybersecurity knowledge graph frameworks, broader graph processing and analytical tools have also demonstrated relevance in the field. Neo4j [38], a graph database management system, along with its Graph Data Science (GDS) library, offers sophisticated capabilities for clustering, pathfinding, and predictive analytics over graph data. Furthermore, tools such as IBM's Adversarial Robustness Toolbox (ART) [39] provide valuable methods for evaluating the resilience of graph-based models against adversarial modifications and noisy data, an aspect that could complement robustness testing of cybersecurity knowledge graphs. Gephi [40], an open-source graph visualization tool, enables interactive exploration of graph structures, making it useful for visualizing relationships and patterns in knowledge graphs, though it primarily focuses on surface-level analytics rather than deep reasoning tasks.

B. Knowledge Graph Reasoning

Knowledge graph embedding is utilized to represent the factoid triples of a knowledge graph, which include entities and relations, in a continuous and low-dimensional embedding space. This embedding aims to preserve the original entities and relations in the resulting vectors. Embedding models that aim to represent knowledge graphs utilize distance-based scoring functions to assess the plausibility of a triple consisting of

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3545394

a topic entity, predicate, and tail entity. These scoring functions measure the distance between the head and tail entities, and popular examples include TransE [22] and its extensions (e.g. TransH [41]), DistMult [23], and ComplEx [42]. In summary, a typical knowledge graph embedding technique involves representing entities and relations, defining a scoring function, and learning entity and relation representations. This approach simplifies manipulation while preserving the inherent structure of the knowledge graph, benefiting downstream tasks like entity alignment [43], relation prediction [44], and knowledge graph-based question answering [45].

Convolutional Neural Networks (CNNs) have been widely adopted in knowledge graph embedding due to their parameter efficiency and fast training capabilities. ConvE [46], a specific knowledge graph embedding model, leverages convolutional feature filters applied to a matrix reshaped from subject and relation embeddings. InteractE [47] enhances the performance of ConvE through the use of feature permutation, checkered reshaping, and circular convolution techniques. These enhancements contribute to improved effectiveness of the model. More CNN-based methods include ConvKB [48], ConvR [49], CapsE [50].

Graph Neural Networks (GNNs) are utilized in knowledge graph embedding to overcome the limitations of traditional neural network architectures, such as CNNs, which are designed to handle only Euclidean data. GNNs enable the modeling of non-Euclidean structured data, such as knowledge graphs, by considering the relationships and connectivity between entities in the graph. This allows for more effective and expressive representation learning in KGE tasks. R-GCN [28]) introduces Graph Convolutional Networks (GCN) [51] and extends them to handle the multi-relational nature of realistic knowledge graphs. SACN [29] utilizes weighted GCN, which learns adaptive weights to control the amount of information obtained from neighboring nodes. A recent model based on GNN is CompGCN [31], which is a framework that generalizes KipfGCN [51], R-GCN [28], and D-GCN [52]. CompGCN combines a composition-based GCN as an encoder with ConvE [46] as a decoder. GCNs typically assign fixed weights to all neighbors, treating them equally during information propagation. To overcome this limitation, Graph Attention Networks (GATs) [53] were introduced, which enable the assignment of varying importance levels to neighbors. KBGAT [44] was the first model to learn graph attention-based embeddings on knowledge graphs, followed by RAGAT [30], which constructs separate message functions for different relations and introduces relation-specific parameters to enhance the expressive capability of these message functions, allowing the model to capture relational information in the parameter space.

C. Hyperdimensional Computing

Hyperdimensional computing (HDC) has gained significant attention in recent years due to its promising capabilities in diverse learning applications such as classification, regression, and reinforcement learning. Notably, HDC has demonstrated the potential to surpass conventional machine learning techniques and even outperform traditional neural networks. [54]–[62]. While hyperdimensional computing has shown great promise in competing with convolutional neural networks for graph learning and representation [21], [63], [64], previous HDC-based approaches have not focused on knowledge graph reasoning tasks. In contrast, our work introduces a novel hyperdimensional framework for knowledge graph reasoning, which incorporates the power of HDC graph representation with knowledge graph embedding models to achieve state-of-the-art results on cybersecurity knowledge graphs.

VI. Conclusion

In this work, we proposed INCYSER, a novel HDC-based interpretable knowledge graph representation method for cybersecurity. It employs embedding-based learning of knowledge graphs, combined with HDC-based graph representation. The model's potential and performance is evaluated through a series of experiments, including reasoning tasks on cybersecurity knowledge graphs, such as link prediction and triple classification, as well as general graphs tasks, such as graph reconstruction and shortest path algorithm. INCYSER outperforms state-of-the-art knowledge graph reasoning models in both of accuracy and resource utilization evaluations. Additionally, a detailed ablation study was conducted to further investigate the performance and adaptability of our model under varying conditions.

Future work will focus on several key directions to expand the capabilities and impact of INCYSER. First, we aim to enhance its scalability to support larger and more complex cybersecurity knowledge graphs, accommodating the increasing size and intricacy of real-world datasets. This may involve developing optimized hyperdimensional computing techniques to improve memory and computational efficiency without compromising performance. Second, we plan to explore integration with complementary security tools, enabling seamless incorporation into existing cybersecurity frameworks for tasks such as threat detection, network monitoring, and automated incident response.

Furthermore, we will investigate the application of INCYSER in real-time threat detection scenarios, adapting the model to operate effectively under dynamic and evolving cyber threats. This could involve developing mechanisms for incremental learning, where the model continuously adapts to new data while preserving past knowledge. Additionally, future work will focus on efficiently handling real-time graph updates, enabling the model to dynamically process changes in graph structure without requiring complete re-computation. This includes exploring incremental encoding techniques and memory-efficient update strategies to address the evolving nature of cybersecurity data. This could also pave the way for deeper integration of INCYSER with existing cybersecurity frameworks. By leveraging its interpretable and efficient graph representations, INCYSER could enhance automated workflows, such as real-time threat detection, vulnerability analysis, and incident response planning. These efforts aim to bridge the gap between theoretical advancements and practical implementations, ensuring that INCYSER aligns with industry needs

and effectively addresses dynamic and complex cybersecurity challenges.

REFERENCES

[1] V. Raskin, C. F. Hempelmann, K. E. Triezenberg, and S. Nirenburg, "Ontology in information security: a useful theoretical foundation and methodological tool," in *Proceedings of the 2001 workshop on New security paradigms*, 2001, pp. 53–59.

[2] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: An ontology for intrusion detection," in *Recent Advances in Intrusion Detection: 6th International Symposium, RAID 2003, Pittsburgh, PA, USA, September 8-10, 2003. Proceedings 6*. Springer, 2003, pp. 113–135.

[3] A. Kim, J. Luo, and M. Kang, "Security ontology for annotating resources," NAVAL RESEARCH LAB WASHINGTON DC, Tech. Rep., 2005.

[4] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in *Proceedings of the 4th international Symposium on information, Computer, and Communications Security*, 2009, pp. 183–194.

[5] A. Ekelhart, S. Fenz, and T. Neubauer, "Aurum: A framework for information security risk management," in *2009 42nd Hawaii International Conference on System Sciences*. IEEE, 2009, pp. 1–10.

[6] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain." in *STIDS*, 2012, pp. 49–56.

[7] E. Kiesling, A. Ekelhart, K. Kurniawan, and F. Ekaputra, "The sepses knowledge graph: an integrated resource for cybersecurity," in *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, 2019, pp. 198–214.

[8] A. Piplai, S. Mittal, A. Joshi, T. Finin, J. Holt, and R. Zak, "Creating cybersecurity knowledge graphs from malware after action reports," *IEEE Access*, vol. 8, pp. 211 691–211 703, 2020.

[9] I. Sarhan and M. Spruit, "Open-cykg: An open cyber threat intelligence knowledge graph," *Knowledge-Based Systems*, vol. 233, p. 107524, 2021.

[10] N. Rastogi, S. Dutta, R. Christian, J. Gridley, M. Zaki, A. Gittens, and C. Aggarwal, "Predicting malware threat intelligence using kgs," *arXiv e-prints*, pp. arXiv–2102, 2021.

[11] K. Kurniawan, A. Ekelhart, E. Kiesling, G. Quirchmayr, and A. M. Tjoa, "Krystal: Knowledge graph-based framework for tactical attack discovery in audit data," *Computers & Security*, vol. 121, p. 102828, 2022.

[12] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang, and Z. Tian, "Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[13] H. T. Nguyen, K. Franke, and S. Petrovic, "Feature extraction methods for intrusion detection systems," in *Threats, Countermeasures, and Advances in Applied Information Security*. IGI Global, 2012, pp. 23–52.

[14] Y. Liu, J. Zhang, A. Sarabi, M. Liu, M. Karir, and M. Bailey, "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*, 2015, pp. 3–9.

[15] W. Hu, Y. Liao, and V. R. Vemuri, "Robust support vector machines for anomaly detection in computer security." in *ICMLA*, 2003, pp. 168–174.

[16] C.-W. Ten, J. Hong, and C.-C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865–873, 2011.

[17] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, and Y. Zhou, "Recent progress of using knowledge graph for cybersecurity," *Electronics*, vol. 11, no. 15, p. 2287, 2022.

[18] R. Gayler, "Multiplicative binding, representation operators and analogy," 1998.

[19] P. Kanerva, "Binary spatter-coding of ordered k-tuples," in *International conference on artificial neural networks*. Springer, 1996, pp. 869–873.

[20] T. Plate *et al.*, "Holographic reduced representations: Convolution algebra for compositional distributed representations." in *IJCAI*. Citeseer, 1991, pp. 30–35.

[21] A. Zakeri, Z. Zou, H. Chen, H. Latapie, and M. Imani, "Conjunctive block coding for hyperdimensional graph representation," *Intelligent Systems with Applications*, vol. 22, p. 200353, 2024.

[22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

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

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] J. S. Garrido, D. Dold, and J. Frank, "Machine learning on knowledge graphs for context-aware security monitoring," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2021, pp. 55–60.

[26] M. Nickel, V. Tresp, H.-P. Kriegel *et al.*, "A three-way model for collective learning on multi-relational data." in *Icml*, vol. 11, no. 10.5555, 2011, pp. 3 104 482–3 104 584.

[27] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," *Advances in neural information processing systems*, vol. 31, 2018.

[28] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "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*. Springer, 2018, pp. 593–607.

[29] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3060–3067.

[30] X. Liu, H. Tan, Q. Chen, and G. Lin, "Ragat: Relation aware graph attention network for knowledge graph completion," *IEEE Access*, vol. 9, pp. 20 840–20 849, 2021.

[31] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," *arXiv preprint arXiv:1911.03082*, 2019.

[32] Y. Zhao, B. Lang, and M. Liu, "Ontology-based unified model for heterogeneous threat intelligence integration and sharing," in *2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*. IEEE, 2017, pp. 11–15.

[33] Y. Deng, D. Lu, D. Huang, C.-J. Chung, and F. Lin, "Knowledge graph based learning guidance for cybersecurity hands-on labs," in *Proceedings of the ACM conference on global computing education*, 2019, pp. 194–200.

[34] M. Du, J. Jiang, Z. Jiang, Z. Lu, and X. Du, "Prtirg: a knowledge graph for people-readable threat intelligence recommendation," in *Knowledge Science, Engineering and Management: 12th International Conference, KSEM 2019, Athens, Greece, August 28–30, 2019, Proceedings, Part I 12*. Springer, 2019, pp. 47–59.

[35] B. A. Mozzaquatro, C. Agostinho, D. Goncalves, J. Martins, and R. Jardim-Goncalves, "An ontology-based cybersecurity framework for the internet of things," *Sensors*, vol. 18, no. 9, p. 3053, 2018.

[36] C. Xiong, T. Zhu, W. Dong, L. Ruan, R. Yang, Y. Cheng, Y. Chen, S. Cheng, and X. Chen, "Conan: A practical real-time apt detection system with high accuracy and efficiency," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 551–565, 2020.

[37] T. Chen, C. Dong, M. Lv, Q. Song, H. Liu, T. Zhu, K. Xu, L. Chen, S. Ji, and Y. Fan, "Apt-kgl: An intelligent apt detection system based on threat knowledge and heterogeneous provenance graph learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[38] Neo4j graph database graph data science. [Online]. Available: https://neo4j.com

[39] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[40] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *Proceedings of the international AAAI conference on web and social media*, vol. 3, no. 1, 2009, pp. 361–362.

[41] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[42] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[43] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu, "Knowledge graph alignment network with gated multi-hop neighborhood aggregation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 222–229.

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

[45] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4498–4507.

[46] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[47] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020, pp. 3009–3016.

[48] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," *arXiv preprint arXiv:1712.02121*, 2017.

[49] X. Jiang, Q. Wang, and B. Wang, "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)*, 2019, pp. 978–987.

[50] T. Vu, T. D. Nguyen, D. Q. Nguyen, D. Phung *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)*, 2019, pp. 2180–2189.

[51] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[52] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *arXiv preprint arXiv:1703.04826*, 2017.

[53] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[54] C.-Y. Chang, Y.-C. Chuang, C.-T. Huang, and A.-Y. Wu, "Recent progress and development of hyperdimensional computing (hdc) for edge intelligence," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2023.

[55] L. Ge and K. K. Parhi, "Applicability of hyperdimensional computing to seizure detection," *IEEE Open Journal of Circuits and Systems*, vol. 3, pp. 59–71, 2022.

[56] Z. Zou, H. Alimohamadi, A. Zakeri, F. Imani, Y. Kim, M. H. Najafi, and M. Imani, "Memory-inspired spiking hyperdimensional network for robust online learning," *Scientific reports*, vol. 12, no. 1, p. 7641, 2022.

[57] X. Jiao, A. Rahimi, C. Fermüller, and J. Y. Aloimonos, "Brain-inspired hyperdimensional computing: Algorithms, models, and architectures," *Frontiers in Neuroscience*, vol. 16, 2022.

[58] M. Imani, A. Zakeri, H. Chen, T. Kim, P. Poduval, H. Lee, Y. Kim, E. Sadredini, and F. Imani, "Neural computation for robust and holographic face detection," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 31–36.

[59] Y. Ni, M. Issa, D. Abraham, M. Imani, X. Yin, and M. Imani, "Hdpg: Hyperdimensional policy-based reinforcement learning for continuous control," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 1141–1146.

[60] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan *et al.*, "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," *Nature Electronics*, vol. 4, no. 1, pp. 54–63, 2021.

[61] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[62] A. Hernández-Cano, Y. Ni, Z. Zou, A. Zakeri, and M. Imani, "Hyperdimensional computing with holographic and adaptive encoder," *Frontiers in Artificial Intelligence*, vol. 7, p. 1371988, 2024.

[63] P. Poduval, A. Zakeri, F. Imani, H. Alimohamadi, and M. Imani, "Graphd: Graph-based hyperdimensional memorization for brain-like cognitive learning," *Frontiers in Neuroscience*, p. 5.

[64] I. Nunes, M. Heddes, T. Givargis, A. Nicolau, and A. Veidenbaum, "Graphhd: Efficient graph classification using hyperdimensional computing," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1485–1490.